Poisson-Binomial Counting for Learning Prediction Sparsity

Submitted in Partial Fulfillment of the Requirement for the Degree of Doctor of Philosophy

Julien Schroeter

School of Computer Science and Informatics
Cardiff University

Supervision:
Dr. Kirill Sidorov
Prof. David Marshall

Examiners:
Prof. Andrea Vedaldi
Prof. Yukun Lai

February 2021

Acknowledgements

First and foremost, I would like to thank my supervisors Prof. David Marshall and Dr. Kirill Sidorov for offering me the opportunity to pursue a PhD in this field and, most importantly, for their continuing support and guidance throughout my three years at Cardiff University. I am also extremely grateful to the School of Computer Science and Informatics who generously granted me a full scholarship for this research project. To my friends and colleagues in the office (Tom, Aled, Joe, Stefano) and to all the ones upstairs (Roberto, Anastazia, ...), I am grateful for all the great moments we shared together. A special thanks to Roberto for his expert eye and generous proofreading of my papers.

I am very grateful to Prof. Tinne Tuytelaars for hosting me in Leuven and for her kind and expert supervision. I also would like to acknowledge the support of the KU Leuven-Cardiff University partnership fund which made this collaboration possible. Thank you to everyone at KU Leuven for making me feel so welcome (especially Thomas and Matthias).

I give my thanks to the examiners (Prof. Andrea Vedaldi, Prof. Yukun Lai) for the constructive feedback.

Finally, I could sincerely not have completed this thesis without Daphné. No amount of words can express how grateful I am for her continued patience, support and kindness. You are simply the best, I love you.

Julien Schroeter

Abstract

The loss function is an integral component of any successful deep neural network training; it guides the optimization process by reducing all aspects of a model into a single number that must best capture the overall objective of the learning. Recently, the maximum-likelihood parameter estimation principle has grown to become the default framework for selecting loss functions, hence resulting in the prevalence of the cross-entropy for classification and the mean-squared error for regression applications (Goodfellow et al., 2016). Loss functions can however be tailored further to convey *prior knowledge* about the task or the dataset at hand to the training process (e.g., class imbalances (Huang et al., 2016a; Cui et al., 2019), perceptual consistency (Reed et al., 2014), and attribute awareness (Jiang et al., 2019)). Overall, by designing loss functions that account for known priors, a more targeted supervision can be achieved with often improved performance.

In this work, we focus on the ubiquitous prior of *prediction sparsity*, which underlines many applications that involve probability estimation. More precisely, while the iterative nature of gradient descent learning often requires models to be able to continuously reach any probability estimates between 0 and 1 during training, the optimal solution to the optimization problem (w.r.t. the ground-truth) is often sparse with clear-cut probabilities (i.e., either converging towards 1 or 0). For instance, in object detection, the decision that must be made by the models to either keep or discard estimated bounding-boxes for final predictions (e.g., non-maximum suppression) is binary. Similarly, in music onset detection, the optimal predictions are sparse: it is known that only a few points in time should be assigned a high likelihood, while no probability mass should be allocated to all other timesteps. In these applications, incorporating this important prior directly in the training process through the design of the loss function would offer a more tailored supervision, that better captures the underlying objective.

To that effect, this work introduces a novel *loss function* that relies on instance counting to achieve prediction sparsity. More precisely, as shown in the theoretical part of this work, modeling occurrence counts as a Poisson-binomial distribution results in a differentiable training objective that has the unique intrinsic ability to converge probability estimates towards sparsity. In this setting, sparsity is thus not attained through an explicit sparsity-inducing operation, but is rather implicitly learned by the model as a byproduct of learning to count instances. We demonstrate that this cost function can be leveraged as a standalone loss function (e.g., for the weakly-supervised learning of temporal localization) as well as a sparsity regularization in conjunction with other more targeted loss functions to enforce sparsity constraints in an end-to-end fashion. By design, the proposed approach finds use in the many applications where the optimal predictions are known to be sparse. We thus prove the validity of the loss function on a wide array of tasks including weakly-supervised drum detection, piano onset detection, single-molecule localization microscopy, and robust event detection in videos or in wearable sensors time series. Overall, the experiments conducted in this work not only highlight the effectiveness and the relevance of Poisson-binomial counting as a means of supervision, but also demonstrate that integrating prediction sparsity directly in the learning process can have a significant impact on generalization capability, noise robustness, and detection accuracy.

Table of Contents

Nomenclature

$\mathbf{X}$	Matrix
$\mathbf{x}$	Vector
x	Scalar
$\big(x_n\big)_{n=1}^{N}$	Sequence of scalars x_n for $n = 1, 2, \ldots, N$
$\hat{x}_\theta$	Estimated scalar using model parameter θ.

Notation: In temporal applications, a sequence is either denoted as a matrix $\mathbf{X}$ if it stands for a sequence of vectors $\big(\mathbf{x}_t\big)_{t=1}^{T}$ (e.g., a spectrogram or a video) or as a vector $\mathbf{x}$ if the elements constituting the sequence are scalars x_i (e.g., a sequence of univariate labels). When the nature of the sequence is unspecified, the default is a matrix $\mathbf{X}$ for temporal input data and a vector $\mathbf{y}$ for label sequences.

$\mathcal{B}(p)$	Bernoulli random variable with success probability p
$\mathscr{C}$	Poisson-binomial Count (Definition 2.2)
C	Deterministic Count (Definition 2.1)
$\mathbb{1}_c$	Discrete Dirac Distribution, i.e., $\Pr(\mathbb{1}_c\!=\!k)\!=\!1 \iff k\!=\!c$
$\mathcal{N}$	Normal Distribution

$\Pr(\cdot)$	Probability
D_{KL}	Kullback-Leibler Divergence (1951)
$\lfloor \cdot \rceil$	Rounding to nearest integer

NMS	Non-maximum Suppression
MSE	Mean-squared Error
CE	Cross-entropy
DNN	Deep Neural Network
LTP	Law of total probability

Chapter 1

Introduction

Learning to *count* is a ubiquitous task in machine learning. Not only does it find direct practical applications in the form of crowd counting (Zhang et al., 2015a, 2016), object counting (Onoro-Rubio & López-Sastre, 2016), microscopy cell counting (Xie et al., 2018), vehicle counting (Mundhenk et al., 2016; Zhang et al., 2017b) or even fruit counting (Rahnemoonfar & Sheppard, 2017), but it can also be leveraged for self-supervised visual representation learning (Noroozi et al., 2017) or question answering (Trott et al., 2018; Acharya et al., 2019). More indirectly, counting also implicitly underlies many other learning tasks. For instance, in object detection (Redmon et al., 2016), a model capable of inferring accurate bounding-box locations has indirectly learned to count the number of object instances. Similarly, in music transcription (Sigtia et al., 2016), a model that can accurately detect notes can also count them. Reversely, a model consistently outputting too few or too many boxes—or notes respectively—is inevitably bound to perform poorly; the inference of correct instance counts is thus a necessary underlying condition for precise and consistent detection. These examples highlight both the intrinsic link between instance counting and object (or event) detection, as well as the nature of counting as a weaker sub-task of the more complex localization objective in most spatial and temporal detection tasks.

In many applications, instance countability, comes in tandem with prediction *sparsity*, i.e., the optimal probability estimates are clear-cut with value 0 or 1. For instance, in object detection, it is quite common for models to first output numerous bounding-boxes that then have to be selected through non-maximum suppression (NMS) or other thresholding heuristics. Thus, the decision that eventually must be made by the models to either keep or discard estimated bounding-boxes as final predictions is binary and, by extension, sparse selection probabilities have to be assigned to each bounding-box. Similarly, in drum transcription (Wu et al., 2018) and piano onset detection (Hawthorne et al., 2017), only a few of the hundred timesteps per second contain a note onset; thus, the non-zero values of the label time series (i.e., the event locations) are sparsely scattered over time. In all these examples, it is the sparsity of instances that make them countable and vice versa. For instance, in object detection, objects can be represented as sparse bounding-boxes because of their countability and, conversely, the number of boxes can be counted because of their sparse and scattered nature. In contrast, water drops in the sea cannot be counted because they are densely distributed in space, nor can their location be precisely determined because they are indistinguishable from one another and thus uncountable.

Despite being a prevalent prior in numerous applications, prediction sparsity is often modeled separately from the learning process or even overlooked. For instance, many object detection methods (Girshick, 2015; Ren et al., 2015; Redmon et al., 2016; Liu et al., 2016) heavily rely on non-maximum suppression to select clear-cut final predictions from the pile of overlapping bounding-boxes they generate, while temporal detection models often require the use of heuristics such as peak-picking (Cogliati et al., 2016; Wu et al., 2018), thresholding (Schlüter & Böck, 2014), or argmax operation (McNally et al., 2019) to achieve the desired prediction sparsity. By splitting the detection process into a trainable component and a fixed heuristic, these approaches break the end-to-end learning paradigm that allows deep learning models to best learn the mapping between input data and learning objective (Krizhevsky et al., 2012; Sutskever et al., 2014; Long et al., 2015; Levine et al., 2016). In contrast, modeling prediction sparsity directly as part of the training process can offer a more tailored and fully end-to-end supervision, which can ultimately lead to improved generalization capabilities.

This thesis proposes a novel paradigm based on the intrinsic link between instance countability and sparsity: **learning to count as a way of learning prediction sparsity**. More specifically, this work introduces a novel loss function (Chapter 2) that relies on instance counting as a means of supervision. The usefulness of the proposed counting-based training objective stems from its unique ability to indirectly drive probability estimates towards sparsity (i.e., towards either 0 or 1) as the learning progresses (Chapter 3). This work thus shows how instance counting can be leveraged to incorporate the common prior of prediction sparsity into the learning of almost any task dealing with probability assignments of countable instances without harming the end-to-end training process.

1.1 Related Works

The loss function introduced in this work has the unique ability to achieve prediction sparsity through count supervision. This section thus highlights how this paradigm relates to other count-based learning models and other sparsity-achieving methods.

1.1.1 Count Supervision

Humans are capable of instantaneously discerning how many objects are present in a visual scene without having to sequentially count or spot each individual instance when the number of objects is small (Kaufman et al., 1949; Mandler & Shebo, 1982). This ability, known as subitizing, is not specific to the visual domain, but also applies, among others, to tactile perception (Riggs et al., 2006) and auditory perception (Camos & Tillmann, 2008). However, beyond the subitizing range (3–4 objects), the enumeration of objects is less intuitive and often requires the explicit counting of individual instances to be accurate. This known dichotomy in the way humans handle the task of counting also appears in the literature on count-based learning models. Indeed, models that learn through count supervision can be split into two distinct categories: direct approaches which directly map the input data to the objective counts (i.e., similar to subitizing) and bottom-up approaches which first identify positive instances before aggregating them into a global count (i.e., similar to explicit instance counting).

Direct Classification

In the realm of direct counting, the closest equivalent to human subitizing consists in formulating the task of counting as a classification problem by viewing each potential count outcome as an independent class. In terms of model training, this straightforward framework allows for a seamless parameter optimization using the standard cross-entropy. This learning approach has been leveraged to train models in numerous application domains including object counting (Zhang et al., 2015b), digit and crowd counting (Seguí et al., 2015), embryonic cell counting (Khan et al., 2016), chimpanzee recognition (Bain et al., 2019), and counting-based visual question-answering (Acharya et al., 2019). However, as each potential count value is mapped to an *independent* class, all information about the actual ordering of count values (e.g., $4 < 7 < 13$) is indirectly discarded. This lack of underlying hierarchy between count classes is problematic since models cannot rely on known structures of counts (e.g., 3 lies between 2 and 4) to learn to count. In such a framework, the ambiguity between the different classes thus increases exponentially with the number of count classes. As a result, similarly to subitizing, this approach does not scale well to larger count values, and thus may only be used for applications with a limited range of potential counts.

Several alternatives have been proposed to alleviate the modeling weaknesses inherent to direct count classification. For instance, Mundhenk et al. (2016) suggested performing car count classification on smaller sub-patches instead of on the larger original image. While this approach artificially reduces the number of instances per processed region and thus ensures that the counts stay close to the subitizing range, this method is limited to applications where the space can be clearly partitioned and where the instances can be uniquely assigned to a single sub-partition. Another approach, proposed by Stöter et al. (2018) in the context of audio source counting, consists in modeling counts not as a Dirac distribution, but rather as a Poisson distribution—a common distribution for counting (Chan & Vasconcelos, 2009; Fallah et al., 2009) and set cardinality in general (Rezatofighi et al., 2017). In this setup, models are thus trained to estimate the value of each count class, which amounts to inferring the mass of the corresponding bins of the Poisson distribution. Overall, by spreading the probability mass across several count classes, this method correlates neighbouring classes, and thus implicitly incorporates a sense of ordering into the classification model. However, while providing a simple solution to the lack of hierarchy between count classes, this approach also presents several drawbacks (e.g., models output a spread-out count distribution rather than a clear-cut count value, models have to invest resources to explicitly learn to replicate a given distribution rather than to directly learn to count).

Direct Regression

Count classification ensures that predicted counts are integer values. However, by relaxing this setting and allowing for fractional counts, the counting problem can be cast as a regression problem. Overall, while non-integer counts can cause modeling issues in some settings, regression—unlike classification—possesses an intrinsic ordering of values, which can benefit the learning process. This framework is thus more suited than direct classification for counting applications that handle counts beyond the subitizing range. In practice, count regression has been leveraged in numerous domains (e.g., crowd counting (Kong et al., 2006; Shang et al., 2016; Huang et al., 2017), fruit counting (Rahnemoonfar & Sheppard, 2017), leaves counting (Giuffrida et al., 2016), and object counting (Song & Qiu, 2018)). During training, the model parameters are commonly optimized using the mean squared error (MSE), but other loss functions have also been considered (e.g., Huber loss (Chattopadhyay et al., 2017) and ratio loss (Giuffrida et al., 2016)).

Count regression can also be used to accurately model counts when, as mentioned above, the original sample is sub-divided into non-overlapping sub-samples in an effort to limit the complexity of the input and reduce the range of counts. Indeed, since in such settings instances often lie in several partitions, it is essential to be able to split and distribute integer counts across several partitions in order to accurately model this effect. The use of count regression is therefore a natural and effective choice in this context, as it offers the possibility of inferring fractional count targets. This specific approach has been used, among others, for crowd detection (Ryan et al., 2009; Hu et al., 2016), dynamic non-maximum suppression (Chattopadhyay et al., 2017), and self-supervised count-based representation learning (Noroozi et al., 2017).

Explicit Instance Counting

While direct counting offers a simple and streamlined modeling of counts, it often lacks both the explainability and the generalization capabilities of models that perform counting through explicit instance identification (e.g., detection of individual objects in computer vision applications). Indeed, for instance, performing car counting through explicit car detection (Moranduzzo & Melgani, 2013; Hsieh et al., 2017) (instead of direct count regression or classification) results in both more interpretable count predictions that can be traced back to the individual detections and a stronger supervision of the learning that can leverage labeled car positions for training. This latter feature is expected to significantly facilitate, among other benefits, the learning of car representations, and thus improve the overall generalization performance of the approach. As a result, such bottom-up approaches often have to rely on finer-grained annotations (e.g., point location of individual instances vs global counts). This additional annotation burden is however minimal when taking into account the way humans

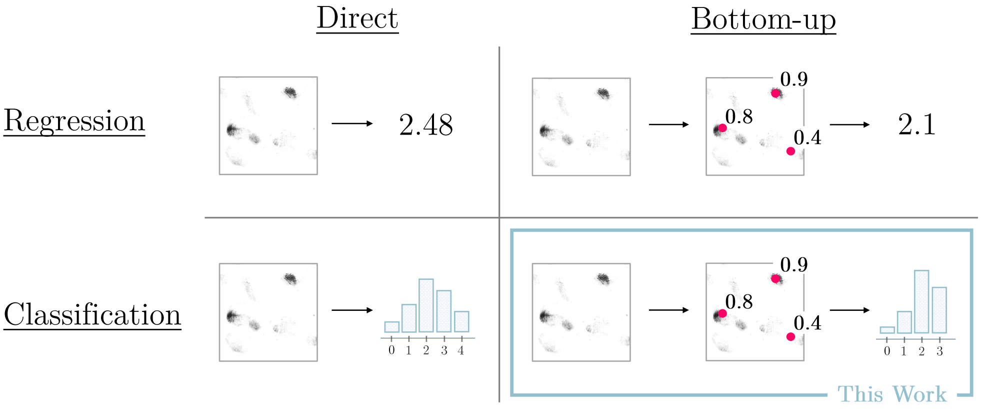


Figure 1.1: Count Supervision. This work proposes a novel bottom-up classification approach to counting. Our model-based method preserves the implicit hierarchy between the count classes (e.g., 3 lies between 2 and 4) in contrast to standard classification-based models, which often discard all information about the underlying ordering of the count classes.

actually count. Indeed, when the number of instances exceeds the subitizing range, annotators have to explicitly identify individual instances in order to determine the correct count; thus, reporting these individual instances on top of the global count does not substantially complicate the annotation process.

In computer vision, numerous methods cast counting purely as an instance detection or localization problem. For instance, in dense scenes (e.g., crowd counting, car counting, and microscopy cell counting), models are directly trained to infer densities through heatmap-matching without any explicit count supervision (Lempitsky & Zisserman, 2010; Arteta et al., 2014, 2016; Boominathan et al., 2016; Onoro-Rubio & López-Sastre, 2016; Zhao et al., 2016; Walach & Wolf, 2016; Paul Cohen et al., 2017; Xie et al., 2018; Cao et al., 2018; Cheng et al., 2019). In such a framework, while the total number of instances can be inferred through density integration, counts do not explicitly appear in the training process but only exist implicitly in the number of point labels. Thus, in order to integrate the actual objective of the pipeline (i.e., counting) more explicitly in the learning, several methods include the target count as a training objective through count regularization (Idrees et al., 2018; Sam & Babu, 2018; Wan & Chan, 2019) or alternate between density estimation learning and count regression learning (Zhang et al., 2015a). In that context, the counting loss is generally defined as the (mean) squared difference between the integrated and the ground-truth counts. Thus, this kind of approach can be viewed as a count regression, where counts are sums of deterministic sub-counts. Other approaches that include count supervision in the learning process exist . For instance, Ma et al. (2019) propose a Bayesian variation of density estimations where point labels are considered as priors rather than learning targets. While the counts are still inferred through density integration, training is based on the ℓ_1-loss between ground-truth and predicted count which is computed through the multiplication of the density estimate and the posterior label probability. Similarly, in temporal event localization, Narayan et al. (2019) propose including count supervision on top of the standard category-based weakly-supervised approach. Once again, the ground-truth counts are compared to the predicted counts, which are the result of integration over the count densities computed as the product of the temporal attention and the temporal class activation map. Finally, (Laradji et al., 2018) perform object counting though the explicit localization and counting of object blobs, which are learned through count and point location supervision.

In count-based visual question-answering, Trott et al. (2018) propose sequentially selecting bounding-boxes until all objects have been accounted for. In this setup, since the model makes hard decisions, the counting loss—which is defined as the absolute error between the total number of selected boxes and the ground-truth count—is not differentiable and thus has to be optimized through reinforcement learning. Their work also introduces as benchmark a softer approach, where each bounding-box is assigned a probability. The count is thus defined as the deterministic sum of these probabilities and the training is done through backpropagation using a Huber loss (Huber, 1964). In terms of explainability, these two bottom-up approaches output count predictions that are significantly more interpretable than the ones obtained though direct count classification (Acharya et al., 2019). Indeed, even though the softer approach yields less clear-cut predictions, the contribution of each bounding-box to the count prediction can be clearly measured by the probability assigned to them.

Other Approaches

There exist several alternative approaches that take advantage of counting as a means of supervision. For instance, Zhang et al. (2017b) propose a hybrid method to tackle temporal vehicle counting that leverages two different approaches to counting: integration of a density map to count spatially and direct count regression to count temporally. Liu et al. (2018) introduce a ranking loss (pairwise ranking hinge loss) which allows the model to exploit the implicit property that image crops contain lower counts than the original image. Finally, Gao et al. (2018) leverage count-based multiple-instance learning to perform weakly-supervised object detection.

Our Approach

This work proposes a novel bottom-up counting loss function. However, in contrast to other explicit instance counting methods which often view count as a deterministic sum of individual instance contributions, this work models count as a sum of Bernoulli distributions (see Chapter 2 for more details). Thus, instead of predicting scalar counts and training the model using regression-based loss functions (e.g., MSE), our proposed approach infers count distributions and the model training is performed through KL-divergence. Overall, while the differences between the two approaches might seem subtle, modeling counts as a sum of individual Bernoulli distribution (i.e., Poisson-binomial counts) offers several key advantages, such as an implicit convergence towards sparse instance predictions (see Chapter 3).

This novel approach can also be viewed as a count classification model with an implicit hierarchy (i.e., ordering) between the count classes. In fact, inferring count distributions can be interpreted as a means to augment the otherwise independent count classes of standard classification approaches with an explicit distribution that models the underlying ordering of the bins (e.g., modeling counts as Poisson distributions implicitly tells that $3<4$). However, in contrast to Stöter et al. (2018) which explicitly estimate the probability assigned to each bin of the count distribution and which add the distribution modeling through a custom loss function, our work estimates the distribution indirectly through a bottom-up counting approach. Indeed, in our framework, the bins of the count distribution do not require a direct and explicit estimation since their value is uniquely defined by the individual instance predictions and the choice of underlying distribution: the Poisson-binomial distribution of count.

1.1.2 Prediction Sparsity

Actionable decision making often calls for hard and definitive choices. For instance, when grocery shopping, someone wishing to buy five tomatoes has to decisively select exactly five whole fruits from the stand to fulfill their objective; they cannot choose ten halves nor select a chunk of each available tomato to reach the desired outcome. Similarly, when a medical professional has to make a decision about a procedure—although it is useful to weigh up its risks and benefits, the final choice whether or not to proceed with the intervention is ultimately binary. Thus, while in some settings it might be beneficial to train deep neural models to quantify uncertainty or to account for outcome variability, in other setups, clear-cut decisions are required.

Training models to make hard decisions in an end-to-end manner is however a challenging feat. Indeed, one of the prerequisites for end-to-end backpropagation-based learning is the continuous differentiability of the loss functions with respect to every model parameter. Thus, for instance, the task of finding (or training) a non-trivial differentiable function that maps continuous inputs to clear-cut categorical outputs (e.g., "yes or no?" rather than "what is the probability of yes?") is mathematically unfeasible (e.g., Heaviside function). As a consequence, any practical approach to prediction sparsity is required to slightly loosen the overall setup. In recent years, two main alternatives have been proposed: either discarding the differentiability assumption or relaxing the hard-sparsity objective.

Non-differentiable Approaches

Dropping the requirement for continuous and well-defined gradients is a straightforward way to output sparse predictions. A common approach to train models in this context is to rely on reinforcement learning to optimize the non-differentiable loss function. For instance, Trott et al. (2018) leverage reinforcement learning to train a hard sequential decision process to select bounding-boxes and decisively answer counting-based questions. Multiple variations of this principle exist in sequential search or detection algorithm (Caicedo & Lazebnik, 2015; Mathe et al., 2016). Another widespread approach in this domain consists in training the model end-to-end on a sub-objective before relying on additional heuristics to perform the non-differentiable sparsity-inducing operation. For instance, this principle is widely applied in object detection where models (Girshick, 2015; Ren et al., 2015; Redmon et al., 2016; Liu et al., 2016) are often trained to output a large collection of potential bounding-boxes (with a wide range of detection scores that are not necessarily close to 0 nor 1), while the clear-cut selection of the final boxes is performed, as a second step, through a fixed non-maximum suppression operation. The same observation holds for temporal detection applications where most models heavily rely on non-differentiable components (e.g., argmax operation (McNally et al., 2019), peak-picking (Cogliati et al., 2016; Wu et al., 2018), and thresholding (Schlüter & Böck, 2014)) to obtain sparse predictions. Overall, the main drawback of this approach is that the optimization does not include all parts of the model, and thus optimality on the sub-task might not necessarily equate to optimality on the actual overall objective.

Soft Sparsity

An alternative consists in relaxing the hard-sparsity objective by allowing for softer probability assignments, and thus keeping the continuous differentiability assumption. More precisely, instead of incorporating the process of hard decision making directly on top of the model, the loss function or the architecture itself can be designed in a way that encourages the sparsity of the predictions rather than explicitly imposes it. For instance, instead of performing hard bounding-box selection (e.g., non-maximum suppression), Hosang et al. (2017) propose a method that relies on the rescoring of the original detection probabilities assigned to each bounding-box; while sparsity is not guaranteed, this softer approach encourages the scores to converge towards a sparser representation. Another strategy consists in replacing the softmax with alternative activation functions that explicitly yield sparser predictions (Martins & Astudillo, 2016; Martins & Kreutzer, 2017; Malaviya et al., 2018). This approach is especially common in text modeling where instances are of a discrete nature. Finally, the Gumbel softmax trick (Jang et al., 2017; Maddison et al., 2017), which allows to sample discrete random variables in a differentiable way, can be considered as a sparsity-inducing operation. While this approach outputs perfectly sparse predictions (i.e., one-hot vectors) in the forward pass, it still relies on soft probability assignments in the backward pass. Thus, in terms of sparsity, this method can still be considered as a relaxation of the hard decision objective.

Our Approach

In this work, prediction sparsity is learned in an end-to-end fashion through count supervision. Similarly to the bounding-box rescoring approach (Hosang et al., 2017), sparsity is not an objective on its own, but models are indirectly driven towards outputting sparser predictions as the learning progresses. More specifically, the loss function is designed in such a way that models are bound to output sparse predictions in order to count successfully (see Chapter 3); thus, predictions sparsity emerges implicitly as the models learn to count instances. In contrast to sparse activation-based loss functions (Martins & Astudillo, 2016; Martins & Kreutzer, 2017; Malaviya et al., 2018) or to the Gumbel softmax trick (Jang et al., 2017; Maddison et al., 2017), the approach introduced in this work does not rely on any explicit sparsity-inducing operation but rather encourages the model itself to learn to output sparse predictions. Finally, by transforming the sparse selection process into a counting problem, the proposed model allows to explicitly control and learn the number of non-zero instances; this contrasts with numerous previous works which achieve sparsity without any direct control on the number of non-zero instances.

1.2 Thesis Structure and Content

Thesis Structure

The first part of the thesis introduces the novel Poisson-binomial counting loss function (Chapter 2) and its unique sparsity-inducing property (Chapter 3). The remainder of the work presents a wide array of applications that demonstrate the effectiveness, versatility, and usability of the proposed loss function: counting-based weakly-supervised temporal localization (Chapter 4), robust temporal event detection (Chapter 5), and multi-instance sub-pixel point detection in images (Chapter 6). Finally, several avenues for future research are also considered (Chapter 7).

The experiments of this work cover a wide array of tasks including counting-based visual question answer (Section 3.3), drum detection (Section 4.3.1, Section 5.6.4), piano onset detection (Section 4.3.2, Section 5.6.3), digit detection (Section 4.3), golf event sequencing in videos (Section 5.6.1, Section 6.4.3), smoking puff detection using wearable sensor data (Section 5.6.2), single molecule localization microscopy

(Section 6.4.1), checkerboard corner detection (Section 6.4.2), and adversarial attack on object classification (Section 7.2).

Publications

The content of this thesis is based on the following publications:

1. Schroeter, J., Sidorov, K., and Marshall, D. Weakly-supervised temporal localization via occurrence count learning. In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 5649–5659, 2019.

2. Schroeter, J., Sidorov, K., and Marshall, D. Robust temporal point event localization through smoothing and counting. In *ICML Workshop on Uncertainty & Robustness in Deep Learning (UDL)*, 2020.

3. Schroeter, J., Tuytelaars, T., Sidorov, K., and Marshall, D. Learning multi-instance sub-pixel point localization. In *Asian Conference on Computer Vision (ACCV)*, 2020.

4. Schroeter, J., Sidorov, K., and Marshall, D. Learning Precise Temporal Point Event Detection with Misaligned Labels. In AAAI (to appear), 2021.

Novelty

The Poisson-binomial loss function, the derivation of its properties, its applications—both as a standalone training objective and as regularizer—to the various tasks included in this work (weakly-supervised temporal localization, robust temporal event detection, and multi-instance sub-pixel point detection in images), the $\mathcal{S}$oftLoc loss function for the robust learning of temporal point event detection, the multi-instance subpixel point detection model based on offset regression, soft localization learning and sparsity regularization, as well as the discussion of future research avenues are all novel contributions of this work.

New Content

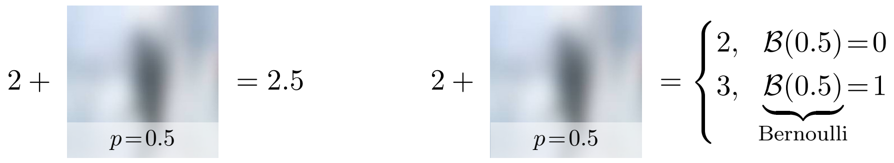
While the content of this thesis is mainly based on work presented in (Schroeter et al., 2019, 2020a,b, 2021), many new elements have been incorporated into the text. The most significant of these include more detailed proofs (Section 2.1.1, Chapter 3), additional implementation details (Section 2.3), a full section about the intuition behind the sparsity-inducing ability of the Poisson-binomial loss function (Chapter 3.1), additional properties of the loss function (Section 3.2.1), an illustrative visual question answering experiment (Section 3.3), a more detailed explanation about the convergence of the weakly-supervised model (Section 4.2.3), extended experiment results and discussions (Section 4.3, Section 5.6, Section 6.4), a succinct extension of the model to semi-supervised learning (Section 7.1), an additional counting-based adversarial attack experiment (Section 7.2), a discussion about learning non-maximum suppression in object detection (Section 7.3), as well as additional illustrations.

Chapter 2

Poisson-Binomial Counting

The action of counting is an unequivocal operation when the instances that have to be counted are well-defined. Indeed, it simply consists in identifying individual instances and enumerating them. However, in the presence of uncertainty about the existence of an instance, counting becomes an ambiguous action. For example, when counting people in a low-resolution image, it can be unclear whether a blurry spot corresponds to a person or should be discarded as noise. In this case, the counting operation cannot simply rely on instance enumeration anymore: it has to take into account the uncertainty underlying the nature of such ambiguous instances.

In fact, integrating such an ambivalent instance—with probability p of being of interest, and consequently with probability $(1-p)$ of being irrelevant—into the global count can be done in two different ways. One the one hand, the value of the instance probability p can be directly added as a non-integer fractional contribution to the count (see Figure 2.1a). This deterministic approach to counting appears in numerous applications—e.g., heatmap integration (Lempitsky & Zisserman, 2010; Idrees et al., 2018) and sub-sample count aggregation (Hu


2 +
$p=0.5$
$= 2.5$
2 +
$p=0.5$
$= \begin{cases} 2, & \mathcal{B}(0.5)=0 \\ 3, & \underbrace{\mathcal{B}(0.5)}_{\text{Bernoulli}}=1 \end{cases}$

(a) Deterministic Counting

(b) Poisson-Binomial Counting

Figure 2.1: Two different approaches for performing addition with uncertain instances: deterministic and stochastic (i.e., Poisson-binomial) counting.

et al., 2016; Noroozi et al., 2017)—as it offers a straightforward means to model and infer counts. However, by mapping stochastic instances to deterministic fractional count contributions, this method discards all uncertainty, and thus the final count scalar does not accurately reflect the ambiguity present in the sample. For instance, in this framework, two instances with probability $p = 0.5$ each or a single instance with probability $p = 1$ would both yield the same overall count, even though the two situations are fundamentally different.

On the other hand, a full contribution of 1 can be added to the count with probability p (see Figure 2.1b). While equivalent to its deterministic counterpart in unambiguous settings (i.e., $p \in \{0, 1\}$), this approach differs in its unique way of modelling counting as a stochastic—rather than deterministic—operation. For instance, in this setup, a single instance with probability $p = 1$ or two instances with probability $p = 0.5$ each do not produce the same overall count. In fact, the instance summation results in a distribution of counts, rather than a scalar, which overall reflects the actual uncertainty contained in the sample more accurately.

This work focuses on the latter stochastic approach to counting and its novel application to machine learning. This chapter formalizes this operation which will be referred to as *Poisson-binomial counting* from now on.

2.1 Model Definition

This section formalizes the two counting operations described above. Both can be characterized as bottom-up counting approaches (see Section 1.1.1) since they rely on the identification of individual instances to infer counts. In practice, these instances are drawn from an input sample $\mathbf{X}$ using a model f. To account for the uncertainty underlying the counting process, the model not only identifies the instances, but also also assigns to each of them a probability $p_i \in [0, 1]$ which reflects their respective likelihood of being relevant to the count, i.e.,

$$\{p_1, \ldots, p_N\} = f(\mathbf{X}) \in [0, 1]^N, \tag{2.1}$$

For example, these probabilities can correspond to bounding-boxes detection scores in object detection (Redmon et al., 2016; Hosang et al., 2017), pixel densities in density-based object counting (Zhang et al., 2015a; Idrees et al., 2018), and temporal event occurrence probabilities in audio detection (Schroeter et al.,

2019) (see Chapter 4). This general setting thus encompasses a wide array of applications.

Definition 2.1 (Deterministic Count)**.** As mentioned above, counting is commonly performed by summing the instance probabilities $\{p_1, \ldots, p_N\}$ (Zhang et al., 2015a; Idrees et al., 2018; Trott et al., 2018), i.e.,

$$C = \textstyle\sum_{i=1}^{N} p_i \,. \tag{2.2}$$

In this framework, the operation of counting is simply viewed as a deterministic sum of individual fractional contributions. While inferring scalar counts allows for a straightforward training of the model through mean squared error (MSE) or other standard regression-based loss functions, this approach does not account for the stochastic nature of the individual contributions. Indeed, for example, if someone has a probability $p{=}0.5$ of picking up a tomato, they do not actually cut the fruit in half and pick one of the two pieces ($C{=}0.5$), but rather either take the entire fruit or leave it on the shelf with equal probability. Similarly, instances in many domains are indivisible and come in integer amounts (e.g., object detection and human pose estimation). Thus, modeling counts as a sum of fractions of indivisible objects does not offer an effective solution for all counting applications.

Definition 2.2 (Poisson-Binomial Count)**.** In contrast, we propose modeling counts as a sum of independent Bernoulli trials $\mathcal{B}$ with probabilities $p_1, \ldots, p_N$, i.e.,

$$\mathscr{C} = \textstyle\sum_{i=1}^{N} \mathcal{B}(p_i) \,, \tag{2.3}$$

where the random variables $\mathcal{B}(p_i)$ are independent of one another. This approach better accounts for the stochastic nature of the individual instances. Indeed, since it is defined as a sum of distributions, the count is itself a distribution—rather than a scalar—thus allowing for a more accurate representation of the variability observed in the sample. For example, if someone has a probability $p{=}0.3$ of picking up a tomato, then there is a 30% chance that they will end up with 1 tomato and a 70% chance they will end up with 0. Thus, this stochastic approach accurately captures this effect, while the deterministic variant simply reports a single scalar value (i.e., $C = 0.3$) that does not retain any information about the underlying uncertainty. The properties of this counting model are discussed in Section 2.1.1 and Chapter 3.

Independence of the Bernoulli Random Variables

In many applications, the instance probabilities p_i are correlated. This is especially the case in temporal applications where the content of two consecutive timesteps are highly dependent. For instance, in piano music, if a note is played at a given time, it is almost certain that this same note will not be played again in the following few milliseconds, simply because of the mechanical and physiological constraints inherent to piano playing. Similar temporal constraints apply, for example, to the detection and counting of events in videos. Thus, in such settings, the predictions p_i in the sequence $p_1, \ldots, p_N$ cannot be independent of one another, if the temporal dynamics are to be modeled properly. The same observation also holds in the spatial domain (e.g., object counting in images) where features are often highly correlated locally. Although this modeling need for mutually dependent instances might appear to be in contradiction with the independence assumption of the Bernoulli random variables introduced in Definition 2.2, the assumption only requires the trial $\mathcal{B}(p_i)$ to be independent and not the success probabilities p_i themselves. Thus, the individual instances can be correlated through the model f without affecting the independence of the Bernoulli distributions; consequently, the individual probabilities p_i are not subject to any restriction.

Higher-order Generalization

In the proposed Poisson-binomial counting setup, each instance can only contribute a maximum of 1 to the count $\mathscr{C}$. While this assumption is met in numerous applications, the current framework might have to be generalized slightly in some very specific cases. For instance, in basketball, shots are awarded between 0 and 3 points depending on the situation (e.g., 0 for a miss or 1 for a successful free-throw). Modeling the total number of points scored during a game as a count using only sums of Bernoulli random variables is inconvenient, as it would require, among others, to model each potential point awarded as a different class. To offer an effective generalization of the setup to such higher-order count contributions, counts could be modeled as a sum of multinomial random variables instead. However, this extension of the proposed model is not developed further in this work.

2.1.1 Properties

Some properties of the Poisson-binomial counting model introduced in Definition 2.2 are derived in this section.

Property 2.1 (Poisson-Binomial Distribution). The count distribution, defined in Equation 2.3 as a sum of independent Bernoulli random variables with probabilities $p_1, \dots, p_N$, follows a well-known distribution: the Poisson-binomial distribution, i.e.,

$$\Pr(\mathscr{C} = k) = \sum_{S \in F_k} \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j), \tag{2.4}$$

Proof.

$$\begin{aligned}
\Pr(\mathscr{C} = k) &\overset{(2.3)}{=} \Pr(\textstyle\sum_i \mathcal{B}(p_i) = k) \\
&\overset{(\text{LTP})}{=} \sum_{S \in P(\{1,\dots,N\})} \Pr(\textstyle\sum_i \mathcal{B}(p_i) = k \mid \mathcal{B}(p_1) = \mathbb{1}_{1 \in S}, \dots, \mathcal{B}(p_N) = \mathbb{1}_{N \in S}) \\
&\qquad \cdot \Pr(\mathcal{B}(p_1) = \mathbb{1}_{1 \in S}, \dots, \mathcal{B}(p_N) = \mathbb{1}_{N \in S}) \\
&= \sum_{S \in P(\{1,\dots,N\})} \mathbb{1}_{|S|=k} \cdot \Pr(\mathcal{B}(p_1) = \mathbb{1}_{1 \in S}, \dots, \mathcal{B}(p_N) = \mathbb{1}_{N \in S}) \\
&= \sum_{S \in P(\{1,\dots,N\})} \mathbb{1}_{|S|=k} \cdot \prod_{i \in S} \Pr(\mathcal{B}(p_i) = 1) \prod_{j \in S^c} \Pr(\mathcal{B}(p_j) = 0) \\
&= \sum_{S \in P(\{1,\dots,N\})} \mathbb{1}_{|S|=k} \cdot \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j) \\
&= \sum_{S \in F_k} \mathbb{1}_{|S|=k} \cdot \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j) + \sum_{S \in F_k^c} \mathbb{1}_{|S|=k} \cdot \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j) \\
&= \sum_{S \in F_k} \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j)
\end{aligned} \tag{2.5}$$

In these equations, $P(\{1, \dots, N\})$ stands for the power set of $\{1, \dots, N\}$ (i.e., the set of all subsets) and S represents a set of indices sampled from $P(\{1, \dots, N\})$

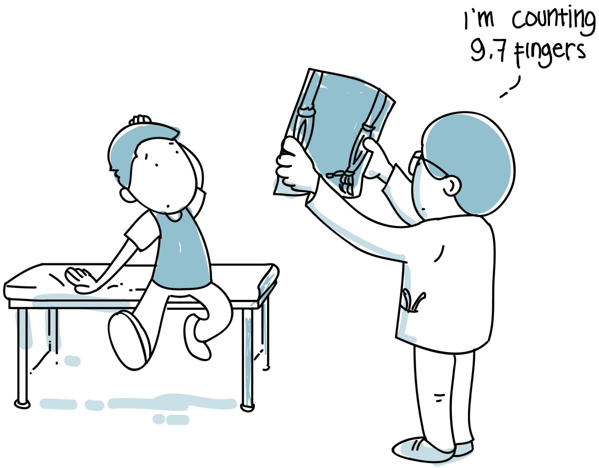


Figure 2.2: Deterministic counting in action. Expressing counts as a sum of fractional instance contributions leads to non-integer values, which do not model well the numerous applications where objects are indivisible and only appear in integer amounts.

that indicates which trials are equal to 1, while its complement S^c indicates which trials are equal to 0. Finally, F_k is the set of all subsets of $P(\{1,...,N\})$ of size k. The most important part of the derivation is the second equality which uses the law of total probability (LTP) on all possible realizations of the Bernoulli random variables. The rest follows seamlessly using the independence of the Bernoulli distribution and elementary set and probability theory. □

An important advantage of modeling counts as a discrete distribution, rather than a continuous scalar, is that only integer counts are inferred. Thus, in contrast to the classical approach to counting (Equation 2.2) which allows for the addition of fractional counts, the proposed approach allows for a natural modeling of counts for the numerous applications where objects are indivisible and only appear in integer amounts. Indeed, reporting that 10.3 people attended an event or that 9.7 fingers were detected on a radiography image could raise more questions than it actually answers (see Figure 2.2 for illustration).

Property 2.2 (Unambiguous Instances)**.**

$$\mathscr{C} = \textstyle\sum_i \mathcal{B}(p_i) = \sum_i p_i = C, \text{ if and only if } p_i \in \{0, 1\}. \tag{2.6}$$

The two approaches to counting (i.e., deterministic and Poisson-binomial) are equivalent in the absence of uncertainty since, in such a scenario, the Bernoulli random variables are identical to their respective success probabilities $p_i \in \{0, 1\}$.

Property 2.3 (Expectation)**.**

$$\mathbb{E}[\mathscr{C}] = \mathbb{E}[\textstyle\sum_i \mathcal{B}(p_i)] = \sum_i \mathbb{E}[\mathcal{B}(p_i)] = \sum_i p_i = C. \tag{2.7}$$

The classical approach to counting (Definition 2.1) can be replicated by taking the expectation either of the count or of the individual Bernoulli random variables, ultimately resulting in the loss of all information about the variability of the individual instances. Thus, the standard approach of optimizing the deterministic count can be viewed as a means to align the first moment of the underlying count distribution with the observed count. While discarding information about the higher moments of the count distribution might seem marginal in a context where the objective is to infer the most accurate count prediction, it actually has a significant impact on the learning process. Indeed, unlike its deterministic counterpart, the stochastic approach can rely on the additional prior information that, for instance, an annotated count of 4 corresponds to a distribution of count with all the mass on 4 (i.e., a Dirac distribution $\mathbb{1}_4$). Such a clear-cut count can thus only be the result of 4 unambiguous individual instances with probability 1. In contrast, in the deterministic framework, a count of 4 can be the result of infinitely many combinations of probabilities. Thus, by only considering the mean of the count distribution, all information about the Dirac nature of the label distribution is lost, and thus the predicted distribution can take any shape as long as the means are aligned. This implicit prior knowledge about the sparsity of the predictions is discussed in detail in Chapter 3 and stands at the core of this work.

2.2 Loss Function

In practice, the instance probabilities and the count can be inferred using a parameterized model $\hat{f}_\theta$ with parameter set θ:

$$\begin{aligned} \{\hat{p}_{\theta,1}, \ldots, \hat{p}_{\theta,N}\} &= \hat{f}_\theta(\mathbf{X}) \in [0,1]^N \\ \hat{\mathscr{C}}_\theta &= \textstyle\sum_i \mathcal{B}(\hat{p}_{\theta,i})\,. \end{aligned} \tag{2.8}$$

In order to train such a model, a loss function has to be defined that compares the estimated count $\hat{\mathscr{C}}_\theta$ with the observed count c. By definition, the count distribution $\hat{\mathscr{C}}_\theta$ follows a Poisson-binomial distribution. Thus, the estimation of the parameter set θ can be done by comparing the distribution $\Pr(\hat{\mathscr{C}}_\theta = k \,|\, \mathbf{X})$ to the target sample distribution determined by c (i.e., $\mathbb{1}_c$). The Kullback-Leibler divergence (1951), which in this specific case corresponds to the cross-entropy and max-likelihood, is therefore a suitable choice for the loss function:

$$\begin{aligned} L(\theta) &= D_{KL}(\mathbb{1}_c \| \textstyle\sum_i \mathcal{B}(\hat{p}_{\theta,i})) \\ &= -\textstyle\sum_j \Pr(\mathbb{1}_{c=j}) \log\left(\frac{\Pr(\sum_i \mathcal{B}(\hat{p}_{\theta,i})=j)}{\Pr(\mathbb{1}_{c=j})}\right) \\ &= -\log\left(\Pr(\textstyle\sum_i \mathcal{B}(\hat{p}_{\theta,i}) = c \mid \mathbf{X})\right) \\ &= -\log\left(\Pr\left(\hat{\mathscr{C}}_\theta = c \mid \mathbf{X}\right)\right). \end{aligned} \tag{2.9}$$

For the sake of notation simplicity, only one sample was considered in the previous equation. However, the extension of this loss function to multiple samples with count estimates $\hat{\mathscr{C}}_\theta^{(i)}$ and count labels $c^{(i)}$ is straightforward:

$$L(\theta) = \textstyle\sum_i -\log\left(\Pr\left(\hat{\mathscr{C}}_\theta^{(i)} = c^{(i)} \mid \mathbf{X}_i\right)\right). \tag{2.10}$$

A similar extension can be done for applications with multiple prediction classes (i.e., multivariate counts).

Structured Classification

As mentioned in Section 1.1.1, the proposed method is a classification approach where the probability assigned to each count class is not directly inferred—unlike direct classification approaches, but rather explicitly given by the individual instance probability estimates and the underlying count model (i.e., Poisson-binomial counting). Thus, in contrast to standard count classification models, this approach does not only implicitly take into account the natural ordering of the count classes, but also allows to leverage finer-grained instance annotations to learn to count. Nevertheless, as with any classification model, the training can be done using the standard cross-entropy between the count distributions and the count labels, which results in Equation 2.9.

Limitations and Opportunity

The Poisson-binomial loss function (Equation 2.9) is defined as the comparison of a count distribution and a *scalar* label—via its corresponding Dirac distribution. This definition is reminiscent of the classical use of the cross-entropy in classification tasks (e.g., image classification). Over the years, it has however been shown that comparing distributions (over all classes) with variance-free one-hot encoded labels—without any additional loss correction—is highly ineffective in modeling class probabilities in uncertain settings (Mnih & Hinton, 2012; Natarajan et al., 2013; Reed et al., 2014; Azadi et al., 2016). By extension, the proposed Poisson-binomial loss function (as defined in Equation 2.9) is thus expected not to fully capture the variability of the count distribution in uncertain settings.

Additionally, while the counting model underlying the Poisson-binomial loss function is probabilistic, the loss function actually encourages the model to output a count distribution that matches the nature of the label: a variance-free count. This sparsity effect (i.e., the optimal prediction consists in assigning all the probability mass to a single bin) might make our approach less effective in modeling variability in the count than other standard probabilistic counting models, e.g., Poisson-regression (Chan & Vasconcelos, 2009; Fallah et al., 2009). Indeed, such approaches are explicitly designed to infer spread-out count distributions.

This work will however show that, while the above properties might appear to be limitations of the model, they are actually a blessing in disguise in many circumstances. In fact, as will be demonstrated in Chapter 3, the sparsity effect of the loss function can be leveraged to learn the non-differentiable function of instance selection in a differentiable manner.

2.3 Implementation Details

Notation $\mathcal{C}_n[k] := \Pr\big(\mathcal{C}_n = k\big)$, $\mathcal{C}_n := \sum_{j \leq n} \mathcal{B}(p_j)$

$\mathcal{C}_n[k]$ corresponds to the probability of having the partial count $\mathcal{C}_n$ (when only considering the first n instances $p_1, \ldots, p_n$) be equal to k. While this definition assumes some ordering of the instances, the ordering itself has no impact on the final count estimate $\mathcal{C}_N$ as the Bernoulli random variables are assumed to be independent; an ordering is only needed for practical purposes.

2.3.1 Loss Computation

The Poisson-binomial distribution is the core component of the loss function defined in the previous section; its efficient and accurate computation is thus crucial for the learning process. While being a straightforward option, the closed-form definition (Equation 2.4) can only be computed efficiently when the number of point predictions p_i is limited. Indeed, given the exponential nature of its complexity, this approach becomes too computationally expensive for most practical applications.

Fortunately, numerous solutions have been proposed over the years to overcome this specific issue (Le Cam, 1960; Roos, 2001; Fernández & Williams, 2010; Howard, 1972; Shah, 1973; Gail et al., 1981; Chen et al., 1994; Chen & Liu, 1997). However, while approximation-based methods (Le Cam, 1960; Roos, 2001) can be very efficient, they are inappropriate in this case since an exact computation of the loss is imperative for gradient descent learning. Additionally, Fourier-based closed-form formulas (Fernández & Williams, 2010) can also be directly discarded from consideration as they are too complex and might be too unstable for the loss computation. Thus, recursive formulas are the only viable alternative (Howard, 1972; Shah, 1973; Gail et al., 1981; Chen et al., 1994; Chen & Liu, 1997).

As a compromise between simplicity, numerical stability, and computational complexity ($O(N^2)$), the following recursive formula—which is in fact a special case of the more general recursion proposed in (Howard, 1972; Gail et al., 1981)—is preferred in this work:

Property 2.4 (Recursion Property)**.**

$$\mathcal{C}_n[k] = \begin{cases} (1-p_n)\,\mathcal{C}_{n-1}[k] & k=0 \\ (1-p_n)\,\mathcal{C}_{n-1}[k] + p_n\mathcal{C}_{n-1}[k-1] & k>0, \end{cases} \tag{2.11}$$

where $\mathcal{C}_0[k] = \mathbb{1}_{k=0}$.

This recursive formula can easily be derived using the law of total probability, the independence assumption of the Bernoulli distribution, and the definition of the Poisson-binomial distribution:

Proof.

$$\begin{aligned} \mathcal{C}_n[k] &:= \Pr\Big(\sum_{j\le n}\mathcal{B}(p_j) = k\Big) \\ &\overset{\text{(LTP)}}{=} \Pr\Big(\sum_{j\le n}\mathcal{B}(p_j) = k \mid \mathcal{B}(p_n) = 1\Big)\Pr\Big(\mathcal{B}(p_n) = 1\Big) \\ &\qquad + \Pr\Big(\sum_{j\le n}\mathcal{B}(p_j) = k \mid \mathcal{B}(p_n) = 0\Big)\Pr\Big(\mathcal{B}(p_n) = 0\Big) \\ &= \Pr\Big(\sum_{j\le n}\mathcal{B}(p_j) = k \mid \mathcal{B}(p_n) = 1\Big)p_n \\ &\qquad + \Pr\Big(\sum_{j\le n}\mathcal{B}(p_j) = k \mid \mathcal{B}(p_n) = 0\Big)(1-p_n) \\ &= \Pr\Big(\sum_{j\le n-1}\mathcal{B}(p_j) = k-1\Big)p_n + \Pr\Big(\sum_{j\le n-1}\mathcal{B}(p_j) = k\Big)(1-p_n) \\ &= p_n\mathcal{C}_{n-1}[k-1] + (1-p_n)\,\mathcal{C}_{n-1}[k]. \end{aligned} \tag{2.12}$$

The initial condition $\mathcal{C}_0[k] = \mathbb{1}_{k=0}$ comes from the fact that the probability of having a count of zero when observing no instances is trivially equal to 1. □

2.3.2 Mass Truncation

The extent of the count distribution's ($\mathcal{C}_i$) sample space is naturally bounded by i. Indeed, since each individual instance $\hat{p}_i$ can only increase the count by at most 1, the range of values that the count distribution $\mathcal{C}_i$ can possibly take spans from 0 to i (see definition 2.3). From a practical standpoint, it can however be beneficial to impose a single stricter bound ($c_{\max}$) on the distribution. Such a threshold allows to truncate the count distribution after the first $c_{\max}$+1 bins and put all the remaining mass above the threshold—i.e., $\Pr(\mathcal{C}_i > c_{\max})$—on the last bin, i.e.,

$$\bar{\mathcal{C}}_N[k] := \begin{cases} \mathcal{C}_N[k] & k \leq c_{\max} \\ \sum_{j>c_{\max}} \mathcal{C}_N[j] & k = c_{\max}+1. \end{cases} \tag{2.13}$$

Such truncation of the count distribution has the main advantage of reducing the complexity of the loss computation from $O(N_i^2)$ to $O(c_{\max} N_i)$. Additionally, having a unique and harmonized count distribution length (i.e., $c_{\max}+2$) regardless of the number of individual instances facilitates the implementation of the loss.

In practice, imposing a stricter bound is often straightforward as most applications are already subject to natural restrictions. For instance, the number of notes played within a second of piano music is bounded by physiological and mechanical constraints, while the number of yellow cards received during a football game is limited by the rules of the game. In these scenarios, it would be highly sub-optimal to compute the probability assigned to each count up to the number of sound samples in the audio sequence (44100 samples per second is common in music) or the number of frames in a football game video respectively.

2.3.3 Weight Initialization

The multiplicative nature of the Poisson-binomial distribution calls for caution when initializing the network's weights. Indeed, it is essential to avoid extreme $\hat{p}_i$ values that may cause cross-entropy surges and exploding gradients. For instance, assuming both that the predictions are obtained via a sigmoid transform of the network's output and that all weights are set to zero (an initialization that is often far from optimal), then $\hat{p}_i = 0.5$, $\forall i \leq N$. Consequently, in such a case, the probability mass of the Poisson-binomial distribution that lies on count zero (i.e., no event occurrence) is equal to $\Pr(\mathcal{C}_N = 0) = 0.5^N$. In most applications, the number of probability estimates can be quite large, which would lead to almost zero mass on the first bin of the count distribution. This can cause significant computational issues when computing the loss function or performing backpropagation (e.g., exploding gradients) if the true count is actually equal to zero.

A simple solution consists in initializing the bias of the final prediction layer in such a way that the count distribution contains a reasonable amount of mass on each of its bins. For instance, one can define an appropriate amount of mass that should initially lie on a count of zero (ω_0), and then the initial corresponding bias (β_0) can be computed analytically:

$$\beta_0 = -\log\left(1 - \frac{1}{\omega_0^N}\right) \tag{2.14}$$

By adding such a bias to any traditional initialization method such as Xavier (Glorot & Bengio, 2010) or He (2016), the count distribution $\mathscr{C} = \mathcal{C}_N$ is ensured to have some non-negligible amount on each of its potential outcomes. This simple operation thus significantly reduces the risk of exploding gradients or loss surges caused by an improper initialization.

2.3.4 Optimization on a Subset of Predictions

While the truncation of the count distribution (Equation 2.11) allows for a more efficient computation of the recursive formula, the loss computation—with a complexity of $O(c_{\max}N_i)$—can still be computationally intensive. Indeed, when working on a small network with numerous output probabilities p_i, the loss computation itself can become a computational bottleneck. Thus, in order to avoid this issue, the probabilities p_i that do not impact the learning in any significant way can be discarded from the optimization. More specifically, in the case that some p_i converge towards zero (see Chapter 3 for a full discussion about the prediction sparsity property of the loss function), the impact of these predictions on the overall loss function and by extension on the learning process becomes negligible. Therefore, these predictions can be removed from the loss computation and from the gradient computation without any effect on the training. Overall, while this trick cannot be applied at the early stages of the training, discarding the irrelevant p_i can lead to significant gains in efficiency in the later stages, and thus on the training in general.

2.4 Conclusion

This chapter proposes Poisson-binomial counting (Definition 2.2) as an alternative to the standard deterministic counting framework (Definition 2.1). One of the main advantages of this approach resides in its ability to better account for the stochastic nature of the individual instances that are being counted. Indeed, while both methods are equivalent when counting unambiguous instances (Property 2.2), modeling counts as random variables, rather than a scalar, preserves more information about the underlying instances in the presence of uncertainty. In fact, the deterministic approach to counting only conserves the first moment of the count distribution, discarding all information about the higher moments (Property 2.3).

As a consequence, the resulting Poisson-binomial loss function (Equation 2.9) offers a more tailored supervision that not only directs models to aligns expected counts, but that ensures that the higher moments of the count label are taken into account. The next chapter actually shows how this more constrained supervision implicitly integrates the prior knowledge about the sparsity of the underlying instance into the learning process.

Chapter 3

Prediction Sparsity Properties

The Poisson-binomial counting model introduced in the previous chapter is not only a bottom-up distribution-based approach to infer counts, but it is also a means to achieve *prediction sparsity* through count supervision. This chapter covers, in detail, how learning to count instances using the Poisson-binomial model can implicitly lead to sparse probability predictions.

3.1 Intuition behind Prediction Sparsity

3.1.1 Matching Distributions Rather Than Scalar Expectations

The Poisson-binomial loss function (Equation 2.9) encourages models to align the count distribution $\hat{\mathscr{C}}$—uniquely determined by the individual instance probability estimates $\hat{p}_1, \ldots, \hat{p}_N$—with the label count distribution $\mathbb{1}_c$ (i.e., a Dirac distribution). The overall objective is thus not only to maximize the amount of mass on the bin corresponding to a count of c, but also to minimize the probabilities attributed to all other counts. This convergence towards a variance-free count distribution has a direct impact on the underlying probability estimates $\hat{p}_1, \ldots, \hat{p}_N$. Indeed, the variance of the Poisson-binomial distribution is given by:

$$\sigma^2(\hat{\mathscr{C}}) = \textstyle\sum_i (1 - \hat{p}_i)\hat{p}_i. \tag{3.1}$$

In fact, in order to closely match the variance of the label distribution (i.e., $\sigma^2(\mathbb{1}_\mathbf{c}) = 0$), the probability estimates are bound to converge towards the $0, 1$ extremes as the learning progresses, i.e.,

$$\sigma^2(\hat{\mathscr{C}}) \equiv \sigma^2(\mathbb{1}_\mathbf{c}) = 0 \iff \hat{p}_i \in \{0, 1\},\ \forall i \leq N. \tag{3.2}$$

Thus, this prediction sparsity property emerges naturally from constraining the estimated count distribution to take the form of a Poisson-binomial distribution.

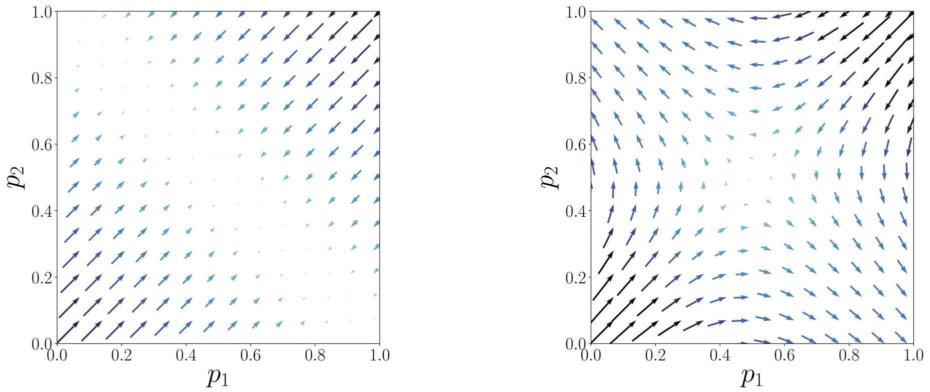
In contrast, as highlighted by Property 2.3, standard counting approaches only encourage models to align the expectation of the estimated count distribution with the label count c. In such a framework, the higher moments of the distribution are not taken into account, and thus, as long as the first moments are aligned, the count distribution can take any form. This lack of implicit restriction on the shape of the estimated distribution—especially on the variance, results in a training objective that, while being suitable for numerous counting applications, only loosely captures the complexity of the learning target.

Overall, training with the Poisson-binomial loss function, rather than traditional counting loss functions, allows to implicitly incorporate into the learning process the common prior that counts are the sum of sparse integer contributions.

3.1.2 Toy Example: Bounding-Box Selection

Object detection methods often heavily rely on non-maximum suppression to select a small set of relevant bounding-boxes per sample. While existing algorithms often use fixed heuristics (Girshick, 2015; Ren et al., 2015; Redmon et al., 2016; Liu et al., 2016) for this selection process, trainable methods also exist. For instance, Hosang et al. (2017) propose to rescore the original detection scores in an attempt to obtain only a few high-probability outputs, rather than numerous scattered predictions. In such a setting, as the optimal solution of the selection process requires the number of selected bounding-boxes to match the number of labeled objects in the image, object counting could be leveraged as an additional means of supervision. This section assesses the impact of adding a counting-based loss function as a regularizer to the rescoring process on a simple two-instance example.

In this example, for the sake of simplicity, it is assumed that the detection model outputs exactly two bounding-boxes (with detection scores $\hat{p}_1$ and $\hat{p}_2$ respectively) and that the number of ground-truth bounding-boxes is equal to 1 (i.e., $c = 1$). Thus, if considering the detection scores only, a rescoring scheme would be optimal in this setup only if it yields exactly one high-probability bounding-box while discarding the other (i.e., $(\hat{p}_1, \hat{p}_2) \to (0, 1)$ or $(\hat{p}_1, \hat{p}_2) \to (1, 0)$).



(a) Deterministic Counting

(b) Poisson-Binomial Counting

Figure 3.1: Gradients of (a) the deterministic counting loss function (L_{MSE}) and (b) the Poisson-binomial loss function (L_{PB}) with respect to instance probability estimates $\hat{p}_1, \hat{p}_2$ with a label count of $c=1$.

The following evaluates whether the integration of deterministic or Poisson-binomial counting-based supervision helps the training converge towards these optimal probability assignments.

Deterministic Counting

The loss function most commonly associated with the standard deterministic approach to counting is the (root) mean squared error between the predicted and the label count which, in this example—with two instances and a target count of 1—results in the following loss:

$$L_{\text{MSE}} = \text{MSE}\Big(\hat{C}, 1\Big) = (\hat{p}_1 + \hat{p}_2 - 1)^2. \tag{3.3}$$

This loss function is minimized if and only if

$$\hat{p}_1 + \hat{p}_2 = 1. \tag{3.4}$$

Thus, any combination of $\hat{p}_1, \hat{p}_2 \in [0,1]$ that satisfies this condition is a stable solution to the minimization of L_{MSE}. This elementary observation is visually confirmed by Figure 3.1a which displays the gradient of the loss function with respect to the probability estimates $\hat{p}_1, \hat{p}_2$. This loss function does therefore not converge the probability estimates towards the overall objective of the rescoring scheme which consists in outputting only one single high-probability detection—either $(\hat{p}_1, \hat{p}_2) \rightarrow (0,1)$ or $(\hat{p}_1, \hat{p}_2) \rightarrow (1,0)$—in this scenario. Incorporating L_{MSE} as a count-based regularization would thus not benefit the learning of bounding-box selection as it has no impact on the sparsity of the predictions.

Poisson-Binomial Counting

In contrast, in the same scenario, the proposed Poisson-binomial loss function (Equation 2.9), i.e.,

$$\begin{aligned}
L_{\text{PB}} &= -\log\Big(\Pr\Big(\hat{\mathscr{C}}=1\Big)\Big) \\
&= -\log\Big(\Pr\Big(\big(\mathcal{B}(\hat{p}_1)=1 \wedge \mathcal{B}(\hat{p}_2)=0\big) \vee \big(\mathcal{B}(\hat{p}_1)=0 \wedge \mathcal{B}(\hat{p}_2)=1\big)\Big)\Big) \\
&= -\log\Big(\Pr\Big(\mathcal{B}(\hat{p}_1)=1, \mathcal{B}(\hat{p}_2)=0\Big) + \Pr\Big(\mathcal{B}(\hat{p}_1)=0, \mathcal{B}(\hat{p}_2)=1\Big)\Big) \\
&\overset{(\text{ind})}{=} -\log\Big(\Pr\Big(\mathcal{B}(\hat{p}_1)=1\Big) \cdot \Pr\Big(\mathcal{B}(\hat{p}_2)=0\Big) + \Pr\Big(\mathcal{B}(\hat{p}_1)=0\Big) \cdot \Pr\Big(\mathcal{B}(\hat{p}_2)=1\Big)\Big) \\
&\overset{(\text{def})}{=} -\log\Big(\hat{p}_1\,(1-\hat{p}_2) + (1-\hat{p}_1)\,\hat{p}_2\Big)
\end{aligned} \tag{3.5}$$

is minimized if and only if $(\hat{p}_1, \hat{p}_2) = (0,1)$ or $(\hat{p}_1, \hat{p}_2) = (1,0)$. Figure 3.1b confirms that the probability estimates converge towards clear-cut values when minimizing L_{PB}. Indeed, the gradients of the loss with respect to the detection probabilities $\hat{p}_1, \hat{p}_2$ show that the estimates are drawn to either one of the two extremes as the learning progresses. While a saddle point exists at $(\hat{p}_1, \hat{p}_2) = (1/2, 1/2)$, this solution is highly unstable and any deviation from it will cause the predictions to converge towards one of the two global minima. The likelihood of being stuck around that point, like the Buridan's ass between the two stacks of hay, is highly unlikely given the stochastic nature of the training process.

Thus, in contrast to the classical count-based loss function, adding the Poisson-binomial loss function as an additional optimization target would certainly benefit the convergence of the detection scores towards the optimal sparse solutions.

Remark

In this example, the counting loss function itself does not take into account the content, nor the context of the bounding-boxes, and thus has to be leveraged in conjunction with another loss function in order to successfully select relevant bounding-boxes. The global minimum towards which the predictions will be drawn highly depends on that other loss function. Overall, in this scenario, the Poisson-binomial counting loss function only acts as a means of achieving predictions sparsity, and not as a means of making meaningful selections.

Higher Dimensions

As the number of instances increases, training with a classical counting loss function might result in highly dispersed predictions. Indeed, a contribution of 1 to the global count could be split into numerous small instance contributions, thus leading to uninterpretable results. In contrast, the ability of the Poisson-binomial loss function to converge the probability estimates towards sparsity remains in higher dimensions (see the next section).

3.2 Prediction Sparsity

The sparsity-inducing capability of the Poisson-binomial loss function is addressed more formally in this section.

3.2.1 Global Minima of the Loss Function

At the minima of the loss function, the predictions $\hat{\mathbf{p}}_\theta$ are sparse. More precisely,

> **Theorem 3.1** (Count Sparsity). For all $c \in \mathbb{N}$,
>
> $$D_{KL}(\mathbb{1}_c \| \textstyle\sum_i \mathcal{B}(\hat{p}_{\theta,i})) = 0 \iff (\|\hat{\mathbf{p}}_\theta\|_0 = c) \wedge (\hat{\mathbf{p}}_\theta \in \{0,1\}^N), \tag{3.6}$$

where $\|\cdot\|_0$ corresponds to the ℓ_0-norm, which actually counts the number of instances that are not equal to zero.

Proof. The equivalence in Theorem 3.1 (i.e., $P \iff Q$) can be proven by showing that each conditional ($P \Longrightarrow Q$ and $Q \Longrightarrow P$) holds:

$\Longrightarrow$

$$\begin{aligned} & D_{KL}(\mathbb{1}_c \| \textstyle\sum_i \mathcal{B}(\hat{p}_{\theta,i})) = 0 \\ & \quad \Longrightarrow \begin{cases} \mu(\mathbb{1}_c) = \mu(\sum_i \mathcal{B}(\hat{p}_{\theta,i})) \\ \sigma^2(\mathbb{1}_c) = \sigma^2(\sum_i \mathcal{B}(\hat{p}_{\theta,i})) \end{cases} \\ & \quad \Longleftrightarrow \begin{cases} c = \sum_i \hat{p}_{\theta,i} \\ 0 = \sum_i (1 - \hat{p}_{\theta,i}) \hat{p}_{\theta,i} \end{cases} \\ & \quad \Longleftrightarrow \begin{cases} c = \sum_i \hat{p}_{\theta,i} \\ \hat{p}_{\theta,i} \in \{0,1\} \end{cases} \\ & \quad \Longrightarrow (\|\hat{\mathbf{p}}_\theta\|_0 = c) \wedge (\hat{\mathbf{p}}_\theta \in \{0,1\}^N) \end{aligned} \tag{3.7}$$

$\Longleftarrow$

$$\begin{aligned} & (\|\hat{\mathbf{p}}_\theta\|_0 = c) \wedge (\hat{\mathbf{p}}_\theta \in \{0,1\}^N) \\ & \quad \Longrightarrow \textstyle\sum_i \mathcal{B}(\hat{p}_{\theta,i}) = \sum_{\{i | \hat{p}_{\theta,i} = 1\}} \mathbb{1}_1 + \sum_{\{i | \hat{p}_{\theta,i} = 0\}} \mathbb{1}_0 \\ & \qquad\qquad\qquad\quad = \textstyle\sum_{i=1}^{c} \mathbb{1}_1 = \mathbb{1}_c \end{aligned} \tag{3.8}$$

□

This theorem highlights two important characteristics of the loss function: its ability to control the number of non-zero instances and the normalized nature of its solutions.

Controlled Sparsity

The first condition on the right-hand side of the equivalence (i.e., $\|\hat{\mathbf{p}}_\theta\|_0 = c$) indicates that the number of non-zero predictions can be controlled through the loss function. While this feature is intuitive—since the loss function is defined as a comparison of count distributions, this ability to enforce a strict number of non-zero predictions sets Poisson-binomial counting apart from several other traditional means of achieving prediction sparsity. Indeed, for instance, non-maximum suppression in object detection and sparse activation functions (Martins & Astudillo, 2016; Martins & Kreutzer, 2017; Malaviya et al., 2018), even though they guarantee a sparse selection of instances, do not allow for such a level of supervision.

Nature of the Predictions

The second term on the right-hand side of the equivalence (i.e., $\hat{\mathbf{p}}_\theta \in \{0,1\}^N$) highlights one of the main underlying assumptions of the model: the optimal non-zero predictions must be equal to 1. This contrasts with several classical sparsity-inducing approaches, which only take into consideration the number of non-zero instances regardless of their value. While often trivially met in applications dealing with probability assignments (e.g., bounding-box selection, existence probability, and class probability), this condition requires careful attention in other settings. For instance, in contrast to the ℓ_1-regularization (i.e., Lasso (Tibshirani, 1996)), the Poisson-binomial loss function cannot directly be used to achieve parameter sparsity in linear regression as the optimal non-zero weights are not necessarily equal to 1.

3.2.2 Local Minima of the Loss Function

While Theorem 3.1 demonstrates that prediction sparsity is a necessary condition for the global minimization of the loss function, it does not show that learning to count instances with the Poisson-binomial loss function implicitly converges the probability estimates towards these global minima. Indeed, gradient-based optimization algorithms are prone to being drawn to the *local* minima of the loss function, and therefore do not necessarily converge towards the *global* minima.

However, despite not being convex with respect to the instance probabilities, the Poisson-binomial loss function implicitly has a property that is important for its successful optimization through gradient-based learning: the set of local minima of the loss function (w.r.t. the instance probabilities) is equivalent to the set of global minima of the loss function, i.e.,

Theorem 3.2 (Local Minima). Let $l(\mathbf{x}) := D_{KL}(\mathbb{1}_c \| \sum_i \mathcal{B}(x_i))$, then $\forall c \leq N$

$$\Big\{\mathbf{p} = \{p_1, \ldots, p_N\} \in [0,1]^N \mid \mathbf{p} \text{ is a local minimum of } l(\mathbf{x})\Big\}$$
$$\equiv \Big\{\mathbf{p} = \{p_1, \ldots, p_N\} \in [0,1]^N \mid l(\mathbf{p}) = \underbrace{D_{KL}(\mathbb{1}_c \| \textstyle\sum_i \mathcal{B}(p_i)) = 0}_{\text{Global Minimum}}\Big\} \tag{3.9}$$

Proof. The complete proof can be found in Appendix A. In summary, the main idea of the proof consists in showing that the function

$$h(\mathbf{x}) := \Pr\Big(\textstyle\sum_i \mathcal{B}(x_i) = c \Big) \tag{3.10}$$

has no local maxima in $(0,1)^N$ by proving that its Hessian $\mathbf{H}h(\mathbf{p})$ is not negative-definite at any point $\mathbf{p} \in (0,1)^N$ for all values of $c \leq N$. Once this is proven, it directly follows that the local maxima of $h(\mathbf{p})$ can only lie at the border of the bounded interval $[0,1]^N$ and it can further be shown that the maxima only lie at the corners of the hypercube (i.e., $\mathbf{p} \in \{0,1\}^N$). Finally, using the definition of the loss function, it can easily be demonstrated that only the corners that satisfy the global maximality criterion $D_{KL}(\mathbb{1}_c \| \sum_i \mathcal{B}(p_i))$ are local minima of the loss function.

Thus, the theorem follows from the fact that the Poisson-binomial loss function is locally minimized in $\mathbf{p}$ if $h(\mathbf{p})$ is locally maximized in $\mathbf{p}$ since the log is a strictly monotonically increasing function on the $(0,1]$ interval. □

Other Critical Points

As mentioned in Section 3.1.2 and as reflected in Figure 3.1b, the Poisson-binomial loss function can present saddle points. In practice, given the stochastic nature of batch-based optimization and given the instability of these critical points, it is highly unlikely that the optimization might get stuck there. These points can nevertheless slow down the learning process since the gradients are relatively small around them. This detrimental effect can however be alleviated by adapting the learning rate accordingly.

There also exist points on the boundary of the domain (e.g., $\mathbf{p} \in \{0,1\}^N$) that are not local minima but whose individual gradients are all equal to zero. For instance, if $\mathbf{p} = 0$ and if the labeled count c is strictly greater than 1, the derivative of the loss with respect to the individual instance probabilities is equal to zero[1], i.e., $\frac{\partial}{\partial p_i} l(\mathbf{p}) = 0$. However, despite the value of the individual gradients, this point does not constitute a local minimum of the loss since, for example, the point $\mathbf{p}' = 0 + \epsilon$ for any $0 < \epsilon < 1$ yields a strictly smaller loss (see full discussion in Appendix A). These shortcomings can thus be overcome either by selecting an

[1] Note that in order to avoid cases where $\log(0) = -\infty$, a small value can be added to each bin of the count distribution in the loss computation without significantly affecting its optimization.

appropriate optimization algorithm or simply by setting the initialization away from these critical points (i.e., away from the strict border of the domain). In practice, if the probability estimates are the result of a sigmoid or a softmax activation function, the instance probabilities implicitly cannot take the value 0 or 1, and thus the problem is non-existent.

Consequence

In conclusion, since the loss function is continuously differentiable over the entire domain, the optimization of the Poisson-binomial loss function is bound to converge towards one of the global minima. This result combined with Theorem 3.1 proves that learning to count instances with the Poisson-binomial loss theoretically converges the individual instance probabilities towards sparsity.

3.2.3 Mode Properties of the Poisson-Binomial Distribution

Additional insights into the convergence of the Poisson-binomial loss function and the instance probabilities can be derived by investigating the mode of the count distribution. Thus, as a preliminary step towards general convergence results, a few important properties about the mass assigned to the mode of the Poisson-binomial distribution are derived in this section.

Recall $\mathcal{C}_n[k] := \Pr\left(\mathcal{C}_n = k\right)$, $\mathcal{C}_n := \sum_{j \leq n} \mathcal{B}(p_j)$

$\mathcal{C}_n[k]$ is defined as the probability that the partial count $\mathcal{C}_n$ (when only considering the first n instances) is equal to k. Thus, the mode of the partial count distribution $\mathcal{C}_n$ is given by $\arg\max_k \ \mathcal{C}_n[k]$ and its probability is equal to $\mathcal{C}_n[\arg\max_k \ \mathcal{C}_n[k]] = \max_k \ \mathcal{C}_n[k]$.

First, the probability $\max_k \ \mathcal{C}_n[k]$ assigned to the mode of the partial count distribution $\mathcal{C}_n[k]$ is a monotonically decreasing sequence in n:

Property 3.1 (Decreasing Maximum)**.**

$$\max\nolimits_k \ \mathcal{C}_n[k] \leq \max\nolimits_k \ \mathcal{C}_{n-1}[k], \tag{3.11}$$

Proof. This property can easily be proven by recalling the recursive formula derived from both the law of total probability and the definition of the Poisson-binomial distribution (Equation 2.11) and by defining, without loss of generality, $\mathcal{C}_n[-1] = 0$:

$$\begin{aligned}
&\mathcal{C}_n[k] \overset{2.11}{=} (1-p_n)\,\mathcal{C}_{n-1}[k] + p_n\mathcal{C}_{n-1}[k-1],\ \forall k \\
\Longrightarrow\ &\mathcal{C}_n[j] = (1-p_n)\,\mathcal{C}_{n-1}[j] + p_n\mathcal{C}_{n-1}[j-1],\ j = \arg\max_i\ \mathcal{C}_n[i] \\
\Longleftrightarrow\ &\max_k \mathcal{C}_n[k] = (1-p_n)\,\mathcal{C}_{n-1}[j] + p_n\mathcal{C}_{n-1}[j-1],\ j = \arg\max_i\ \mathcal{C}_n[i] \qquad (3.12) \\
\overset{p_n \geq 0}{\Longrightarrow}\ &\max_k \mathcal{C}_n[k] \leq (1-p_n)\max_k\ \mathcal{C}_{n-1}[k] + p_n \max_k\ \mathcal{C}_{n-1}[k] \\
&\qquad\qquad = \max_k\ \mathcal{C}_{n-1}[k].
\end{aligned}$$

□

This property reveals that once the mass assigned to the mode of $\mathscr{C}$ is reduced, it cannot be increased back. (Note that the probability assigned to the mode decreases, not the location of the mode itself, which can only increase over time, i.e., $\arg\max_k\ \mathcal{C}_{n+1}[k] \geq \arg\max_k\ \mathcal{C}_n[k]$.) In other words, any instance probability that has not yet been included in the count can only cause the probability of the mode to decrease. In fact, this feature comes in tandem with the observation that the variance of $\mathcal{C}_n[k]$ is non-decreasing with n:

$$\sigma^2(\mathcal{C}_n[k]) - \sigma^2(\mathcal{C}_{n-1}[k]) = (1-p_n)p_n \geq 0. \tag{3.13}$$

Indeed, once the mass of the count distribution has been dispersed, it cannot be reconcentrated back.

Lemma 3.1 (First Upper Bound)**.**

$$\max_k \mathcal{C}_n[k] \leq \tfrac{1}{2} + \min_{j\leq n} \|\tfrac{1}{2} - p_j\|. \tag{3.14}$$

Proof.

$$\begin{aligned}
\mathcal{C}_n &\overset{\text{def}}{=} \sum_{j\leq n} \mathcal{B}(p_j) \\
&\overset{\text{ind.}}{=} \mathcal{B}(p_i) + \sum_{i\neq j\leq n} \mathcal{B}(p_j)\,,\ i := \arg\min_{j\leq n} \|\tfrac{1}{2} - p_j\| \\
\overset{3.1}{\Longrightarrow}\ &\max_k \mathcal{C}_n[k] \leq \max_k \mathcal{B}(p_i)\,[k],\ i := \arg\min_{j\leq n} \|\tfrac{1}{2} - p_j\|
\end{aligned}$$

$$
\begin{aligned}
&= \max\{p_i, 1-p_i\},\ i := \operatorname*{arg\,min}_{j\leq n} \|\tfrac{1}{2} - p_j\| \\
&= \tfrac{1}{2} + \min_{j\leq n} \|\tfrac{1}{2} - p_j\|. \qquad (3.15)
\end{aligned}
$$

□

This inequality highlights that even a single prediction $\hat{p}_i$ around $\frac{1}{2}$ can cause the probability assigned to the mode of $\mathscr{C}$ to drop permanently. In fact, this bound (Equation 3.14) is extremely loose as it only stems from a single $\hat{p}_i$; according to the decreasing maximum property (Equation 3.11), all remaining $\hat{p}_i$ can only reinforce this effect.

3.2.4 Convergence of Instance Probabilities

These first results about the mode of the Poisson-binomial distribution can appear abstract. However, the connection between distribution upper-bounds and prediction convergence becomes more evident once the definition of the Poisson-binomial loss function is restated:

$$
\begin{aligned}
L(\theta) &= -\log\left(\Pr\left(\hat{\mathscr{C}}_\theta = c \mid \mathbf{X}\right)\right) \\
&= -\sum_i \log\left(\hat{\mathcal{C}}_N[\mathbf{c}]\right) \qquad (3.16) \\
&\overset{(3.14)}{\geq} -\log\left(\tfrac{1}{2} + \min_k \|\tfrac{1}{2} - \hat{p}_{\theta,k}\|\right).
\end{aligned}
$$

In other words, if a loss of $-\log(\alpha)$ is reported, then there is no estimated instance probability p_i that can satisfy $\alpha \leq \frac{1}{2} - \|\frac{1}{2} - p_i\|$ regardless of the accuracy of the predictions. Thus, a more in-depth understanding of $\max_k \mathcal{C}_N[k]$ can help uncover convergence properties about the individual instance estimates as the learning progresses.

To this end, further upper-bounds could be derived using Petrov's theorem (2007) that proposes a lower-bound for the tail of distributions with finite fourth moment. However, the complexity of the final statements overshadows its potential relevance. On the other hand, Le Cam's theorem (1960) combined with the properties derived so far yields a more interpretable result, which reveals that, as the loss decreases, small p_i will quickly converge towards zero:

Property 3.2 (Le Cam Upper Bound)**.**

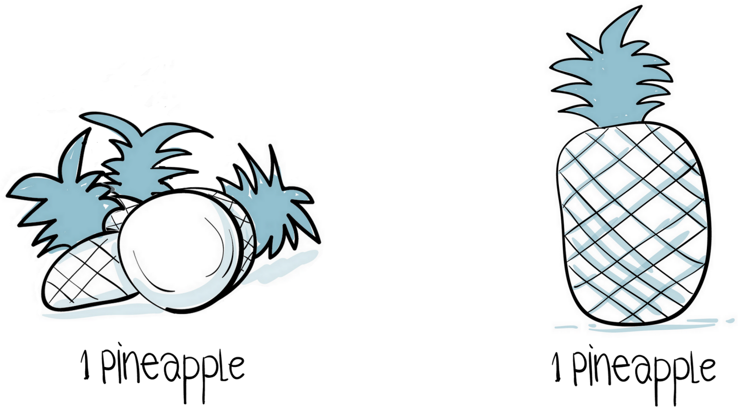
$$\begin{aligned}
\max_k \mathcal{C}_N[k] &\overset{(3.11)}{\leq} \min_{n\leq N} \max_k \mathcal{C}_n[k] \overset{\text{ind}}{=} \min_{\sigma\in P} \min_{n\leq N} \max_k \mathcal{C}_{N,\sigma}[k] \\
&\overset{\text{Le Cam}}{\leq} \min_{\sigma\in P} \min_{n\leq N} \max_k \frac{\lambda_{\sigma,n}^k e^{-\lambda_{\sigma,n}}}{k!} + 2\sum_{j\leq n} p_{j,\sigma}^2 \\
&\overset{\text{def}}{=} \min_{\sigma\in P} \min_{n\leq N} \max_k \frac{\left[\sum_{j\leq n} p_{j,\sigma}\right]^k e^{-\left[\sum_{j\leq n} p_{j,\sigma}\right]}}{k!} + 2\sum_{j\leq n} p_{j,\sigma}^2,
\end{aligned} \tag{3.17}$$

where P represents the set of all permutations and $p_{j,\sigma}$ stands for p_j after permutation σ; the same notation is used for $\mathcal{C}$ and λ. This property can be more easily interpreted if one considers the permutation σ which sorts the p_i in ascending order.

Example (Numerous Small p_i)**.** Suppose that the hundred smallest p_i are equal to 0.01. Substituting into Equation 3.17 then yields $\max_k \mathcal{C}_N[k] \leq \frac{1}{e} + 0.02$, which leads to a sizable cross-entropy value. Consequently, as the learning progresses, even the smallest p_i have to decrease and converge towards 0 to avoid the count distribution $\mathscr{C}$ to diffuse.

Prediction Sparsity

In conclusion, almost binary predictions will emerge implicitly from the model constraints as a byproduct of learning to count instances. First, Theorem 3.1 demonstrates that the loss is minimized if and only if the predictions are sparse. Second, Theorem 3.2 shows that the optimization is bound to converge towards one of the global minima since the set of local minima of the loss function with respect to the individual instance probabilities is the set of global minima itself. Overall, the combination of these statements demonstrates that the predictions converge towards sparsity as the learning progresses. These central theorems are further supported by convergence properties which highlight how the probability estimates converge towards one of these minima as the learning progresses. Indeed, on the one hand, the upper-bound derived using Le Cam's inequality (Equation 3.17) states that, unlike most benchmark models, a contribution of 1 to the count cannot be split into numerous small p_i contributions as the loss decreases. This feature



(a) Deterministic Counting (b) Poisson-Binomial Counting

Figure 3.2: Impact of sparsity on the action of counting. Deterministic counting only takes into consideration the sum regardless of the nature of the individual contributions (e.g., 1 pineapple is equivalent to a stack of pineapple leaves). In contrast, Poisson-binomial counting ensures that counts can be traced back to indivisible integer instance contributions (e.g., 1 pineapple is always equivalent to a whole pineapple).

implies that most of the mass for a single occurrence must be assigned to a few instances only. On the other hand, convergence property (Equation 3.16) derived from the first upper-bound (Equation 3.14) shows that the instance probability estimates converge towards clear-cut values (i.e., either towards 1 or 0).

> In summary, if a model accurately learns to count using the Poisson-binomial loss function, the instance probabilities it infers will implicitly be sparse.

Application Limitations

This unique prediction sparsity effect has a limiting effect on the use of the Poisson-binomial loss function (Equation 2.9) as a counting model in uncertain settings (i.e., in setups where the optimal count distribution has non-zero variance)—as already mentioned in Section 2.2. Indeed, even though the counting approach is stochastic, the model is trained to match the variance-free distribution of the labels (scalar values), and thus does not necessarily have the leeway to infer or capture the potential variability of the data. This is however not an issue for any of the applications presented in the next chapter since, in this work, counting is a means to achieve a sparsity-inducing effect rather than an objective on its own.

Further, since the loss function forces the model to make hard decisions, it does not allow for a flexible quantification of the uncertainty of the individual instances. For instance, in object detection tasks, it is highly likely that several similar bounding boxes cover the same object. In such a setup, the optimal detection probability with respect to the Poisson-binomial loss function consists in assigning all the probability mass to any one of these bounding boxes and setting all remaining probabilities to zero. This sparsity effect does thus not allow to model the fact that any of the bounding boxes that cover the object of interest could also have been selected as final detection. Overall, the Poisson-binomial model offers a differentiable way to learn instance selection, but does not reflect the uncertainty of the individual instance probabilities without the use of additional techniques—e.g., MC-dropout (Gal & Ghahramani, 2016).

3.3 Illustration: Visual Question Answering

The work of Trott et al. (2018) addresses a specific visual question answering (VQA) (Antol et al., 2015) sub-task, namely answering *counting-based* questions about images (e.g., "How many people are wearing blue shorts in this image?" and "How many cars are there in this image?"). While their main contribution resides in the proposed hard sequential bounding-boxes selection process that is trained through reinforcement learning, the simple counting-based baseline proposed in the paper is of particular interest for this section. In fact, the benchmark is very similar to the deterministic counting-based approach defined in Equation 2.2 and described in Section 3.1.2. Indeed, the approach consists in weighting bounding-box proposals obtained using a pre-trained detection model. The model thus assigns a value $\hat{p}_i \in [0, 1]$ to each instance (i.e., the bounding-boxes) and infers counting-based answers by simply summing these individual contributions, i.e., $C = \sum_i \hat{p}_i$. The model is then trained using a squared loss between the estimated count and the labeled count as described in Section 3.1.2.

In order to assess to what extent the theoretical sparsity-inducing abilities of the Poisson-binomial loss function hold in practice, this section replicates the baseline experiment conducted in (Trott et al., 2018) using the Poisson-binomial counting loss function instead of the deterministic counting-based approach for training.

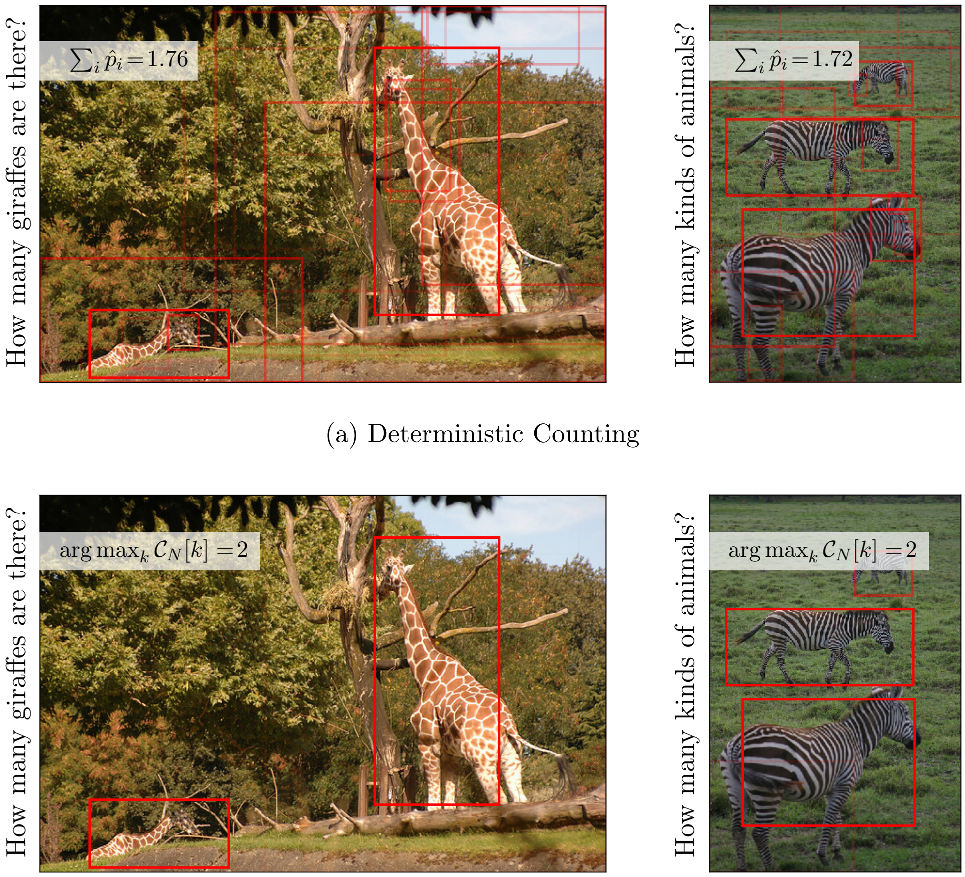
All other components of the model (e.g., architecture and hyperparameters) are kept unchanged so as to allow for an unbiased comparison.

Experiment Results

Replacing the original counting loss function with the Poisson-binomial counting one has almost no effect on raw performance when considering the metrics proposed in (Trott et al., 2018). Indeed, the test accuracy (i.e., the proportion of rounded counts that match the label) slightly slides from 49.6% to 48.2%, while the root-mean-square error faintly increases from 2.36% to 2.37%. However, the two approaches differ significantly when considering the sparsity of the predictions. On the one hand, as depicted in Figure 3.3b, the predictions yielded by the model after training with the Poisson-binomial-based loss function are more clear-cut than the ones obtained from models trained with the original objective. Indeed, in the former case, only a fraction of the bounding-boxes are given a non-zero weighting resulting in a very *sparse* probability assignment. On the other hand, the standard approach relies on a large number of small but non-negligible contributions to infer the count, see Figure 3.3a; almost every bounding-box is assigned a non-zero weighting. While this difference in behavior has no direct impact on the overall performance of the models in this specific task, the sparsity of the predictions of the Poisson-binomial model yields counts that are significantly more interpretable. Indeed, in the example showed in Figure 3.3, the count-based answers can directly be traced back to only a few bounding-boxes, which is not the case for the original approach. Thus, this example demonstrates that models trained with the proposed loss can implicitly learn prediction sparsity in an end-to-end fashion as a byproduct of learning to count instances.

Experiment Limitations

Changing the loss function does not solve the inherent shortcomings of the original model. Indeed, the simple counting baseline proposed by Trott et al. (2018) only relies on the visual content of the bounding-boxes, and thus discards any information about their spatial location and relationship. (The full hard sequential bounding-box selection process they propose integrates this crucial information in the learning process.) This lack of spatial awareness is especially problematic in the very common situation where several similar boxes are outputted by the


How many giraffes are there?
$\sum_i \hat{p}_i = 1.76$
How many kinds of animals?
$\sum_i \hat{p}_i = 1.72$
How many giraffes are there?
$\arg\max_k \mathcal{C}_N[k] = 2$
How many kinds of animals?
$\arg\max_k \mathcal{C}_N[k] = 2$

(a) Deterministic Counting

(b) Poisson-Binomial Counting

Figure 3.3: Counting-based visual question answering. The model—optimized using either the original loss function (Trott et al., 2018) or the Poisson-binomial loss function—is trained to answer counting-based questions by assigning a probability to each bounding-box and by aggregating these values. The color intensity of the bounding-boxes represents the probability assigned to them. Overall, the sparsity of the predictions resulting from the Poisson-binomial-based learning makes the answers much more interpretable, since the individual contributions can easily be traced back to a few high-probability bounding-boxes.

bounding-box proposal model. In this case, the model has no possibility of knowing whether the content of these boxes corresponds to the same instance or to multiple similar objects scattered across the image. In fact, if the model is given similar bounding-boxes, there exists an inherent *uncertainty* about how much weighting has to be assigned to each of them.

In addition, there is no guarantee that the model is given at least one bounding-box for each instance that needs to be counted. For instance, in order to correctly answer the question "How many layers are in the cake?"—an actual question in the dataset—the bounding-box proposal model must have outputted at least one box per layer in the cake; a requirement that is understandably not always met. When the condition is not fulfilled, the training process will encourage the model to assign prediction mass to other unrelated bounding-boxes, which will further increase the uncertainty surrounding the optimal probability assignment.

Overall, these limitations of the underlying model explain, in part, why incorporating to the learning process—through the use of the Poisson-binomial loss function—the additional prior that predictions have to be sparse does not increase the performance of the model. Indeed, in this specific example, the optimal probability assignment is, in fact, not necessarily sparse.

3.4 Conclusion

This chapter shows how successfully learning to count instances using the Poisson-binomial loss function implicitly teaches the model to achieve prediction sparsity. This claim is supported theoretically by both an analysis of the global maxima of the loss function (Theorem 3.1) and by several convergence theorems (Equation 3.16 and Property 3.2). Intuitively, this unique sparsity-inducing ability stems from the modeling of counts as distributions rather than scalars (see Chapter 2), which allows the model to take into account the higher moments of the label distribution during the training process. Thus, by penalizing count predictions that do not match the variance-free distribution of the labels, the Poisson-binomial loss function implicitly rewards models that infer a sparser assignment of probabilities. Overall, this approach offers the possibility to indirectly learn sparsity in an end-to-end manner without the need for an explicit (often non-differentiable) sparsity-inducing operation.

The sparsity-inducing ability of the loss function is also demonstrated empirically in the counting-based visual question answering experiment of Section 3.3. Indeed, in contrast to standard counting loss functions, the Poisson-binomial loss function trains models that infer much more interpretable answers as the individual contributions constituting the count can easily be traced back to a few high-probability bounding-boxes (Figure 3.3). The results have also highlighted that sparser predictions both do not necessarily lead to improved performance and do not necessarily overcome the shortcomings inherent to the original model. Indeed, in order to be successful, the incorporation of prediction sparsity into the learning process needs to be warranted by the task.

The rest of this work therefore focuses on applications where the optimal predictions are known to be sparse. In these settings, the inclusion of a more tailored supervision is expected to be beneficial to the performance and generalization capabilities of the models. The next chapter shows how valuable learning prediction sparsity can be in some settings.

Chapter 4

Weakly-Supervised Temporal Localization

Based on *Weakly-Supervised Temporal Localization via Occurrence Count Learning*, Schroeter J, Sidorov K, Marshall D, ICML 2019

At first glance, the Poisson-binomial counting loss function appears more suitable as an additional means of supervision used in conjunction with more traditional fully-supervised loss functions, rather than as a standalone training objective. Indeed, even though achieving prediction sparsity is certainly valuable in some scenarios, making sure that the non-zero predictions are accurate and meaningful is even more crucial. For instance, in discrete-time drum hit detection (Wu et al., 2018), it is not only important to detect the correct number of event occurrences, but also to detect the correct occurrence times. Prediction sparsity alone without temporal correctness of occurrence is often of limited value.

This chapter demonstrates how the Poisson-binomial counting loss, introduced in Chapter 2, can nevertheless be used as a *standalone* loss function for the learning of temporal event detection in specific settings. As a matter of fact, in the context of discrete-time temporal point event localization, being able to successfully count and, by extension, detect the correct number of events is already in itself a relevant feature. Indeed, if a model is capable of triggering the right number of times, all that remains is to ensure that these triggers occur neither too early nor too late (w.r.t. the ground-truth) in order to obtain accurate and precise temporal point estimates. Overall, this chapter not only shows how training with the Poisson-binomial counting loss implicitly and effectively prevents early detection in causal settings, but also proposes a simple trick to encourage the model to infer predictions without any temporal delay (i.e., thus avoiding any late bias). All in all, this chapter demonstrates that, in several setups, precise temporal point event detection can be achieved merely by learning to count event occurrences in a *weakly-supervised* fashion without any additional supervision.

4.1 Counting-based Weakly-Supervised Temporal Detection

This chapter tackles the precise temporal detection of *instantaneous* events in *causal* sequences with applications in many domains including sports video analysis (e.g., detection of goals, fouls, passes on video extracts), music transcription (e.g., detection of piano notes or drum hits in sound extracts), time series analysis (e.g., detection of events in wearable sensors time series). The difficulty of the detection problem can often be attributed to the high temporal precision required of the model predictions. Indeed, in music transcription, an error tolerance of 50ms for considering a prediction as correct is standard practice (Hawthorne et al., 2017; Wu et al., 2018). In such a setting, models are challenged not only to classify notes or drum hits correctly, but also to detect these events very accurately in time. Further, temporal applications are often characterized by complex temporal dynamics; there exist intricate underlying dependencies between the consecutive timesteps of the training samples and their respective label sequences. For instance, in order to detect the last jump of a triple jump, the first two jumps have to be accurately detected and remembered long enough. In music, while a drum hit is instantaneous, the sound produced by the action can resonate for multiple seconds. Thus, in this case, the model has to learn not to trigger more than once, despite hearing the drum sound over multiple timesteps. Overall, while action occurrences can be instantaneous, they often cannot be detected without complete awareness of their temporal context.

The two main assumptions underlying the problem definition (i.e., instantaneous events and causal temporal setting) can appear to narrow the range of potential applications; yet they are only weak constraints. First, in such a setup, event durations can nevertheless be modeled by labeling the beginning and end of events (event boundaries) as two separate event classes. Secondly, while the causal setting assumption prevents the use of bi-directional architectures and architectures comprised of representation learning with large temporal receptive fields, temporal dynamics can often be captured effectively through causal models, since time itself is causal.

Novelty

The novelty of the framework introduced in this chapter is that the training of precise temporal localization models is performed using only event occurrence counts as labels. The drive towards weaker labels is motivated by the fact that dataset labeling has become an ever-growing burden for the efficient deployment of deep learning solutions to real work applications. Indeed, while an unfathomable amount of unstructured and unlabeled data are being produced and stored every day, their manual annotation is extremely tedious and time-consuming (Deng et al., 2009b); dataset labeling remains the bottleneck in the dataset creation process. While automated labeling based on external data (Abu-El-Haija et al., 2016) (e.g., meta-data or comments on YouTube videos) can significantly speed up the process, the produced annotations are often not reliable enough for precise applications. In addition, both hand-labeling and automated-labeling suffer from the inherent risk of introducing errors or imprecision in the dataset (Frénay & Verleysen, 2014). In fact, even domain experts would struggle to label fast-paced piano music extracts with perfect accuracy and millisecond precision.

In contrast, occurrence counts can be obtained fairly easily or are, in some cases, readily available. For instance, in many sports, the total numbers of events over the whole game are often made available in an aggregated form (e.g., number of fouls, goals, passes), while their exact timestamps are more rarely provided. In piano music, counting the number of notes on a sheet of music is straightforward, while spotting them with millisecond precision on an audio extract is far more challenging. Additionally, given the continuous nature of temporal event localization (i.e., events can lie anywhere in $\mathbb{R}$), the annotation of the exact timestamps is always subject to errors. Thus, in practice, the aim of the annotation process is not to perfectly transcribe the unobtainable ground-truth, but simply to limit as much as possible these inevitable temporal discrepancies. In contrast, occurrence counts are often well-defined and take values in $\mathbb{N}$, thus alleviating the risk of introducing biases and imprecision in the data.

4.1.1 Formal Problem Formulation

Since the proposed learning paradigm (count-based supervision) slightly differs from that of other works in weakly-supervised temporal localization, we begin by formally defining the task tackled in this chapter in order to better grasp the specificities of the approach.

Let $\mathcal{D}$ be the training data with N samples:

$$\mathcal{D} := \left\{ \left(\mathbf{X}^{(i)}, \mathbf{c}^{(i)} \right) : 0 < i \leq N \right\}. \tag{4.1}$$

Let us consider the relationship between predictor $\mathbf{X}^{(i)}$ and dependent variable $\mathbf{c}^{(i)}$. First, each $\mathbf{X}^{(i)}$ is assumed to be an observable temporal sequence, i.e., $\mathbf{X}^{(i)} = \left(\mathbf{x}_t^{(i)}\right)_{t=1}^{T^{(i)}} \in \mathbb{R}^{T^{(i)} \times \lambda}$. (Depending on the application, this can stand for any λ-dimensional time-series such as spectrograms, financial time-series or DNN-learned representations.) Second, we assume there exists an underlying unobservable event process $\mathbf{E}^{(i)} = \left(\mathbf{e}_t^{(i)}\right)_{t=1}^{T^{(i)}} \in \{0, 1\}^{T^{(i)} \times d}$, indicating the presence of events. (For ease of explanation, instead of general multivariate event processes we consider the univariate case ($d = 1$) throughout the remainder of this section.) Each event process is assumed to be a function of its predictors:

$$\left(e_t^{(i)}\right)_{t=1}^{\tau} = g\left(\left(\mathbf{x}_t^{(i)}\right)_{t=1}^{\tau} \right), \forall \tau \leq T^{(i)}. \tag{4.2}$$

This assumption is essential as it guarantees that there exists a direct—and potentially learnable—function between the data and the underlying event process. For instance, having a video of a tennis match as predictor $\mathbf{X}$ and a temporal transcription of a piano piece as event process $\mathbf{E}$ is a clear violation of this assumption since there is no direct connection between the two processes. In terms of model training, the existence of such a function indicates that the event process can potentially be estimated. The causal assumption further ensures that, if the relationship function g is known, then no knowledge of future observations is necessary to determine the existence of an event at any given time t. In fact, this causal property is often already present implicitly in temporal applications since time itself flows in only one direction. For example, the validity of a football goal is only determined by the goal itself and the preceding actions. The fact that the referee ultimately takes a stochastic decision does not change the intrinsic validity of the action. Of course, there exist events that cannot be detected in a causal setting. For example, detecting with certainty the first jump of a triple jump as it occurs—without any additional context nor future information—is extremely difficult since it can easily be mistaken as a standard long jump. Nevertheless, while such events can occur, the vast majority of actions can be modeled in a causal setting. Indeed, even the occurrence of the first jump of the challenging acausal triple jump scenario could be determined in a causal fashion by having some additional temporal context (e.g., whether the previous contender did a triple or a long jump), or by having some additional spatial context (e.g., position of the sand-pit with respect to the takeoff board).

Finally, the observable dependent variables $c^{(i)}$ are defined as the total number of occurred events:

$$c^{(i)} = \textstyle\sum_t e_t^{(i)}. \tag{4.3}$$

In fact, the nature of the dependent variable (i.e., a count) is the main distinguishing feature of the approach. Indeed, while the objective is to estimate the function g, and thus by extension to predict the underlying event process, only aggregated event counts are available for training. This contrasts with standard fully-supervised approaches which can rely on labels that are almost identical to the event process $\mathbf{e}$. More formally, the problem tackled in this section is the following:

> **Objective** (Weakly-Supervised Temporal Event Detection)**.**
> Estimate the event process $\mathbf{E}^{(\text{new})} = \left(e_t^{(\text{new})}\right)_{t=1}^{T} \in \{0,1\}^T$ underlying a new test process $\mathbf{X}^{(\text{new})}$ using only the data $\mathcal{D}$ for training.

Remark (Other Application)**.** In addition to localization in unseen data, the capability of the model to estimate event processes from counts is useful in itself and can be leveraged, for example, for enriching the training data. Indeed, once trained, the detection model can be applied to infer precise event occurrence estimates for each sequence in the training set. These finer-grained pseudo-labels can then be used, for example, for training fully-supervised models that rely on precise event location for learning or simply for analyzing the data.

This problem formulation entails additional implicit assumptions about the nature of the event processes:

Event Uniqueness In this discrete-time setting, each timestep can only contain a maximum of one event of each event class. Indeed, the event process—which takes values in $\{0,1\}^{T_i \times d}$—only indicates the presence, or the absence, of an event at each timestep. While this condition is often implicitly met in most datasets, the issue, if present, can easily be alleviated by the use of smaller temporal granularities.

Instantaneous Events As already mentioned, the events are assumed to be instantaneous (i.e., lasting only one timestep in our discrete setting). In this context, there are, nevertheless, multiple ways of modeling event duration. For instance, the starting and end point of an action can be modeled as two separate classes. Another approach could leverage the fact that this localization assumption only needs to hold in the representation space, and thus that $\mathbf{x}_t^{(i)}$ may correspond to representations of longer time-intervals in the original data space (e.g., windowing).

4.1.2 Related Works

In recent years, research on weakly-supervised temporal localization has focused on two main applications: video (Duchenne et al., 2009; Bojanowski et al., 2014; Huang et al., 2016b; Graves et al., 2006; Richard et al., 2017; Niebles et al., 2008; Nguyen et al., 2009; Gan et al., 2015; Singh & Lee, 2017; Wang et al., 2017; Nguyen et al., 2018; Shou et al., 2018; Niebles et al., 2010) and audio (Xu et al., 2017, 2018; Kong et al., 2017; Kumar & Raj, 2016; Liu & Yang, 2016; Lee et al., 2017; Adavanne & Virtanen, 2017) event detection. These papers are briefly examined in this section.

Weakly-Supervised Video Action Localization

Weakly-supervised temporal action localization in videos has been an active area of research. First, Duchenne et al. (2009) proposed a discriminative clustering approach to segment action snippets from the background. This clustering framework was later revisited by Bojanowski et al. (2014) to handle temporal assignment problems—i.e., to partition the sequence using an ordered list of actions. This problem was also addressed by Huang et al. (2016b) using an extended Connectionist Temporal Classification method (Graves et al., 2006) and by Richard et al. (2017) who introduced a fine-grained subaction model.

Another prevailing problem in action localization consists in action intervals prediction rather than temporal segmentation. Initial works include the unsupervised generative "bag of spatio-temporal interest points" approach proposed by

Niebles et al. (2008) which infers a more general spatio-temporal localization. The same problem was also tackled by Nguyen et al. (2009) through the simultaneous learning of segment selection and classification. Later, Gan et al. (2015) used spatio-temporal saliency maps that were obtained by back-passing through the classification CNN to achieve localization, while Singh and Lee (2017) extended their Hide-and-Seek approach to action localization. Recently, attention-based approaches have been used extensively. First, Wang et al. (2017) introduced the UntrimmedNet—an attention model performing localization on pre-selected video segments. The mechanism was further improved by Nguyen et al. (2018) with class-specific activation maps, while Shou et al. (2018) replaced the fixed thresholding with a dynamic approach based on the proposed Outer-Inner-Contrastive (OIC) loss.

However, by focusing on subsegments regardless of their temporality, most methods neglect additional temporal information contained in the data (e.g., the relative order of events and what precedes an event). To address this issue, Niebles et al. (2010) modeled actions as a composition of motion segments. In this chapter, as the core of the proposed approach relies on recurrent units, the temporal nature of the data is intrinsically taken into account.

Weakly-Supervised Audio Localization

As in action localization, attention-based models have become a common solution to weakly-supervised audio localization tasks. Xu et al. improved their own attention-based convolutional recurrent neural network (Xu et al., 2017) by applying a trainable gated linear unit instead of the classical ReLU (Xu et al., 2018), while Kong et al. (2017) performed joint detection and classification on overlapping blocks. Alternatively, Kumar and Raj (2016) leveraged multiple-instance learning to address the localization task. A similar method based on convolutional networks rather than support vector machines or classical neural networks was proposed later by Liu and Yang (2016). Lee et al. (2017) further improved the framework by incorporating segment-level and clip-level predictions ensembling. Finally, Adavanne and Virtanen (2017) used a stacked convolutional and recurrent network to sequentially predict stronger and weaker labels.

> Overall, as for actions in videos, weakly-supervised localization in audio is also commonly achieved explicitly using attention mechanisms or segment-level detection and classification. This contrasts with the approach proposed in this chapter which *implicitly learns localization* while the occurrence count is being learned. Another unique feature resides in the temporal nature of the events: this chapter focuses on localizing instantaneous events precisely (sometimes in the order of milliseconds) rather than estimating the approximate extent of longer actions.

4.2 Temporal Poisson-Binomial Counting Model

The main idea of this chapter is to design a model such that localization implicitly emerges by constraint: the model is intrinsically bound to output clear-cut estimates of event processes in order to make valid predictions of the number of occurrences.

4.2.1 Model Definition

The specificities of the task at hand are reminiscent of the Poisson-binomial counting model introduced in Chapter 2. Indeed, the labels provided for training are counts which, in this case, correspond to the number of observed events. Most importantly, each of one these counts is further defined as the sum of individual and *sparse* occurrence timesteps. Thus, the Poisson-binomial counting framework seems a natural fit for modeling the task addressed in this chapter:

$$\begin{aligned} \mathscr{C}^{(i)} &= \textstyle\sum_t E_t^{(i)}, \\ E_t^{(i)} &= \mathcal{B}\Big(p_t^{(i)}\Big), \text{ ind. Bernoulli,} \\ p_t^{(i)} &= f\left(\left(\mathbf{x}_n^{(i)}\right)_{n=1}^{t}\right), \end{aligned} \tag{4.4}$$

where the event occurrences $e^{(i)}$ and counts $c^{(i)}$ are realizations of $E^{(i)}$ and the (stochastic) count distributions $\mathscr{C}^{(i)}$ respectively. Once again, the independence assumption of the Bernoulli distributions is not problematic as temporal dependencies can be carried by the instance probabilities $p^{(i)}$ (see Section 2.1).

The main component of the model is the unspecified function f that ties the input data $\mathbf{X}$ and the event occurrences together. In this chapter, given the causal nature of the model, this function f is estimated using standard recurrent neural networks—such as LSTM (Hochreiter & Schmidhuber, 1997) or GRU (Cho et al., 2014) with model parameters θ:

$$\hat{p}_{\theta,t}^{(i)} = \hat{f}_\theta \left(\left(\mathbf{x}_n^{(i)} \right)_{n=1}^{t} \right) . \tag{4.5}$$

4.2.2 Poisson-Binomial Loss Function

As described in detail in Chapter 2, the parameter set θ can be estimated using the Poisson-binomial loss function introduced in Equation 2.9:

$$L(\theta) = -\textstyle\sum_i \log \left(\Pr \left(\hat{\mathscr{C}}_\theta^{(i)} = c^{(i)} \mid \mathbf{X}^{(i)} \right) \right) , \tag{4.6}$$

where the count distribution has the following form (see Equation 2.4),

$$\Pr(\hat{\mathscr{C}}_\theta^{(i)} = k \mid \mathbf{X}^{(i)}) = \sum_{A \in F_k} \prod_{l \in A} \hat{p}_{\theta,l}^{(i)} \prod_{j \in A^c} (1 - \hat{p}_{\theta,j}^{(i)}), \tag{4.7}$$

with F_k the set of all subsets of $\{1, 2, \ldots, T^{(i)}\}$ of size k. Thus, intuitively, the optimization is done by comparing the estimated distribution $\Pr(\hat{\mathscr{C}}_\theta^{(i)} = k \mid \mathbf{X}^{(i)})$ to the target sample distribution determined by $c^{(i)}$ using the Kullback-Leibler divergence (1951) which, in this specific case, corresponds to the cross-entropy and max-likelihood. (see Section 2.3 for more comprehensive implementation details about the computation of the loss).

The computation of this loss function is illustrated in Figure 4.1 for the specific example application of drum detection in music sequences: first, the instance probabilities $\hat{p}_{\theta,1}, \ldots, \hat{p}_{\theta,N}$ are estimated for each bin of the spectrogram—and each event class—using a model f **(a)**. The count distributions $\hat{\mathscr{C}}_\theta$ are then computed for each drum type **(b)** before being compared to the label counts to produce the final loss function **(c)**. All in all, the model *only* learns to count event occurrences; indeed, there is no finer-grained supervision (i.e., stepwise supervision) nor complex attention mechanism for learning localization.

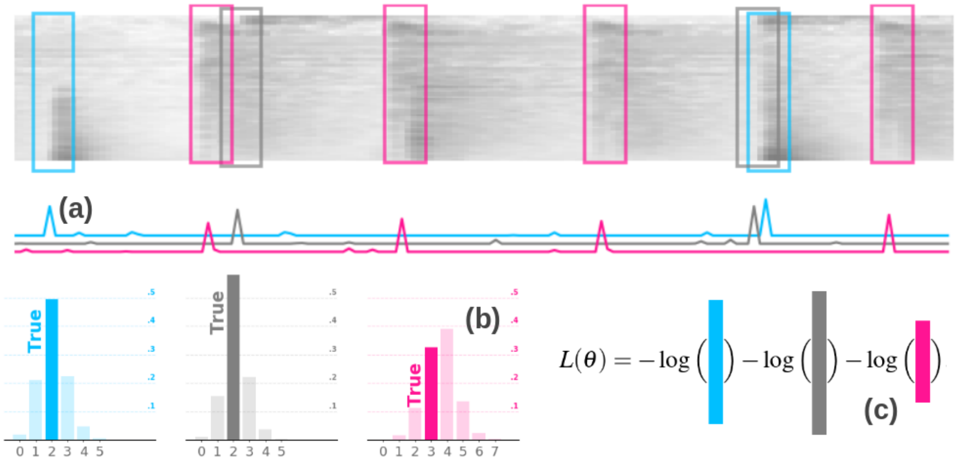


Figure 4.1: Illustration of loss computation for drum event counting. The model is given as input a spectrogram and successively estimates **(a)** the event occurrence probabilities $\hat{p}_{\theta,t}$ for each timestep and each drum type, **(b)** the count distributions $\hat{\mathscr{C}}_\theta$ for each class, and **(c)** the loss $L(\theta)$ through comparison of the estimated count distributions and the observed counts.

4.2.3 Why Does It Work?

The number of observed events, as a metric, is invariant to the temporal location of the occurrences. Indeed, if all notes of a piece of music were to be played a few seconds earlier or later, the number of notes played during the extract would remain unchanged. Therefore, it is unclear, at first sight, how using only such weak information for the training could lead to the successful learning of precise temporal event detection.

In this chapter, precise temporal event localization is achieved by concurrently ensuring that the probability assignments are sparse, that the occurrences are not detected too early, and that the model does not present a systematic late bias. In fact, this section demonstrates that these three elements can be learned successfully through count-supervision merely by relying on the properties of the Poisson-binomial loss function and by using a simple implementation trick.

Sparse Temporal Predictions

Learning temporal localization model through Poisson-binomial counting supervision, and thereby taking advantage of its implicit sparsity-inducing properties, ensures both that the number of predicted events converges towards the correct number of events observed in the sequence and that the event occurrence probability estimates converge towards sparsity. Thus, if the model successfully learns to count instances, one clear-cut prediction with high probability (i.e., converging towards 1) will emerge implicitly for each event in the sequence, while the probabilities assigned to every other timesteps will converge towards zero. Therefore, as sparse predictions are guaranteed, the final step towards precise temporal event localization is to ensure that these detections are made at the correct times; not too late, nor too early with respect to the ground-truth.

No Early Triggering

The benefits of training with the Poisson-binomial loss function are not confined to prediction sparsity. Indeed, the underlying counting model also presents some inherent constraints that have a significant impact on the learning process:

Property 4.1 (Mass Shift Irreversibility)**.**
The sequence of random variable $\left(\mathcal{C}_n\right)_{n=1}^{T}$ is monotonically increasing.

This statement directly follows from the definition of $\mathcal{C}_n$ as a sum of non-negative random variables (i.e., $\mathcal{C}_n := \sum_{j \leq n} \mathcal{B}(p_j)$). Intuitively, the count can only increase as new instance probabilities p_{i+1} are being observed. This feature implies that any probability mass shift towards increasing count values can never be shifted back to smaller counts. Thus, even if new evidence comes to light when processing the later timesteps of a sequence, the count distribution cannot be reduced. This constraint clearly sets our approach apart from other sequential count classification-based models, which can freely update their mass distribution over time (e.g., standard recurrent neural networks trained with cross-entropy).

This strong implicit model constraint actually prevents the model from triggering early in *causal* scenarios. Indeed, this irreversibly of mass shifts deters the model from anticipating events and triggering early since, if the actual event does ultimately not occur, the model has no means of reducing the count a posteriori, leading to a significant surge in the loss. Counting models trained with the Poisson-binomial loss function have thus no incentive to trigger early; to the contrary, any anticipated mass movement can only result in an increase of the loss. Consequently, as the count-based learning progresses and as the loss decreases, the model is increasingly enticed to avoid the negative loss contributions stemming from the anticipation of events, and thus the model implicitly and progressively learns to reduce the occurrence of early detections until sparsity is achieved.

No Systematic Translational Late Bias

In contrast to early triggering, the loss function does not implicitly preclude the model from triggering too late. While the ability of models to postpone detections is implicitly hindered by the nature of the learning process—since it would require them to keep detections in memory and thus allocate more of its resources to a worthless (in terms of loss minimization) mechanism—there is no hard constraint against such effect.

Nevertheless, this issue can be permanently addressed by training the model with sequences of different lengths. Intuitively, this implementation trick strips the model of its ability to learn the length of samples; the model thus never knows when a given sequence will end. This constant uncertainty about the existence of the next timestep prevents by itself the delaying of detections. Indeed, if the model were to systematically postpone its decision, any event occurring at the end of a sequence would be missed since the model has no opportunity to trigger after the last timestep. In conclusion, systematic late bias can simply be avoided by feeding sequences of variable lengths to the model for training.

In practice, there exist different ways of producing sequences of variable lengths. For instance, in music transcription, the length of the extracts can be artificially altered by slowing down or speeding up the rendition of the sequences through the modification of the sampling rate. Of course, if the task is highly dependent on the correctness of pitch (i.e., piano transcription), then an appropriate pitch correction has to be implemented. Another approach consists in padding the beginning of the sequence or in repeating timesteps (e.g., in video analysis).

Model Convergence

In summary, if the model accurately learns to count occurrences and if the events are detectable, then a coherent localization ability will emerge naturally:

> **Timely Detections** Overall, since the model outputs a single clear-cut trigger for each detection—because of the implicit prediction sparsity property of Poisson-binomial learning (Chapter 3)—and since the model can neither trigger too early nor systematically too late—because of the properties highlighted above, *precise* and *accurate* temporal event localization can be learned in causal settings using only occurrence counts for training.

4.3 Experiments

In this section, the localization capability of our approach is assessed on three challenging tasks: drum hits detection, piano onsets detection, and digits detection. All these tasks are relevant real-world applications that could benefit from an effective weakly-supervised model that alleviates the need for the costly manual annotation of training samples. The code used for these experiments is publicly available[1].

4.3.1 Drum Detection Experiment

The detection and classification of drum hits in audio extracts is an important task in music transcription. In the context of this chapter, the interest in these experiments is two-fold. First, given the instantaneous and highly-localized nature of drum hits, the model can be tested under minimal violation of our model assumptions. Secondly, the task—which requires predictions to be within 50ms of ground truth (Wu et al., 2018)—challenges the temporal localization precision of our model.

In this section, we replicate standard experiments proposed by Wu et al. (2018) and compare the effectiveness of our weakly-supervised model against *fully-supervised* state-of-the-art drum detection models.

[1] https://github.com/SchroeterJulien/ICML-2019-Weakly-Supervised-Temporal-Localization-via-Occurrence-Count-Learning

Drum Datasets

The experiments are based on two standard drum detection datasets: IDMT-SMT-Drums (Dittmar & Gärtner, 2014) and ENST Drums (Gillet & Richard, 2006) each with a different purpose. On the one hand, IDMT-SMT-Drums is comprised of real-world acoustic samples, as well as extracts from drum sample libraries and drum synthesizers. The drum extracts included in this dataset are relatively simple; the difficulty mainly comes from the significant sound differences between synthetic and acoustic drum hits. Overall, as described in (Wu et al., 2018), this dataset is used to test models on the specific task of "drum transcription of drum-only recordings" (DTD). On the other hand, ENST-Drums minus-one (Gillet & Richard, 2006) is considered a more challenging dataset, since it comprises of drum extract played live by three drum players on different drum kits. In addition, the dataset contains a significantly wider array of drum types. Thus, in this context, the model not only has to learn to detect hits from the three classes of interest—hi-hat (HH), snare drum (SD), and bass kick drum (KD), but it also has to learn not to trigger when other drum types are being played. In fact, this dataset can be a challenge even for fully-supervised models, since certain drum types have similar acoustic features. Overall, as described in (Wu et al., 2018), this dataset is used for the evaluation of "drum transcription in the presence of percussion" (DTP).

While both are standard datasets in the field of drum detection, they contain only a very limited number of extracts: 104 and 64 samples respectively. This is an issue for the training of the model, as learning drum detection and classification in a weakly-supervised manner with only the information conveyed by a few dozen counts is unrealistic. Thus, in order to artificially increase the size of the dataset, each audio extract is split into non-overlapping 1.5s snippets. For each of these snippets, the total number of occurrences for each drum type (hi-hat (HH), snare drum (SD), and bass kick drum (KD)) are then determined and used as training labels, thus discarding any localization information.

Model Architecture

In order to avoid overfitting, the network architecture is kept simple as the datasets are quite limited in size. First, the representation learning is comprised of six convolutional layers of size 3×4 with 8 to 16 filters intertwined with ReLU activations and max-pooling layers (see implementation for more details). The temporal dynamics are then captured through a 24-unit LSTM (Hochreiter & Schmidhuber, 1997). The final predictions (i.e., event occurrence probabilities) are obtained through a final 16-node fully-connected layer, followed by a sigmoid activation. Finally, while not necessary for the effectiveness of our approach, three separate models are trained to each detect a specific drum class in order to further simplify the learning process.

Model Training

Mel-spectrograms (Stevens et al., 1937) stacked together with their first derivatives are used as model input. The inclusion of the derivatives helps simplify the training process, as the model does not have to allocate any capacity to learn these features. In addition, data augmentation in the form of sample rate variations (i.e., playing sequences faster or slower) is applied during both training and inference. As the exact pitch of the drum hits is not essential to the task, there is no need to compensate for the shift in pitch that is associated with sampling rate alterations. This augmentation does not only constitute a simple way of artificially increasing the richness of the dataset but, most importantly, helps ensure that the model converges towards the actual event locations by generating sequences of variable lengths—as explained in Section 4.2.3 . In this setting, the final predictions are then obtained through the ensembling of predictions of a same extract sampled with different sampling rates.

The loss function is optimized using the standard Adam algorithm (Kingma & Ba, 2015). Finally, in order to speed up the loss computation, the count distribution is truncated using $c_{\max} = 29$, as described in Section 2.3. Note that since no sequence in the training set contains more than 28 events of a same class, this operation does not cause any loss of relevant information about the count distribution.

Model Evaluation

The effectiveness of our model is evaluated on both the D-DTD and D-DTP tasks (Wu et al., 2018), based on the IDMT-SMT-Drums (Dittmar & Gärtner, 2014) and ENST-Drums minus-one (Gillet & Richard, 2006) datasets respectively. In their work, Wu et al. (2018) further define two ways of sampling the train/validation/test splits for the cross-validation. The first one consists in randomly sampling extracts from the dataset regardless of the characteristics of the sample (*Eval Random*). To account for the stochastic nature of the training (i.e., the splits are sampled randomly), cross-validated results are then obtained by aggregation of six independent runs. The second sampling approach consists in partitioning the datasets based on drum specifications (i.e., drum synthesizers, real drum, drum from sound libraries) for the D-DTD task and based on drum players for the D-DTP task (*Eval Subset*). The models are then tested using a leave-one-partition-out cross-validation. Overall, this experiment helps assess the generalization capabilities of models, as they have to detect drum hits either from sources or from drum kits on which they have not been trained. The detailed evaluation protocol can be found in the work of Wu et al. (2018).

Drum Detection Results

Overall, our *weakly-supervised* approach is competitive against *fully-supervised* state-of-the-art drum transcription models on almost all metrics, as shown by the results in Table 4.1. Such a performance demonstrates that precise temporal localization can be achieved without any localization information, using only occurrence counts for training.

The model slightly under-performs in terms of snare drum detection on the *Eval Subset* experiments, especially on the leave-one-partition-out of the D-DTD task. A more in-depth analysis of these results reveals that these lower scores are caused by the ambiguity of snare drum (and kick drums) in the synthetic drum splits—as discussed above. Indeed, our weakly-supervised model is more conservative and does not trigger as easily when out-of-sample events do not sound like in-sample events. While this feature negatively impacts the overall score of the method on the D-DTD task, this rigidity allows for a significant reduction of the number of false positive triggers on the D-DTP task. Indeed, our method reports very high precision levels, beating all fully-supervised models on that metric. The imbalance between precision and recall can partially be explained by the design choice of selecting a simple narrow architecture rather than a larger network. From a practical standpoint, if needed, the precision and recall could be rebalanced by leveraging model ensembling and by adjusting the selection threshold accordingly.

Table 4.1: Drum Detection Results. Comparison between our *weakly*-supervised model and *fully*-supervised models evaluated in (Wu et al., 2018). The F_1 scores per instrument (KD/SD/HH), as well as the average precision, recall, and overall F_1 are displayed, [%]. For details: RNN, ReLUts (Vogl et al., 2016), RNN, tanhB (Southall et al., 2016), GRUts (Vogl et al., 2017) and lstmpB (Southall et al., 2017).

		D-DTD dataset						D-DTP dataset					
	Method	KD	SD	HH	Pre	Rec	F_1	KD	SD	HH	Pre	Rec	F_1
Random	RNN	97.2	92.9	97.3	95.7	96.9	95.8	**94.7**	79.5	88.3	84.1	**93.3**	87.5
	tanhB	95.4	93.1	97.3	93.9	97.1	95.3	92.4	84.6	87.1	86.3	92.1	88.0
	ReLUts	86.6	93.9	**97.7**	92.7	95.0	92.7	91.3	83.8	85.2	83.7	92.3	86.8
	lstmpB	**98.4**	**96.7**	97.4	**97.7**	**97.6**	**97.5**	94.4	84.1	91.4	90.8	90.8	**90.0**
	GRUts	91.4	93.2	96.2	91.8	97.2	93.6	94.2	**87.1**	87.7	88.6	92.7	89.7
	ours	96.0	90.4	97.1	95.1	93.9	94.5	92.3	81.2	**93.0**	**90.9**	87.1	88.9
Subset	RNN	88.0	85.3	93.2	86.0	95.1	88.9	**91.0**	57.8	82.2	72.8	**88.3**	77.0
	tanhB	91.9	89.9	**94.4**	**95.1**	91.2	92.1	82.7	61.6	84.8	74.1	83.8	76.4
	ReLUts	91.2	**90.9**	91.6	89.2	**95.8**	91.2	79.4	62.1	80.8	69.6	84.2	74.1
	lstmpB	**96.0**	88.7	93.8	93.8	94.0	**92.8**	85.8	**68.8**	83.7	78.3	84.7	**79.4**
	GRUts	89.1	90.6	91.7	89.6	94.2	90.5	87.7	62.3	79.4	73.0	85.2	76.5
	ours	88.0	79.5	93.9	90.6	84.3	87.1	84.9	59.4	**90.0**	**84.8**	73.5	78.1

While a temporal error tolerance of 50ms is strict and can seem particularly challenging for a model that does not have access to any localization information during training, the temporal localization error of our model is often significantly smaller than this predefined threshold. For example, by reducing the tolerance to 20ms, the F_1-score achieved by our model for hi-hats detection on the D-DTD *Eval Random* task only drops from 97.1% to 96.3%. In this case, the standard deviation of the distance between true and predicted timestamps is only 4.35ms, which is smaller than the granularity of the mel-spectrogram features used as model input. Overall, the results show that our model learns very precise temporal detection through count supervision only.

Remark

It must be noted that the results obtained through specification-based partitioning (Eval Subset) are slightly ambiguous. Indeed, in some extracts, the kick-drums and snare-drums generated by the drum synthesizers do not acoustically resemble the hits of real-life drums, nor the ones of sound libraries. Consequently, it is questionable whether these hits should actually be counted as false negatives.

While this is the case in standard evaluation protocols (Wu et al., 2018), the inclusion of these ambitious samples blurs the line between models that trigger more strictly and models that present a lack of generalization capabilities. In fact, this setup highly benefits overly sensitive models that are prone to trigger easily. This bias is exacerbated by the fact that the datasets only contain drum and percussion hits: there is little room for models to produce false positives. A clearer picture could thus be provided if the datasets were to include other sound events (e.g., dog barks, or gunshots) since it would challenge over-sensitive models not to detect resembling out-of-class events. Overall, while this bias cannot easily be alleviated, it is important to be aware of it.

Conclusion

In conclusion, the performance of the proposed weakly-supervised model on this task is remarkable, as it achieves, with much weaker labels, performance comparable to that of fully-supervised methods. Indeed, even though occurrence count labels do not convey any localization information to the network during training, our approach is still able to successfully learn precise temporal localization of drum hits. This confirms that, as long as the number of occurrences is estimated correctly, precise localization emerges naturally.

4.3.2 Piano Onset Detection Experiment

Note onset detection—detecting when notes are being played—is an essential part of music transcription. However, with 88 different channels, corresponding to the number of keys on a standard piano, and complex interactions, the specific task of piano onset detection is particularly challenging, especially for weakly-supervised approaches. In fact, not only does the task require the simultaneous detection of 88 different channels (i.e., notes) that are sometimes intertwined in complex musical compositions, but piano notes are far from independent from one another in terms of spectral representation. Indeed, the action of pressing a piano key activates the string of the corresponding note, which, in turn, causes the vibration of a multitude of other notes (i.e., harmonics) which combined produce a harmonically complex sound. Thus, accurate piano note detection relies on the ability of the models to learn to successfully recognize whether a note is heard as a harmonic to another note, or as a note on its own—which constitutes a difficult task of its own. Finally, the task requires the detections to be extremely precise with respect to the ground-truth; a tight temporal error margin of 50ms is often set to consider a prediction as correct. This feature is particularly relevant to our weakly-supervised approach since it has to achieve precise detection without any localization information for training—only occurrence counts. All in all, while being challenging, the task of piano onset detection is ideal to assess the effectiveness of our approach in complex setups, both in terms of class accuracy and temporal precision.

In this section, we replicate the experiment conducted by Hawthorne et al. (2017) using our weakly-supervised loss function. Even though their model also predicts note offsets and note velocities, only onset times are considered for this experiment since the estimation of these additional metrics can be achieved separately.

Piano Transcription Dataset

The MAPS database (Emiya et al., 2010) is used for training and evaluation. In order to strictly follow the dataset creation protocol from (Hawthorne et al., 2017), the model is trained on the synthesized pieces, while the evaluation is performed on the Disklavier pieces. Additionally, only the samples containing actual piano pieces are kept, thus discarding all samples containing only chords and single notes. Overall, these choices allow for a more realistic, albeit more challenging, evaluation.

Similar to the drum experiment, each audio extract is split into 1.5s non-overlapping segments to artificially increase the dataset size since the number of piano pieces included in the dataset is extremely limited (i.e., 210 samples). Once again, we train our model using only occurrence counts as labels.

Model Architecture and Training

The only difference between the model architecture chosen for this experiment and the one used in Section 4.3.1 for drum transcription lies in the number of convolution filters and recurrent units. These have been slightly increased by a factor of 2 and 4 respectively since the task at hand presents more complex events and dynamics. Once again, the loss is optimized using the Adam algorithm (Kingma & Ba, 2015). To improve the model performance, we perform slight data augmentation during training in the form of time stretching and extract

Table 4.2: Piano Onset Detection Results. Comparison on the MAPS dataset between our *weakly*-supervised approach and *fully*-supervised models evaluated in (Hawthorne et al., 2017). Precision, recall, and F_1 scores are shown in %.

Method	Pre.	Rec.	F_1
Sigtia et al. (2016)	44.97	49.55	46.58
Kelz et al. (2016)	44.27	61.29	50.94
Hawthorne et al. (2017)	**84.24**	**80.67**	**82.29**
ours	76.22	68.61	71.99

stacking (i.e., playing two samples simultaneously). As already highlighted for the drum experiment in Section 4.3.1, the time stretching operation plays a key role in the model, since it generates sequences of variable length (see Section 4.2.3 for a discussion of the importance of having sequence of different lengths in the dataset). At inference time, we ensemble the predictions obtained for samples augmented with different randomly sampled time stretching factors. Finally, we truncate the count distribution after $c_{\max} = 40$ bins (see Section 2.3).

Model Evaluation

For the evaluation, as mentioned above, a prediction is considered correct if it falls within 50ms of the ground-truth (Hawthorne et al., 2017). To be consistent with the results in (Hawthorne et al., 2017), final metrics (i.e., precision, recall, and F_1-score) are reported as the mean across all pieces' scores, which are computed using the `mir_eval` library (Raffel et al., 2014).

Piano Onset Detection Results

The proposed weakly-supervised model achieves remarkable piano onset detection performance as shown in Table 4.2. Indeed, despite only having access to much weaker labels for training, the counting-based training yields results that are almost comparable to that of fully-supervised models (Hawthorne et al., 2017), while significantly outperforming the previous fully-supervised state-of-the-art approaches (Sigtia et al., 2016; Kelz et al., 2016). Figure 4.2, which depicts the out-of-sample detection performance of our model, shows that the approach yields strong results on complex piano pieces. These observations emphasize once again

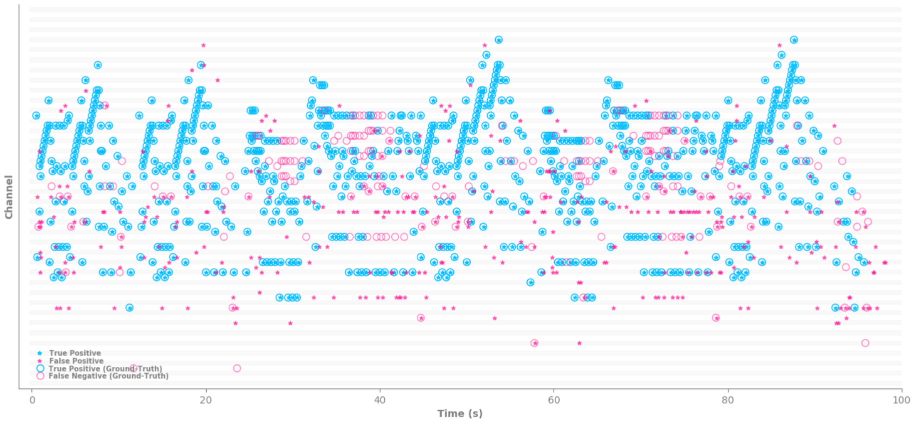


Figure 4.2: Out-of-sample piano onset detection for the piece *Lyric Pieces Book III, Opus 43, No. 1 "Butterfly"* by E. Grieg.

the effectiveness of our approach, which achieves precise temporal localization (within 50ms) without using any localization information for training.

Additional investigations reveal that our model achieves very strong detection performance for medium to high notes while being less effective in the lower registers. This effect could be explained both by the richer harmonic structures induced by lower notes, by the fact that these notes are simply less frequent in the dataset, and by the increased inaccuracy of the spectrograms in the lower register. These shortcomings could be alleviated by applying specific spectral transformations to the sound extracts and by artificially increasing the number of lower notes in the dataset (e.g., pitch-shifting). However, while incorporating these heuristics in the localization pipeline would make our method even more competitive, they are beyond the scope of this section. Indeed, the central aim of this work is to show the potential of Poisson-binomial count-based supervision, and not to evaluate additional heuristics in an attempt to optimize raw metrics.

As observed in the drum experiment (Section 4.3.1), training with the Poisson-binomial counting loss yields a model that achieves significantly higher precision than recall. This effect might emerge from the strong prediction sparsity properties of the model (see Chapter 3). In fact, as the learning progresses and as the predictions converge towards clear-cut probabilities (i.e., either towards 0 or 1), the model has no means of expressing uncertainty when borderline cases arise. Indeed, timesteps must either contain a detection or not, there is no other alternative. Therefore, it can be expected that the model is more reluctant to trigger in highly ambiguous situations or in scenarios that have not been observed before (refer to the drum experiment for a similar observation). Overall, rather than a flaw, this cautious behavior is simply a feature of the model that has to be taken into account when training with this unique loss function.

4.3.3 MNIST Digits Detection Experiment

In this section, we show the versatility of the Poisson-binomial counting by learning to detect digits in images using only occurrence counts for training. This experiment also allows for a more in-depth quantification of the representation learning and the localization learning capabilities of the model separately.

Dataset Generation

The samples used for this experiment are generated using the well-known MNIST dataset (LeCun et al., 1998). More precisely, for each image in our training dataset, we randomly sample hand-written digits from the training set and place them uniformly at random—subject to a non-overlapping condition—on the image (see Figure 4.3 for examples). The test set is generated similarly by randomly sampling digits from the MNIST test set.

From Images to Sequences

Our counting-based weakly-supervised temporal localization model requires both the model inputs to be sequences and the model outputs to be point localizations. Thus, in order to achieve image digit detection in this setup, we first have to map the original images ($\mathbb{R}^{W\times H\times d}$ space) into sequences of sub-images ($\mathbb{R}^{T\times(w\times h\times d)}$ space). This can be achieved by sampling windows of size $w \times h$ along a space-filling curve (Peano, 1890). For this experiment, we propose using the standard Hilbert curve (1891). Given a sequence of sub-images, the model then outputs temporal localization estimates $\{\mathbf{p}_1, \ldots, \mathbf{p}_T\}$, indicating for each digit class and each sub-image of the sequence the probability that the sub-image contains a digit of that class. Thus, in order to obtain the final spatial detection estimates, the temporal sequence of probabilities is then mapped back onto the original image space (see Figure 4.3 for examples of resulting detection).

Prediction Sparsity

In such a setting, given how the sequences are generated, the same digit can potentially be found on multiple sub-images of the input sequence. Therefore, in order to be able to minimize the counting loss, the model has to learn to trigger only once per digit. Overall, this is quite a challenging feat as the model, while only having access to occurrence counts for training, has to learn how the space is mapped (i.e., through a Hilbert curve) in order to avoid duplicate detections—on top of learning to detect and recognize digits. Thus, while the Poisson-binomial loss function was theoretically shown to encourage prediction sparsity, this experiment tests the limits of its convergence ability since the task tackled here requires sparsity, space mapping, detection, and recognition to be learned simultaneously in a weakly-supervised manner.

Model Architecture and Training

The representation learning part of the network is identical to the convolutional layers of the VGG-13 architecture (Simonyan & Zisserman, 2014)—without the final fully-connected prediction layer, while the localization learning part consists of a 48-unit LSTM. Finally, the temporal representations outputted by the recurrent units are mapped to digit class probabilities using a 24-node fully-connected prediction layer followed by a sigmoid transform. The simultaneous learning of sparsity, digit representation, space mapping, and localization is done in a *weakly-supervised* fashion using only the number of occurrences of each digit in the *original* image—i.e., not in the individual sub-images—as training labels; indeed, no additional information is given to the network.

Digit Recognition Performance

We first assess the digit recognition capability of our model—trained on the images of digits described earlier—in comparison to that of the standard fully-supervised VGG-13 architecture to test whether learning via count-based supervision is detrimental to recognition accuracy. This experiment focuses on the digit classification ability of the models with no regard for localization. While the computation of the recognition accuracy for the standard VGG-13 model is straight-forward (i.e., given the image of a digit, the final class prediction is defined as the class with

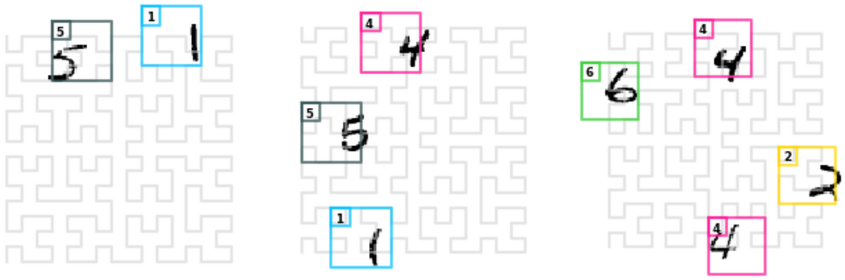


Figure 4.3: Out-of-sample predictions for the detection of MNIST digits. (Raw predictions without post-processing nor non-maximum suppression.)

the highest inferred probability), the evaluation of our model requires slightly more work. Indeed, given the structure of our model, the original test digits from the MNIST dataset cannot be inputted without modifications. Therefore, each original test digit is first pasted onto a larger image—of the same size as the images used for training, before being transformed into a sequence of sub-images, as described above. The model is then used to infer an occurrence probability for each digit class on each sub-image. These probabilities are combined—using the sum of Bernoulli distribution as described in Chapter 2—to yield an estimated count distribution for each digit class. As the objective of the classification task is to output the most likely digit, we select the digit class with the highest mean expected count (i.e., from the Poisson-binomial count distribution) as the final prediction to compute the recognition accuracy.

Overall, our model achieves a digit recognition accuracy of 99.12%, which is remarkably better than the score obtained by the fully-supervised VGG-13, 98.51%, even though both networks share the exact same representation learning architecture. This result shows that the learning of digit detection through count-based supervision is not achieved to the detriment of raw recognition accuracy.

Representation Learning Performance

The representation learning capability of our model alone—i.e., without taking the classification nor the localization ability into account—can be assessed by feeding original (28×28) MNIST digits as the sub-images composing our model's input sequence. Indeed, the intermediate convolutional representations yielded by the

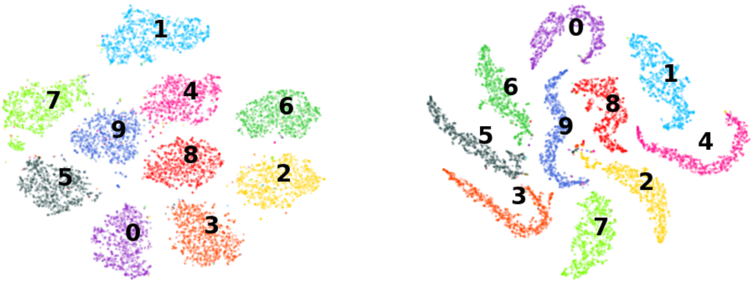


Figure 4.4: Digit Representations. Comparison of t-SNE digit feature representations resulting from the *fully*-supervised VGG-13 architecture (left) and from our *weakly*-supervised approach (right).

network are independent of the localization and classification part of the model. Thus, visualizing the learned features using t-SNE representations (Maaten & Hinton, 2008) can help evaluate how well the model separates the different digit classes early in the learning. In short, the t-SNE algorithm maps the N-dimensional representations onto a 2-dimensional space, while trying to preserve the spatial characteristics of the representations. The same analysis can be conducted for the fully-supervised VGG-13 network, by visualizing the representations produced by the last convolutional layer.

The convolution representations produced by our model and VGG-13 respectively can be observed in Figure 4.4. Overall, both models yield representations that are comparable in terms of class discrimination. The topological differences might simply be a result of the t-SNE projection and do not affect the class separability.

This small experiment demonstrates once again that the indirect nature of the counting-based learning and the overall weaker supervision have little to no impact on the quality of the learned representations, which are almost similar to the ones obtained using a comparable fully-supervised model.

Localization Learning Performance

We can finally assess the accuracy of the spatial localization by computing the mean absolute distance between true and estimated bounding-box centers. Our network achieves a value of 9.04 pixels, which is extremely close to the granularity of the space-filling curve (8 pixels). Recall, these fine-grained detections are achieved using only occurrence counts for training without any prior knowledge of localization and without any explicit example of the actual goal of the task (i.e., clear-cut and spatially precise bounding-boxes). It is important to note that the predictions displayed in Figure 4.3 are raw model outputs obtained *without any post-processing*. This last feature highlights, once again, the unique sparsity-inducing ability of the Poisson-binomial loss function.

In conclusion, the remarkable precision of the predicted bounding-boxes locations, the high digit recognition capability, and the meaningful representation learning ability of the model demonstrate, once again, the effectiveness of our *weakly-supervised* approach.

Model Limitations

This array of experiments shows how effective the Poisson-binomial counting loss can be in terms of localization learning. However, the setting used in this section is quite narrow and the current approach presents several non-negligible limitations when considering more advanced weakly-supervised object detection applications (e.g., MS-COCO (Lin et al., 2014) or Open Images Dataset (Kuznetsova et al., 2020)). For instance, the model couples the scale of the predicted bounding-boxes to the size of the sub-images. Even though going beyond the fixed-scale setup might be possible using multi-scale architectures or even more advanced adaptive-scale scanning processes learned via reinforcement learning, it is uncertain whether the information conveyed by count supervision will be sufficient to train these more complex models.

Nevertheless, while the setup is limited, the clear-cut predictions produced by the model without any non-maximum suppression or other post-processing operation clearly demonstrate the power of the sparsity-inducing property of the Poisson-binomial counting approach.

4.4 Conclusion

In this chapter, we show how implicit model constraints can be used to ensure that accurate localization emerges as a byproduct of learning to count occurrences. Experimental validation of the model demonstrates its competitiveness against fully-supervised methods on challenging tasks, despite much weaker training requirements. In particular, both precision in the order of a few milliseconds in the drum detection task and strong performance in the piano transcription experiment are achieved without any localization prior. Furthermore, the proposed approach displays the ability to naturally learn meaningful representations while learning to count instances.

The experiments in this chapter confirm the effectiveness of the sparsity-inducing property of the Poisson-binomial loss function. While being useful as a standalone training objective, the loss function might thus be even more valuable when leveraged as a sparsity regularization in conjunction with more targeted loss functions. This specific application of Poisson-binomial counting is addressed in the next two chapters on the tasks of robust temporal point detection (Chapter 5) and multi-instance sub-pixel point detection (Chater 6).

Chapter 5

Robust Temporal Point Detection with Misaligned Labels

Based on *Learning Precise Temporal Point Event Detection with Misaligned Labels*, Schroeter J, Sidorov K, Marshall D, AAAI 2021

Robust Temporal Point Event Localization through Smoothing and Counting, Schroeter J, Sidorov K, Marshall D, ICML Workshop on Uncertainty & Robustness in Deep Learning 2020

The previous chapter presented how the Poisson-binomial loss function can be leveraged as a standalone objective function for the weakly-supervised learning of temporal localization. The sparsity-inducing properties of the loss (e.g., Theorem 3.1) can, in fact, find much broader application when leveraged as a regularizer in conjunction with other more targeted loss functions to enforce sparsity constraints in an end-to-end fashion. Thus, this chapter explores a setting where the learning can benefit from such explicit inclusion of a sparsity constraint: namely, the robust learning of precise temporal point event detection with misaligned labels. Indeed, in practice, this task is often characterized by a discrepancy between the optimal predictions which are known to be sparse and the actual model predictions which are often widely dispersed over time. The main issue stemming from the scattered nature of the predictions is its inevitable adverse effect on the temporal precision of the detections. Therefore, this chapter offers a more effective modeling of the problem at hand by encouraging models to infer sparser predictions through count supervision.

Count supervision is especially relevant in the presence of misaligned labels since it offers a reliable means of supervision regardless of the noise level. Indeed, occurrence counts are higher-level descriptors when compared to event occurrence

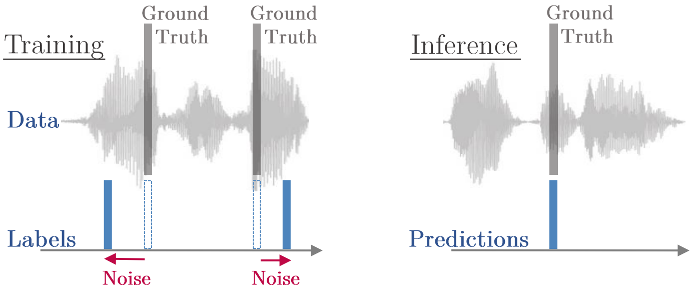


Figure 5.1: Task illustration. Model training solely relies on noisy labels that differ from the actual ground-truth, while the final inference objective is the *precise* localization of events.

timestamps; counts are implicitly contained in location information. Thus, while offering a weaker level of supervision when used as training labels, counts are inherently more invariant to temporal perturbations than the finer-grained location annotations. In fact, in temporal detection applications, occurrence counts remain correct regardless of the level of misalignment of the labels relative to the ground-truth. This chapter thus not only shows how count-based regularization helps achieve prediction sparsity in temporal point event detection applications (e.g., piano note onset detection, instantaneous event detection in videos) but also demonstrates how adding counting as an noise-invariant means of supervision allows for a more robust learning of temporal localization, especially when the labels are subject to large temporal misalignments.

5.1 Introduction

The surge of deep neural networks (LeCun et al., 2015; Schmidhuber, 2015) has accentuated the evergrowing need for large corpora of data (Banko & Brill, 2001; Halevy et al., 2009). The main bottleneck for the efficient creation of datasets remains the annotation process. Over the years, while new labeling paradigms have emerged to alleviate this issue (e.g., crowdsourcing (Deng et al., 2009a) or external information sources (Abu-El-Haija et al., 2016)), these methods have also highlighted, and emphasized, the prevalence of *label noise*. Unfortunately, deep neural networks are not immune to such perturbations, as their intrinsic ability to memorize and learn annotation errors (Zhang et al., 2017a) can be the cause of training robustness issues and poor generalization performance. In this context, the development of models robust to label noise is essential.

This chapter tackles the problem of precise temporal localization of point events (i.e., determining when and which instantaneous events occur) in sequential data (e.g., time series, video, or audio sequences) despite only having access to poorly aligned (w.r.t. the ground-truth) annotations for training (see Figure 5.1). This task is characterized by the discrepancy between the noisiness of the training labels and the precision expected of the predictions during inference. Indeed, while models are trained on inaccurate data, they are evaluated on their ability to predict event occurrences as precisely as possible with respect to the actual ground-truth. In such a setting, effective models have to infer event locations more accurately than the labels they relied on for training. This requirement is particularly challenging for most classical approaches that are designed to learn localization by strictly mimicking the provided annotations. Indeed, as the training labels themselves do not accurately reflect the event location, focusing on replicating these unreliable patterns is incompatible with the overall objective of learning the actual ground-truth. These challenges highlight the need for more relaxed learning approaches that are less dependent on the exact location of labels for training.

The presence of temporal noise in localization tasks is ubiquitous given the continuous nature of the perturbation; in contrast to classification noise where only a fraction of the samples are misclassified, no sample is perfectly aligned and clean extracts are simply the ones with the smallest error magnitude. Thus, temporal labeling is characterized by an inevitable trade-off between annotation precision and time investment. For instance, while a coarse manual transcription of a minute of complex piano music might be achieved within a moderate time frame, a millisecond precision requirement—a common assumption for deep learning models—significantly increases the annotation burden. In this respect, models alleviating the need for costly annotations are key for a wide and efficient deployment of deep learning models in temporal localization applications.

This chapter introduces a novel model-agnostic loss function that yields sparse point predictions despite relaxing the reliance of the learning process on the exact temporal location of the annotations. This softer learning approach inherently makes the model more robust to temporally misaligned labels.

5.2 Related Works

Classification with Noisy Labels

Classification in the presence of label noise—i.e., misclassified samples—has been a very active area of research (Nettleton et al., 2010; Raykar et al., 2010; Frénay & Verleysen, 2014) with three main solution axes: explicit noise modeling, loss function adaptation, and training on clean subsets. The direct application of classification-specific explicit noise modeling (Goldberger & Ben-Reuven, 2017; Liu & Tao, 2016; Patrini et al., 2017) or loss correction (Mnih & Hinton, 2012; Natarajan et al., 2013; Reed et al., 2014; Azadi et al., 2016) to temporal noise robustness is however limited as classification noise patterns differ from temporal noise structures (e.g., categorical vs. continuous). In addition, training with a subset of clean data (Han et al., 2018; Jiang et al., 2018) or underweighting noisy samples (Wang et al., 2018) does not generalize well to multi-class and multi-instance temporal applications.

Temporal Localization Under Label Misalignment

The literature on temporal noise robustness is limited despite the critical relevance of this issue. First, Yadati et al. (2018) propose solutions combining noisy and expert labels; however, these methods require a sizable clean subset of annotations, unlike our approach. Second, while Adams and Marlin (2017) achieve increased robustness by augmenting simple classifiers with an explicit probabilistic model of the noise structures, the effectiveness of the approach on more complex temporal models (e.g., LSTM) still needs to be demonstrated. Finally, Lea et al. (2017) perform robust temporal action segmentation by introducing an encoder-decoder architecture. However, the coarse temporal encoding comes at the expense of finer-grained temporal information, which is essential for the precise localization of short events (e.g., drum hits). In this paper, rather than a new architecture, we propose a novel and flexible loss function—agnostic to the underlying network—which allows for the robust training of temporal localization networks even in the presence of extensive label misalignment.

Weakly-Supervised Learning

Some weakly-supervised models leverage weaker annotations to infer finer-grained concepts. In such frameworks, noisy labels are implicitly bypassed by the use of higher-level labels—which are more invariant to perturbations. For instance, some works achieve object detection (Fergus et al., 2003; Bilen & Vedaldi, 2016) or temporal localization (Kumar & Raj, 2016; Wang et al., 2017) using only class-level annotations for training, while others only rely on occurrence counts (Gao et al., 2018; Schroeter et al., 2019) (see Chapter 4). However, finer-grained labels, even noisy ones, often contain some additional information that is essential for optimal performance.

Classical Heuristics

Our approach is closely linked to the more standard trick of label smoothing or target smearing (e.g., applying a $\tilde{\sigma}^2$-Gaussian filter $\Phi_{\tilde{\sigma}^2}$ to the labels) which has been considered to increase robustness to temporal misalignment of annotations (Schlüter & Böck, 2014; Hawthorne et al., 2017). However, this slight modification of the input data converts the original point prediction problem into a distribution prediction problem, which ultimately leads to several issues such as location ambiguity and prediction entanglement (see full discussion in Section 5.4.2). In contrast, our novel loss function does not suffer from any of these issues, and still manages to achieve a more robust localization learning.

5.3 Problem Formulation

For consistency with previous works—although not necessary for the definition and use of our loss function—time is assumed to be *discrete*. Apart from that, the main assumption of this chapter is the *instantaneous* nature (i.e., lasting exactly one timestep) of the events to detect. (Event duration can be modeled in such a framework by labeling the beginning and end of each event class as two separate channels.) In this setting—similar to the definitions in Section 4.1.1, each predictor $\mathbf{X}^{(i)}$ of the training data $\mathcal{D} := \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}) \mid 0 < i \leq N\}$ is an observable temporal sequence of length $T^{(i)}$ (i.e., $\mathbf{X}_i = (\mathbf{x}_t^{(i)})_{t=1}^{T_i} \in \mathbb{R}^{T^{(i)} \times \lambda}$), such as a DNN-learned representation or any λ-dimensional time-series. The *observed* label $\mathbf{Y}^{(i)} = (\mathbf{y}_t^{(i)})_{t=1}^{T^{(i)}} \in \{0,1\}^{T^{(i)} \times d}$ and the *unobservable* ground-truth event locations

$\mathbf{G}^{(i)} = \big(\mathbf{g}_t^{(i)}\big)_{t=1}^{T^{(i)}} \in \{0,1\}^{T^{(i)} \times d}$ are discrete sequences indicating whether one event of class d was observed—or occurred—at time t. (For the sake of simplicity, we set $d=1$; the case with multiple event classes (i.e., $d>1$) is a trivial extension.)

This chapter addresses the problem of label misalignment, i.e., $\mathbf{Y}^{(i)} \neq \mathbf{G}^{(i)}$. To that end, we model temporal label misalignment by assuming that the timestamps of labeled events $\mathcal{T}^{(i)}(Y) := \{t \in \mathbb{N}^{\leq T^{(i)}} \,|\, y_t^{(i)} = 1\}$ are perturbed versions of the unobservable ground-truth timestamps of event occurrences $\mathcal{T}^{(i)}(G) := \{t \in \mathbb{N}^{\leq T^{(i)}} \,|\, g_t^{(i)} = 1\}$, i.e.,

$$\underbrace{\{t \in \mathbb{N}^{\leq T^{(i)}} \,|\, y_t^{(i)} = 1\}}_{:=\mathcal{T}^{(i)}(Y) \in \mathcal{P}([1,\ldots,T^{(i)}])} = \{t_k + \boldsymbol{\epsilon_k} \,|\, g_{t_k}^{(i)} = 1, t_k \in \mathbb{N}^{\leq T^{(i)}}\}, \epsilon_k \overset{iid}{\sim} E, \tag{5.1}$$

where E is a discrete noise distribution. The aim of this chapter is thus the following:

Objective (Precise Event Detection)**.**

Estimate the true event occurrence times $\mathcal{T}(G)$ of an unseen input sequence $\mathbf{X}$ using only the noisy data $\mathcal{D}$ for training.

5.4 Classical Models

In what follows, for the sake of notation simplicity, all loss functions are presented for a batch size of 1 (e.g., the label sequence $\mathbf{Y}^{(i)}$ and its elements $\mathbf{y}_t^{(i)}$ become $\mathbf{y}$ and y_t respectively).

5.4.1 Stepwise Cross-Entropy

In this discrete setting, the standard approach to temporal point detection (Wu et al., 2018; Hawthorne et al., 2017) consists in densely predicting—often iteratively—an event occurrence probability $\hat{p}_t$ at each timestep t of the input time series $\mathbf{X}$ using a model f_θ with parameter θ, i.e., $\hat{\mathbf{p}}_\theta = f_\theta(\mathbf{X})$. Thus, the temporal granularity of the sequence of probabilities $\hat{\mathbf{p}}_\theta$ is coupled with the granularity of the input sequence $\mathbf{X}$. In this dense classification setup, the training of the model—e.g., RNN and LSTM (Hochreiter & Schmidhuber, 1997)—is commonly done through backpropagation using the stepwise cross-entropy as loss function:

$$\mathcal{L}_{\text{CE}}(\hat{\mathbf{p}}_\theta, \mathbf{y}) = -\textstyle\sum_t y_t \log((\phi * \mathbf{x})_t) + (1 - y_t) \log(1 - \hat{p}_{\theta,t}). \tag{5.2}$$

A key feature of this objective function is that it views each timestep as an independent classification task (i.e., strict local focus). Indeed, in order to minimize the loss, the model is driven to maximize $\hat{p}_{\theta,t}$ at timesteps where an event was labeled ($t \in \mathcal{T}^Y$) and to minimize them for all other timesteps, independently of the nature of the neighboring timesteps:

$$\begin{aligned}\underbrace{\mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{p}}_\theta, \mathbf{y})}_{\downarrow\text{loss}} &= -\textstyle\sum_t \mathbb{1}_{[y_t=1]} \log(\hat{p}_{\theta,t}) + \mathbb{1}_{[y_t=0]} \log(1-\hat{p}_{\theta,t}) \\ &= \underbrace{-\textstyle\sum_{t\in\mathcal{T}(Y)} \log(\hat{p}_{\theta,t})}_{\uparrow\hat{p}_{\theta,t} \ \text{ for } t\in\mathcal{T}(Y)} \underbrace{-\textstyle\sum_{t\notin\mathcal{T}(Y)} \log(1-\hat{p}_{\theta,t})}_{\downarrow\hat{p}_{\theta,t} \ \text{ for } t\notin\mathcal{T}(Y)} . \end{aligned} \tag{5.3}$$

While this feature allows for an efficient learning of event representations in noise-free settings as the training can rely not only on local evidence of event occurrences but also on local patterns indicating non-events, this rigidity is very detrimental to the training process when annotations are subject to temporal misalignment. In fact, even in the presence of the slightest label misalignment (i.e., $\mathcal{T}(Y) \neq \mathcal{T}(G)$), correct predictions that match the ground-truth rather than the labels yield an infinite loss $\mathcal{L}_{\mathrm{CE}}(\mathbf{g}, \mathbf{y}) = \infty$. Besides that, the learning of meaningful representations in the presence of noise is hindered by the strict independence of timesteps induced by $\mathcal{L}_{\mathrm{CE}}$. Indeed, as the loss function does not allow to leverage labels from neighboring timesteps to learn local representations, the model has to rely on ambivalent local patterns that are sometimes concurrently labeled locally as events and non-events in the dataset. Such high levels of uncertainty negatively impact the quality of the learned representation.

In order to demonstrate the temporal detection capability of the loss function in isolation from the representation learning, we propose the following simple example:

Example (Localization Learning)**.**

Let the predictors $\mathbf{x}^{(i)}$ be of the form $x_t^{(i)} = \mathbb{1}_{[t=t^{(i)}]}$ and the unique ground-truth event occurrence $\mathcal{T}^{(i)}(G) = \{t^{(i)}\}$; by extension, using Equation 5.1 the noisy label sequence is equal to $y_t^{(i)} = \mathbb{1}_{[t=t^{(i)}+\epsilon^{(i)}]}, \epsilon^{(i)} \overset{iid}{\sim} E$. This scenario describes a situation where the event occurrence is clearly discernible in the data—no representation learning is necessary, and where the identity function is the optimal model. Given the nature of the data, the problem is similar to learning a 1D convolution filter ϕ, i.e., $\hat{\mathbf{p}}_\theta^{(i)} = f_\theta(\mathbf{x}^{(i)}) = \phi * \mathbf{x}^{(i)}$. In this setting, the optimal prediction $\mathbf{p}^{*\,(i)}$

Figure 5.2: How a model being trained with the stepwise cross-entropy is looking for evidence. The stepwise cross-entropy views each timestep as an independent classification task. While this strict focus on local patterns can be effective in noise-free settings, this rigidity of the learning is highly detrimental when the evidence in the data is not aligned with the training annotations.

that minimizes the loss $\sum_i \mathcal{L}_{\text{CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ has the form:

$$p^{*\,(i)}_{t^{(i)}+\tau} \approx P(E = \tau). \tag{5.4}$$

Proof. As shown in Appendix B.1,

$$\phi^* = \arg\min_\phi \textstyle\sum_i \mathcal{L}_{\text{CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \Leftrightarrow \phi^*(\tau) \approx P(E = \tau). \tag{5.5}$$

Then, Equation (5.4) follows from the definition of the convolution. □

Thus, in this scenario, while the predictions $\hat{\mathbf{p}}^{(i)}_\theta$ converge towards the ground-truth $\mathbf{g}^{(i)}$ in the noise-free setting (i.e., $P(E = \tau) = \mathbb{1}_{[\tau=0]}$), models are trained to infer dispersed predictions when labels are subject to temporal misalignment. This result further indicates that the dispersion of the prediction mass is given by the noise distribution E.

In conclusion, in noisy settings, models trained with the stepwise cross-entropy are not only expected to struggle to learn meaningful representations, but are also expected, given perfect representations, to yield dispersed predictions that thus are temporally ambiguous.

5.4.2 Label Smoothing

Label smoothing (i.e., applying a Gaussian filter to the point label) is a common and state-of-the-art methodology in 2D image point detection applications where spatial uncertainty must be dealt with (Tompson et al., 2014, 2015; Merget et al., 2018). This methodology is also considered to improve robustness to label misalignment in temporal applications, e.g., (Schlüter & Böck, 2014). More precisely, when applied to the stepwise cross-entropy, this approach yields the following *relaxed* loss function:

$$\begin{aligned}\mathcal{L}_{\mathrm{LS|CE}}(\hat{\mathbf{p}}_\theta, \mathbf{y} \,|\, \Phi) &= \mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{p}}_\theta, \Phi * \mathbf{y}) \\ = -\textstyle\sum_t \Big(&\underbrace{\textstyle\sum_{\tau=0}^{T} y_\tau \Phi(t-\tau)}_{(\Phi*\mathbf{y})_t} \log(\hat{p}_{\theta,t}) \\ &+ (1 - \underbrace{\textstyle\sum_{\tau=0}^{T} y_\tau \Phi(t-\tau)}_{(\Phi*\mathbf{y})_t}) \log(1-\hat{p}_{\theta,t})\Big),\end{aligned} \tag{5.6}$$

where Φ is a 1D convolutional filter, e.g., a Gaussian filter:

$$\Phi_{\sigma^2}(x) = (2\pi\sigma^2)^{-1/2} e^{-x^2/2\sigma^2}. \tag{5.7}$$

While a potentially unbounded penalization of false predictions (i.e., predictions that are wrong w.r.t the labels can produce infinite loss: $\log(0) = -\infty$) might be ideal when training with clean data, such extreme behavior can be highly detrimental when labels are subject to temporal misalignment. Indeed, in the presence of noise, highly penalizing predictions that do not match the—potentially unreliable—annotations can be counterproductive. This is reminiscent of the observation made earlier about the stepwise cross-entropy where the slightest shift in the labels can result in the correct predictions (i.e., predictions that actually match the underlying ground-truth) yielding an infinite loss. Thus, a bounded alternative based on the squared error might be preferred when dealing with high levels of noise:

$$\mathcal{L}_{\mathrm{LS|SE}}(\hat{\mathbf{p}}_\theta, \mathbf{y} \,|\, \Phi) = \sum_t \Big(\hat{p}_{\theta,t} - \underbrace{\textstyle\sum_{\tau=0}^{T} y_\tau \Phi(t-\tau)}_{(\Phi*\mathbf{y})_t}\Big)^2. \tag{5.8}$$

Once again, let us consider the example introduced in the previous section:

Example (Localization Learning, *continued*)**.**
Let ϕ, $\mathbf{x}^{(i)}$, $\hat{\mathbf{p}}_\theta^{(i)}$, $\mathbf{y}^{(i)}$ and $\mathbf{g}^{(i)}$ be defined as in the example of Section 5.4.1. Recall that this setup, with $x_t^{(i)} = \mathbb{1}_{[t=t^{(i)}]}$ and $y_t^{(i)} = \mathbb{1}_{[t=t^{(i)}+\epsilon^{(i)}]}, \epsilon^{(i)} \overset{iid}{\sim} E$, describes a scenario where the unique event occurrence is clearly discernible and where the optimal model $f_\theta(\mathbf{x}^{(i)}) = \phi * \mathbf{x}^{(i)}$ is the identity function itself. Then, the optimal prediction $\hat{\mathbf{p}}^{*\,(i)} := \hat{\phi} * \mathbf{x}^{(i)}$ that minimizes the loss $\sum_i \mathcal{L}_{\mathbf{LS|CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi)$ has the form:

$$p^{*\,(i)}_{t^{(i)}+\tau} \approx (E * \Phi)_\tau = \textstyle\sum_k P(E = k)\Phi(\tau - k) \tag{5.9}$$

Proof. As shown in Appendix B.2,

$$\begin{aligned} &\phi^* = \arg\min_\phi \textstyle\sum_i \mathcal{L}_{\mathrm{LS|CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi) \\ \iff\ &\phi^*(\tau) = (E * \Phi)_\tau = \textstyle\sum_k P(E{=}k)\Phi(\tau{-}k). \end{aligned} \tag{5.10}$$

Then, Equation (5.9) follows from the definition of the convolution. □

A similar result can be obtained for $\mathcal{L}_{\mathrm{LS|SE}}$. Thus, in comparison to $\mathcal{L}_{\mathrm{CE}}$, models optimized with smoothed labels are trained to infer even more dispersed predictions. For instance, even in a noise-free setting, the optimal predictions with respect to the loss function are dispersed over time according to the smoothing filter Φ.

Thus, despite its intuitive nature, the traditional solution of smoothing the labels presents several inherent drawbacks (see Figure 5.3) when applied to temporal point localization:

(Issue 1) As models are designed to output dispersed predictions that are spread out over several timesteps, additional tailored heuristics (e.g., peak picking (Böck et al., 2013) or complex thresholding) are required to obtain precise point predictions. Consequently, the learning of point localization is not done in an end-to-end fashion.

(Issue 2) Even advanced peak picking struggles to *disentangle* close events. For example, a single maximum might emerge in the middle of two events (see Figure 5.3), thus significantly harming the precision of the final predictions.

(Issue 3) Even in a noise-free setting, the optimal prediction at any given time does not only depend on previous event occurrences, but also on all closely upcoming events:

$$\begin{aligned} p^*_t &= \textstyle\sum_{\tau=0}^{T} y_\tau \Phi(t-\tau) \\ &= \underbrace{\textstyle\sum_{\boldsymbol{\tau\leq t-1}} y_\tau \Phi(t-\tau)}_{\text{past events}} + y_t \Phi(0) + \underbrace{\textstyle\sum_{\boldsymbol{\tau\geq t+1}} y_\tau \Phi(t-\tau)}_{\text{future events}}. \end{aligned} \tag{5.11}$$

This implies that correctly detecting an event is not enough; the context—before and after—also has to be estimated accurately. This cross-influence from other timesteps is especially problematic for causal models (i.e., models that make predictions at time t only with data up to time t), for one-sided recurrent networks, and for fully convolutional architectures with limited receptive fields. Indeed, these models have little or even no ability to integrate information from future timesteps. Thus, for example, requiring them to estimate the left tail of the label distribution might force them to learn irrelevant features preceding the actual event occurrence, leading to poor generalization.

The presence of strong label misalignment further worsens all these issues as increased noise commonly warrants increased smoothing, dispersing the label (and consequently the prediction) mass even more (e.g., Equation (5.9)). Overall, experimental evidence in Section 5.6 shows that just one of these issues can prove to be very detrimental to the noise robustness of this classical approach.

5.5 Our $\mathcal{S}$oftLoc Loss Function

5.5.1 Soft Localization Learning Loss

While the general principle of relaxing the localization learning is intuitive and potentially powerful if carefully implemented, smoothing *only* the labels is problematic, especially in causal settings. Many of the drawbacks arising from the asymmetric nature of the one-sided smoothing can however be alleviated by filtering not only the labels but also the predictions. The comparison of these two

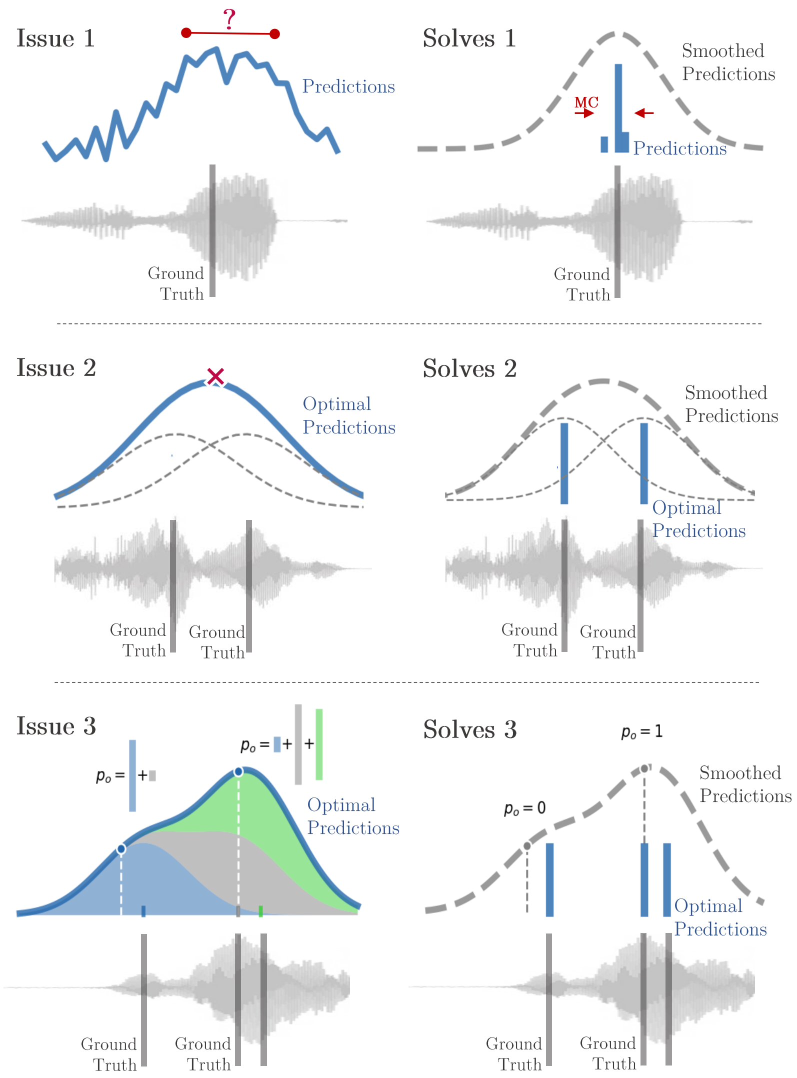


Figure 5.3: Drawbacks of label smoothing and solutions provided by our approach. **Issue 1**: ambiguous predictions of event locations require the use of additional heuristics **Issue 2**: closely grouped events cannot be easily disentangled **Issue 3**: temporal cross-influence from other timesteps requires an awareness of past and future event occurrences to make optimal predictions (e.g., left tail estimation for causal models).

smoothed processes yields a relaxed loss function for the soft learning of temporal point detection:

$$\begin{aligned}\mathcal{L}_{\text{SLL}}(\hat{\mathbf{p}}_\theta, \mathbf{y} \,|\, \Phi, \mathcal{E}) &= \mathcal{L}_{\text{LS|SE}}(\mathcal{E} * \Phi * \hat{\mathbf{p}}_\theta, \mathbf{y} \,|\, \Phi) \\ &= \textstyle\sum_t \Big(\underbrace{\textstyle\sum_{\tau,\tilde{\tau}=0}^{T} \hat{p}_{\theta,\tau} \Phi(\tilde{\tau}-\tau)\mathcal{E}(t-\tilde{\tau})}_{(\mathcal{E}*\Phi*\hat{\mathbf{p}}_\theta)_t} - \underbrace{\textstyle\sum_{\tau=0}^{T} y_\tau \Phi(t-\tau)}_{(\Phi*\mathbf{y})_t} \Big)^2 \\ &= \textstyle\sum_t \Big(\textstyle\sum_{\tau=0}^{T} (\underbrace{\textstyle\sum_{\tilde{\tau}=0}^{T} \hat{p}_{\theta,\tilde{\tau}}\, \mathcal{E}(\tau-\tilde{\tau})}_{(\mathcal{E}*\hat{\mathbf{p}}_\theta)_\tau} - y_\tau)\Phi(t-\tau) \Big)^2,\end{aligned} \tag{5.12}$$

where Φ and $\mathcal{E}$ are smoothing filters. The learning is characterized as *soft* since slight temporal shifts do not cause any abrupt increase in loss—a property that contrasts with $\mathcal{L}_{\text{CE}}$. Thus, the model's reliance on exact label locations is relaxed. We once again prefer the (bounded) squared error over the (potentially unbounded) log-based measures, especially in the presence of high misalignment levels.

Example (Localization Learning, *continued*).
Let ϕ, $\mathbf{x}^{(i)}$, $\hat{\mathbf{p}}_\theta^{(i)}$, $\mathbf{y}^{(i)}$ and $\mathbf{g}^{(i)}$ be defined as in the example of Section 5.4.1, then the optimal prediction $\hat{\mathbf{p}}^{*\,(i)}$ that minimizes the loss $\sum_i \mathcal{L}_{\mathbf{SLL}}(\phi * \mathbf{x}^{(\mathbf{i})}, \mathbf{y}^{(i)} \,|\, \Phi, \mathcal{E})$ has the form:

$$(\mathcal{E} * \mathbf{p}^{*\,(i)})_\tau \approx (E * \mathbf{g}^{(i)})_\tau, \text{ if } 1 * \mathcal{E} = 1 \text{ and } 1 * \Phi = 1 \tag{5.13}$$

Proof. See Appendix B.3. □

Regardless of the chosen filter $\mathcal{E}$, the optimal prediction is independent of the chosen smoothing filter Φ. Thus, in contrast to label smoothing, our approach can rely on heavy smoothing without causing a certain increase in dispersion of the predictions.

This example further reveals that if $\mathcal{E} = E$, then the predictions converge towards the ground-truth event locations, i.e., $p_t^{*\,(i)} \approx g_t^{(i)}$. However, while an estimate of the error distribution can be obtained by altering loss minimization and noise estimation during the training (e.g., (Patrini et al., 2017)), this theoretical result requires an exact account of the noise distribution and any deviation from it might cause prediction dispersion. Thus, in practice, while alleviating the issues observed for the label smoothing approach, the soft localization learning loss $\mathcal{L}_{\text{SLL}}$

does not fully solve them, and thus does not on its own guarantee clear-cut (i.e., without dispersion) location estimates.

5.5.2 Counting-based Sparsity Constraint

This section addresses how the sparsity-inducing ability of the Poisson-binomial loss function can be leveraged to ensure that predictions do not present any temporal ambiguity nor entanglement issues since these are not actively prevented by the soft localization learning loss $\mathcal{L}_{\text{SLL}}$ alone. An intuitive way of alleviating these potentially remaining issues is to force the model to output only one single high-probability detection per event occurrence.

We propose to achieve this prediction sparsity through the addition of explicit constraints to the optimization problem:

$$\begin{aligned} \min_\theta \quad & \mathcal{L}_{\text{SLL}}(\hat{\mathbf{p}}_\theta, \mathbf{y} \,|\, \Phi, \mathcal{E}) \\ \text{s.t.} \quad & (\|\hat{\mathbf{p}}_\theta\|_0 = c) \wedge (\hat{\mathbf{p}}_\theta \in \{0,1\}^T). \end{aligned} \tag{5.14}$$

In a nutshell, the first constraint ensures that exactly c timesteps have non-zero probability, while the second one imposes that their value is equal to 1. Thus, in practice, we would set c to the number of labeled events. (Note that the number of event occurrences is invariant to the exact event locations, and thus is unaffected by label misalignment.)

An unconstrained optimization problem can be derived by integrating these constraints as penalty functions to the objective function, e.g.,

$$\min_\theta \mathcal{L}_{\text{SLL}}(\hat{\mathbf{p}}_\theta, \mathbf{y} \,|\, \Phi, \mathcal{E}) + \lambda \underbrace{(\|\hat{\mathbf{p}}_\theta\|_0 - c)^2}_{\text{Non-Diff.}} + \textstyle\sum_t \lambda_t \hat{p}_{\theta,t}(1 - \hat{p}_{\theta,t}), \tag{5.15}$$

where the weights λ, λ_t are gradually increased during the training to progressively enforce the constraints. However, as the ℓ_0-norm in the second term is non-differentiable, a differentiable surrogate has to be introduced.

Counting Constraint

To that end, we propose using the Poisson-binomial counting loss, introduced in this work, as a differentiable surrogate for the ℓ_0-norm. Indeed, by modeling the number of predicted events as a sum of independent Bernoulli distributions with probability $\hat{\mathbf{p}}_\theta$ and the labeled count as an discrete Dirac distribution $\mathbb{1}_c$, the counting loss $\mathcal{L}_{\text{PB}}(\hat{\mathbf{p}}_\theta, c)$ (Equation 2.9) can be leveraged as a differentiable replacement for the ℓ_0-norm-based constraint:

Theorem 5.1 (Surrogate Regularization)**.**

$$\begin{cases} \min_\theta & \mathcal{L}_{\text{SLL}}(\hat{\mathbf{p}}_\theta, \mathbf{y}\,|\,\Phi, \mathcal{E}) \\ \text{s.t.} & \|\hat{\mathbf{p}}_\theta\|_0 = c \\ & \hat{\mathbf{p}}_\theta \in \{0,1\}^T \end{cases} \iff \begin{cases} \min_\theta & \mathcal{L}_{\text{SLL}}(\hat{\mathbf{p}}_\theta, \mathbf{y}\,|\,\Phi, \mathcal{E}) \\ \text{s.t.} & \mathcal{L}_{\text{PB}}(\hat{\mathbf{p}}_\theta, c) = 0. \end{cases} \tag{5.16}$$

Proof. The equivalence of the constraints follows from Theorem 3.1, while the differentiability of the constraint is achieved naturally, since the Poisson-binomial loss function is defined as the logarithm of a product of differentiable and positive functions (see Equation 2.4 and Equation 2.9). □

Finally, by updating Equation (5.15), we obtain a *differentiable* penalized objective function:

$$\min_\theta \quad \mathcal{L}_{\text{SLL}}(\hat{\mathbf{p}}_\theta, \mathbf{y}\,|\,\Phi, \mathcal{E}) + \lambda \cdot \mathcal{L}_{\text{PB}}(\hat{\mathbf{p}}_\theta, c). \tag{5.17}$$

> **Regularized Loss Function** However, as the weight λ gradually increases during training, so does the loss. In order to offset this effect—which can be detrimental to the training, we propose to optimize the following scaled loss function:
>
> $$\begin{aligned} \mathcal{L}_{\mathcal{S}\text{oftLoc}}(\underbrace{f(\mathbf{X},\theta)}_{\hat{\mathbf{p}}_\theta}, \mathbf{y}) := (1-\alpha_\tau)&\mathcal{L}_{\text{SLL}}(\hat{\mathbf{p}}_\theta, \mathbf{y}\,|\,\Phi_{\mathcal{S}_M^2}, id) \\ &+ \alpha_\tau \mathcal{L}_{\text{PB}}(\hat{\mathbf{p}}_\theta, \textstyle\sum y_i), \end{aligned} \tag{5.18}$$

where $\Phi_{\mathcal{S}_M^2}(x) := (2\pi\mathcal{S}_M^2)^{-1/2} e^{-x^2 / 2\mathcal{S}_M^2}$. In this equation, the weight α_τ regulates the predominance of the prediction sparsity regularization against the soft location learning (for training iteration τ). (Note that this constraint could not be added to $\mathcal{L}_{\text{CE}}$, $\mathcal{L}_{\text{LS|SE}}$ nor $\mathcal{L}_{\text{LS|CE}}$ as the regularization and these loss functions have conflicting objectives.)

Example (Localization Learning, *continued*)**.**

Let ϕ, $\mathbf{x}^{(i)}$, $\hat{\mathbf{p}}_\theta^{(i)}$, $\mathbf{y}^{(i)}$ and $\mathbf{g}^{(i)}$ be defined as in the example of Section 5.4.1, then the optimal prediction $\hat{\mathbf{p}}^{*\,(i)}$ that minimizes the loss $\sum_i \mathcal{L}_{\mathcal{S}\text{oftLoc}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ converges towards the ground-truth sequence $\mathbf{g}^{(i)}$, if $P(E{=}k){=}P(E{=}{-k})$.

Proof. See Appendix B.4 □

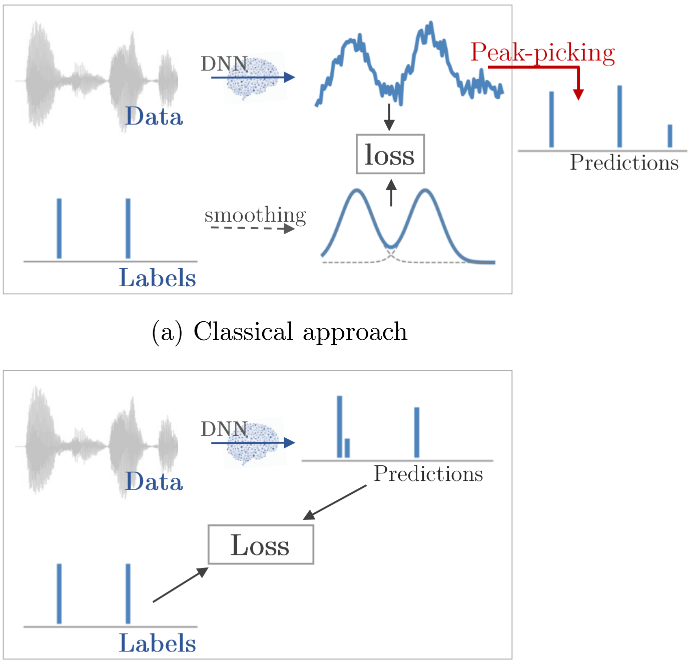
End-to-end Learning of Localization

Overall, adding this prediction sparsity constraint as a regularizer to our soft localization learning loss $\mathscr{L}_{\text{SLL}}$ allows the model to directly output unique precise impulse-like localizations (i.e., a single high-likelihood detection per event), without weakening its noise robustness properties. Thus, in contrast to more classical approaches, the proposed method offers an *end-to-end* solution to the problem of temporal localization in the presence of misaligned labels as it eliminates the need for hand-crafted components (e.g., peak picking) or post-processing (see Figure 5.4). Indeed, in such a setting, the model is given point labels and directly infers *point predictions* in an end-to-end fashion without having to explicitly resort to heatmaps nor distributions; it is only the loss function that formulates these point labels and point predictions as smoothed processes. Therefore, since the end-to-end learning paradigm is one of the key factors of the predominance of the deep learning models over classical ones (Collobert et al., 2011; Krizhevsky et al., 2012), we expect our model to better serve the task at hand.

In conclusion, our novel loss function, which combines soft localization learning with sparsity regularization, solves all the issues of label smoothing-based models presented in Section 5.4.2 (see Figure 5.3) while retaining their relaxed localization learning ability. Thus, our approach is expected to outperform existing methods—a claim that is confirmed by the experiments in the next section.

Sparsity & Uncertainty

Quantifying model and prediction uncertainties is often considered better practice than inferring a single scalar estimate or clear-cut prediction. However, while classical smoothing-based approaches (Section 5.4.2) infer more scattered location estimates, the ambiguity of their predictions does not correctly reflect the underlying uncertainty, but is rather a forced consequence of the design of the loss function (e.g., Equation (5.9)). In fact, all benchmark models use some form of post-processing (e.g., NMS) to reduce this approach-induced uncertainty, and thus our sparsity-inducing approach merely help reduce that uninformative ambiguity in an end-to-end fashion. However, there are no limitations on combining



(a) Classical approach

(b) Our end-to-end approach

Figure 5.4: Modeling differences between our SoftLoc model and the classical smoothing-based approaches. By smoothing *both* the labels and predictions, our model directly infers point predictions rather than distributions. Among other benefits, this modification allows for an end-to-end learning of temporal event localization.

our model with uncertainty quantification techniques, e.g., MC-dropout (Gal & Ghahramani, 2016).

5.5.3 Dealing with Uncertainties

The introduced softness s_M is a flexible parameter that can be leveraged to deal with different kinds of uncertainties. First, in contrast to the traditional approach of aggregating the annotations of multiple individuals (thus trading off dataset richness for noise reduction), our model can directly be trained on all individual label sequences even though they might be conflicting, since our approach can cope with noisy annotations. Second, an annotator-specific softness s_a^2 can also be implemented to model their respective reliability. Finally, an extract-specific softness $\mathcal{S}_i$ can be incorporated to capture the noise or annotation complexity of certain more challenging sequences.

This softness parameter only acts as a coarse indicator of temporal uncertainty and thus does not need to strictly match the underlying noise distribution. Indeed, experiments conducted in the section below show that the performance is robust to large variations in this hyperparameter.

5.6 Experiments

In order to demonstrate the effectiveness and flexibility of our approach, a broad range of challenging experiments are conducted (video action detection, times series detection, and music event detection). The code and the experiment details are freely available[1].

5.6.1 Golf Swing Sequencing in Video

In this section, we replicate the video event detection experiment from (McNally et al., 2019) using either the original cross-entropy ($\mathcal{L}_{\text{CE}}$), the label smoothing benchmarks ($\mathcal{L}_{\text{LS}|\text{CE}}$ and $\mathcal{L}_{\text{LS}|\text{SE}}$), or our proposed loss ($\mathcal{L}_{\mathcal{S}\text{oftLoc}}$) for training (not changing anything else). The task consists in the precise detection (within a one frame tolerance) of eight different classes of golf swing events in video extracts (e.g., address and impact). To assess robustness to noisy annotations, rounded normally distributed misalignments (i.e., $\epsilon_m \sim \lfloor \mathcal{N}(0, \sigma^2) \rceil$) are artificially applied to the event timestamps of the training samples, while the test labels are kept intact for unbiased inference.

Experiment Characteristics

Among other aims, this experiment helps measure the impact of *prediction ambiguity* (i.e., Issue 1 in Section 5.4.2) on the performance of the $\mathcal{L}_{\text{LS}|\text{CE}}$ and the $\mathcal{L}_{\text{LS}|\text{SE}}$ approaches. Indeed, as video extracts in the dataset contain exactly one occurrence of each event type, most of the issues highlighted in Section 5.4.2 do not occur (e.g., no prediction entanglement, no cross-influence from future events,

[1] https://github.com/SchroeterJulien/AAAI-2021-Learning-Precise-Temporal-Point-Event-Detection-with-Misaligned-Labels

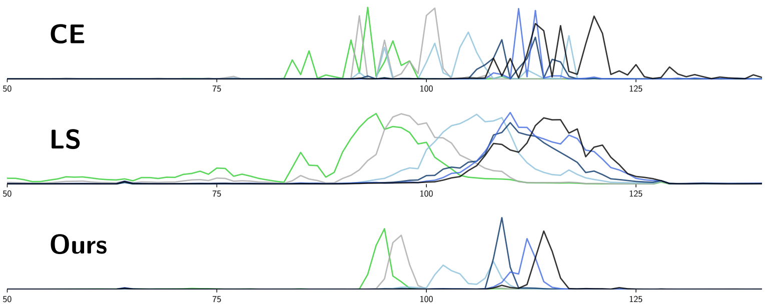


Figure 5.5: Out-of-Sample Golf Swing Action Predictions. **Ours**: sharp predictions, **LS** (label smoothing): ambiguous predictions, **CE**: multiple peaks. (Test sequence: 0, split: 1, noise level: $\sigma = 3$ frames.)

and no complex peak-picking required). In addition, the model architecture in this experiment includes a bidirectional RNN to model temporal dependencies, enabling the estimation of the two tails of the event distribution (i.e., no Issue 3). Thus, in this task, the only defining component that distinguishes the label smoothing benchmarks ($\mathcal{L}_{\mathrm{LS|CE}}$, $\mathcal{L}_{\mathrm{LS|SE}}$) from our loss function is the potential ambiguity of prediction locations.

Golf Swing Sequencing Results

Table 5.1a confirms the intuitive understanding that the cross-entropy ($\mathcal{L}_{\mathrm{CE}}$) is not well suited to effectively deal with label misalignment. Indeed, we observe here that attempting to strictly mimic unreliable annotations leads to poor generalization performance. The results further reveal that even just one of the issues presented in Section 5.4.2 (here, prediction ambiguity) can negatively impact the prediction accuracy, as shown by the significant performance gap between our approach ($\mathcal{L}_{\mathcal{S}\mathrm{oftLoc}}$) and the label smoothing benchmarks ($\mathcal{L}_{\mathrm{LS|CE}}$, $\mathcal{L}_{\mathrm{LS|SE}}$) in noisy settings. Indeed, while our approach infers sharp predictions, the predictions yielded by the classical label-smoothing benchmarks ($\mathcal{L}_{\mathrm{LS|SE}}$ and $\mathcal{L}_{\mathrm{LS|CE}}$) are highly ambiguous as illustrated in Figure 5.5. In strict settings with low error tolerance, the dispersion of the predictions of label smoothing-based models, theoretically highlighted in Section 5.4.2 and observed in this experiment, leads to suboptimal performance. (Note that even more clear-cut point predictions could be achieved for $\mathcal{L}_{\mathcal{S}\mathrm{oftLoc}}$ by training the model further than the 10k iterations set by McNally et al. (2019), which would allow for full convergence of the counting loss function.)

Table 5.1: Golf Swing Action Detection. Performance comparison of the same model when trained with various training losses ($\mathcal{L}_{\text{CE}}$, $\mathcal{L}_{\text{LS|CE}}$, $\mathcal{L}_{\text{LS|SE}}$, and $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$) on the golf swing sequencing task (McNally et al., 2019) with respect to various label-misalignment levels $\lfloor \mathcal{N}(0, \sigma^2) \rceil$ (σ in number of frames). The cross-validated (4-folds) mean accuracy is reported.

(a) Bidirectional RNN

	$\sigma = 0$	1	2	3	4
$\mathcal{L}_{\text{CE}}$	68.1	60.4	51.6	43.1	36.9
$\mathcal{L}_{\text{LS\|CE}}$	66.7	64.7	59.1	54.8	49.1
$\mathcal{L}_{\text{LS\|SE}}$	**69.1**	66.2	60.6	54.7	50.7
$\mathcal{L}_{\mathcal{S}\text{oftLoc}}$	67.2	**68.0**	**65.6**	**58.6**	**54.2**

(b) Unidirectional RNN (i.e., *causal model*)

	$\sigma = 0$	1	2	3	4
$\mathcal{L}_{\text{CE}}$	62.8	57.2	47.3	40.9	35.3
$\mathcal{L}_{\text{LS\|CE}}$	57.0	54.2	50.6	46.4	42.5
$\mathcal{L}_{\text{LS\|SE}}$	61.3	59.5	55.2	49.9	46.5
$\mathcal{L}_{\mathcal{S}\text{oftLoc}}$	**63.0**	**62.2**	**59.3**	**54.9**	**50.7**

The same conclusion can be drawn from the additional experiment conducted using a (causal) unidirectional RNN, instead of the original bidirectional architecture (see Table 5.1b for results). Our approach achieves the best overall performance on all noise levels, including the noise-free case $\sigma = 0$. These results demonstrate that the theoretical advantages of our approach (see Section 5.5) can translate to a significant increase in performance, especially for causal applications.

In Chapter 4, the counting loss $\mathcal{L}_{\text{PB}}$ was shown to be able to successfully train *alone* precise temporal models in a weakly-supervised manner. However, this is not the case for this experiment. Aside from the complexity of the task, the main issue resides in the nature of the dataset. Indeed, the number of event occurrences for all classes is exactly equal to 1 in all sequences. The model thus does not even have to learn to count to estimate the correct number of event occurrences since this number is a constant in the data. For example, in this setup, trivially triggering every time at the first timestep only, would be an optimal solution with respect to the Poisson-binomial counting loss. Consequently, training the golf event sequencing model with the counting loss *alone* is bound to fail, thus highlighting the importance of $\mathcal{L}_{\mathrm{SLL}}$ for learning precise localization.

5.6.2 Wearable Sensors Time Series Detection

The timely detection of events in healthcare time series is a crucial challenge to improve medical decision making. The task tackled in this section consists in the precise temporal detection of smoking episodes using wearable sensors features from the puffMarker dataset (Saleheen et al., 2015). The noise robustness analysis replicates the experiment conducted in (Adams & Marlin, 2017), which involves normally distributed label misalignment (i.e., $\epsilon_m \sim \mathcal{N}(0, \sigma^2)$) and no error tolerance (i.e., detections have to be perfectly aligned with the ground-truth to be considered as correct).

Model and Benchmarks

As the focus is set on robustness rather than raw performance, the neural architecture is kept extremely simple: a 14-node fully connected layer followed by a 14-unit (unidirectional) LSTM and a final fully connected layer with softmax activation. The stepwise cross-entropy ($\mathcal{L}_{\mathrm{CE}}$), the label smoothing benchmarks ($\mathcal{L}_{\mathrm{LS|CE}}$ and $\mathcal{L}_{\mathrm{LS|SE}}$), and our ($\mathcal{L}_{\mathcal{S}\mathrm{oftLoc}}$) loss function (with $\mathcal{S}_M$=3 frames) are used for training. The statistical LR-M model proposed by Adams and Marlin (2017), which was developed to achieve strong robustness to temporal misalignment of labels on this particular dataset, is also considered as a benchmark.

Experiment Characteristics

Each timestep in this dataset represents a full respiration cycle. Thus, multiple smoking episodes can occur consecutively (i.e., one after another without interruption), which contrasts with the sparse distribution (over time) of golf events in the previous experiment. Such dense sequences of events in conjunction with a causal architecture and a very strict tolerance help assess how Issue 3 (i.e., cross-influence between timesteps) might penalize the performance of the $\mathcal{L}_{\mathrm{LS|SE}}$ benchmark, unlike our method.

Smoking Puff Detection Results

The results, produced using ten 6-fold (leave-one-patient-out) cross-validations, are summarized in Table 5.2. Not only does training with the proposed $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ loss function yields a strong improvement in robustness when compared to the standard cross-entropy (CE), but our simple recurrent model also significantly outperforms the robust LR-M model on all metrics.

In addition to normally distributed label misalignment, more challenging noise patterns are also investigated: binary constant length shifting of labels ($\pm\delta$ steps with equal probability) denoted $\mathcal{B}(-\delta, \delta)$ and skewed-normal noise distribution $\mathcal{SN}(0, \sigma^2, \alpha = -2)$. Aside from exhibiting strong overall performance on all noise levels, our approach displays scores with low standard deviations which underlines the consistency and robustness of the learning process. These observations hold for all noise distributions confirming that the Gaussian filtering does not have to match the actual noise distribution of the data to be effective. Indeed, the smoothing distribution only acts as a means of relaxing the dependence of the learning on the exact location of the labels, and not as a model for the underlying noise (see Section 5.5).

As expected, the label smoothing benchmarks ($\mathcal{L}_{\text{LS}|\text{CE}}$, $\mathcal{L}_{\text{LS}|\text{SE}}$) yield poor overall results on this task. In fact, the causal architecture makes the learning with these loss functions especially difficult, as the model is unable to properly learn the target smoothed labels given that it does not have the ability to leverage crucial information from future timesteps (see Equation 5.11 and Issue 3). Even in noise-free settings, the gap between optimal predictions and optimal attainable predictions is further widened by the fact that the multiple events can occur consecutively (see Figure 5.6). In contrast, the optimal predictions for the $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ are the labels themselves, regardless of the density of event occurrences.

5.6.3 Piano Onset Experiment

We make use of the piano onset detection dataset introduced in Section 4.3.2 to further assess the effectiveness of our proposed approach. Recall that piano transcription and, more specifically, piano onset detection is a difficult problem, as it requires precise and simultaneous detection of hits from 88 different polyphonic channels. Similarly to Section 4.3.2, we reproduce the experiment from Hawthorne et al. (2017) using the MAPS database (Emiya et al., 2010) and we

Table 5.2: Smoking Puff Detection. Comparison of LR-M (Adams & Marlin, 2017) and the deep model trained with $\mathcal{L}_{\text{CE}}$, $\mathcal{L}_{\text{LS|CE}}$, $\mathcal{L}_{\text{LS|SE}}$, and $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ with respect to misalignment distributions $\lfloor\mathcal{N}(0,\sigma^2)\rceil$, $\mathcal{B}(-\delta,\delta)$ and $\lfloor\mathcal{SN}(0,\sigma^2,\alpha=-2)\rceil$. Reported metrics are mean and standard deviation of ten 6-fold cross-validated F_1-scores.

		$\delta,\sigma = 0$	1	2	3	4
$\lfloor\mathcal{N}\rceil$	LR-M	93.0 (3.2)	80.6 (8.6)	65.9 (17.4)	64.0 (15.6)	55.0 (19.7)
	$\mathcal{L}_{\text{CE}}$	92.6 (2.9)	55.3 (16.2)	36.0 (15.6)	28.9 (17.0)	25.8 (16.2)
	$\mathcal{L}_{\text{LS\|CE}}$	63.9 (7.9)	58.7 (7.4)	50.6 (9.1)	49.5 (9.0)	43.3 (9.2)
	$\mathcal{L}_{\text{LS\|SE}}$	63.5 (9.5)	59.2 (6.3)	54.6 (5.9)	49.4 (7.8)	46.3 (8.5)
	$\mathcal{L}_{\mathcal{S}\text{oftLoc}}$	**93.1** (2.5)	**90.6** (3.4)	**87.8** (4.1)	**83.6** (5.2)	**79.0** (6.9)
$\mathcal{B}$	LR-M	—	65.5 (14.5)	54.9 (20.4)	44.1 (19.7)	51.8 (19.8)
	$\mathcal{L}_{\text{CE}}$	—	41.7 (15.3)	28.3 (14.5)	26.6 (15.3)	22.8 (15.1)
	$\mathcal{L}_{\text{LS\|CE}}$	—	60.7 (6.7)	53.0 (8.8)	43.7 (10.1)	34.8 (13.0)
	$\mathcal{L}_{\text{LS\|SE}}$	—	45.2 (8.3)	54.7 (9.2)	45.1 (11.6)	35.4 (11.8)
	$\mathcal{L}_{\mathcal{S}\text{oftLoc}}$	—	**90.8** (3.3)	**87.0** (4.7)	**81.7** (7.2)	**72.4** (10.1)
$\lfloor\mathcal{SN}\rceil$	LR-M	—	79.7 (10.4)	68.3 (15.6)	61.4 (20.7)	54.7 (18.2)
	$\mathcal{L}_{\text{CE}}$	—	57.6 (16.6)	27.8 (13.7)	20.0 (13.9)	16.1 (14.4)
	$\mathcal{L}_{\text{LS\|CE}}$	—	53.8 (9.8)	49.6 (8.5)	43.9 (10.7)	41.1 (8.6)
	$\mathcal{L}_{\text{LS\|SE}}$	—	57.0 (8.2)	52.1 (8.4)	48.3 (7.9)	44.4 (9.6)
	$\mathcal{L}_{\mathcal{S}\text{oftLoc}}$	—	**90.4** (3.9)	**88.2** (5.0)	**84.2** (6.1)	**79.1** (9.0)

only consider onsets for the comparison. Once again, to evaluate the robustness of our model, the training labels are artificially perturbed according to a normal distribution ($\epsilon_m \sim \mathcal{N}(0,\, \sigma^2)$).

Experiment Characteristics

In contrast to the wearable sensor experiment in Section 5.6.2, events are more sparsely distributed and the architecture includes temporal convolutions (i.e., it is not a fully causal model). Consequently, the label smoothing benchmarks are expected to be less impacted by Issue 3. However, as a piano note can be played multiple times within a very short time span, prediction entanglement (Issue 2) might arise when training with such one-sided-smoothing-based approaches, e.g., $\mathcal{L}_{\text{LS|SE}}$.

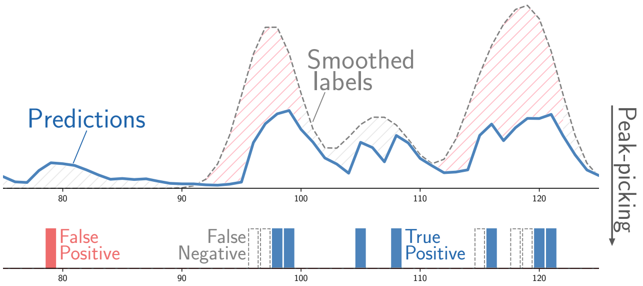


Figure 5.6: Out-of-sample smoking predictions for the label-smoothing approach. The model struggles to estimate the left tail of the event distribution, especially when multiple events occur consecutively. (Patient: 16, seq.: 2, $\sigma = 2$).

Benchmarks

For the comparison, three additional classical benchmarks, based on a model proposed by Hawthorne et al. (2017) that shows state-of-the-art performance on clean data, are considered: first, the original model itself which is highly representative of models aiming for optimal performance with little regard for annotation noise (ORIGINAL); second, a version with extended onset length (i.e., target smearing) (EXTENDED); a version trained with the soft bootstrapping loss proposed by Reed et al. (2014) instead of the cross-entropy for increased robustness.

Model Architecture, Training, and Evaluation

The network in this experiment is highly reminiscent of the architecture proposed in Section 4.3.2: six convolutional layers (representation learning) followed by a 128-unit LSTM (temporal dependencies learning) and two fully-connected layers (prediction mapping). Once again, the network is trained using mel-spectrograms (Stevens et al., 1937) and their first derivatives stacked together as model input, while data augmentation in the form of sample rate variations is applied for increased robustness and performance. The models are evaluated on the *noise-free* test set using the *mir_ eval* library (Raffel et al., 2014) with a 50ms tolerance as in (Hawthorne et al., 2017). ($\mathcal{S}_M = 100\text{ms}$, $\alpha_\tau = \max(\min(\frac{\tau - 10^5}{10^5}, .9), .2)$.)

Table 5.3: Piano Onset Detection. Performance comparison of models trained with $\mathcal{L}_{\text{LS|SE}}$, $\mathscr{L}_{\text{SLL}}$, and $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ ($\mathcal{S}_M = 100\text{ms}$) as well as the diverse classical benchmarks (Hawthorne et al., 2017) with respect to label misalignment distribution $\epsilon_m \sim \mathcal{N}(0, \sigma^2)$. The mean F$_1$-score is reported.

	$\sigma =$ 0ms	50ms	100ms	150ms	200ms
Hawthorne (ORIGINAL) (2017)	**82.1**	38.5	2.0	0.5	0.2
Hawthorne (EXTENDED)	77.7	68.0	30.7	9.2	3.9
Hawthorne (BOOTSTRAP)	79.1	74.2	32.5	15.4	6.9
$\mathcal{L}_{\text{LS\|SE}}$	73.1	70.5	59.2	41.3	28.0
$\mathcal{L}_{\text{SLL}}$	76.1	76.0	75.1	66.9	46.9
$\mathcal{L}_{\mathcal{S}\text{OFTLOC}}$	76.0	**76.3**	**75.9**	**74.0**	**73.7**

Piano Onset Detection Results

As summarized in Table 5.3, our proposed approach $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ displays strong robustness against label misalignment: in contrast to all benchmarks, the performance appears almost invariant to the noise level. For instance, at $\sigma = 150\text{ms}$, only 26% of training labels lie within the 50ms tolerance (see Figure 5.7 for illustration); in such a context, the score achieved by our model $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ (i.e., $\sim 75\%$) is unattainable for classical approaches, which do not take label uncertainty into account and attempt to strictly fit the noisy annotations. While standard tricks, such as label smoothing ($\mathcal{L}_{\text{LS|SE}}$) or label smearing (EXTENDED), slightly improve noise robustness, their effectiveness is limited. The results also reveal that, as the noise level increases, the addition of the prediction sparsity regularization $\mathcal{L}_{\text{PB}}$ to $\mathcal{L}_{\text{SLL}}$ is crucial to achieve strong robustness. Finally, a fixed parameter set is used throughout this experiment, which explains the small performance gap between our approach and (Hawthorne et al., 2017) for the noise-free case. This could be partially remedied by adapting the loss settings (e.g., $\alpha_\tau = 1$ and $\mathcal{S}_M^2 \rightarrow 0\text{ms}$).

To further illustrate the complexity of the localization task when annotations are subject to misalignment, we compare the training labels with the actual ground-truth event locations. Figure 5.7.b displays an example of the quality of the training labels. Obviously, in the noise-free setting (i.e., $\sigma = 0\text{ms}$), the localization is spotless as the training labels and the ground-truths are identical. However, as the noise level increases, the proportion of labels that stay within the 50ms tolerance window decreases significantly. More precisely, the performance (i.e., F_1-score) corresponding to using the labels themselves as predictions is

68.2%, 39.8% and 23.7% for σ equal to 50ms, 100ms and 200ms respectively. This contrasts with the performance of our approach, which appears almost invariant to the noise level (see Figure 5.7.a).

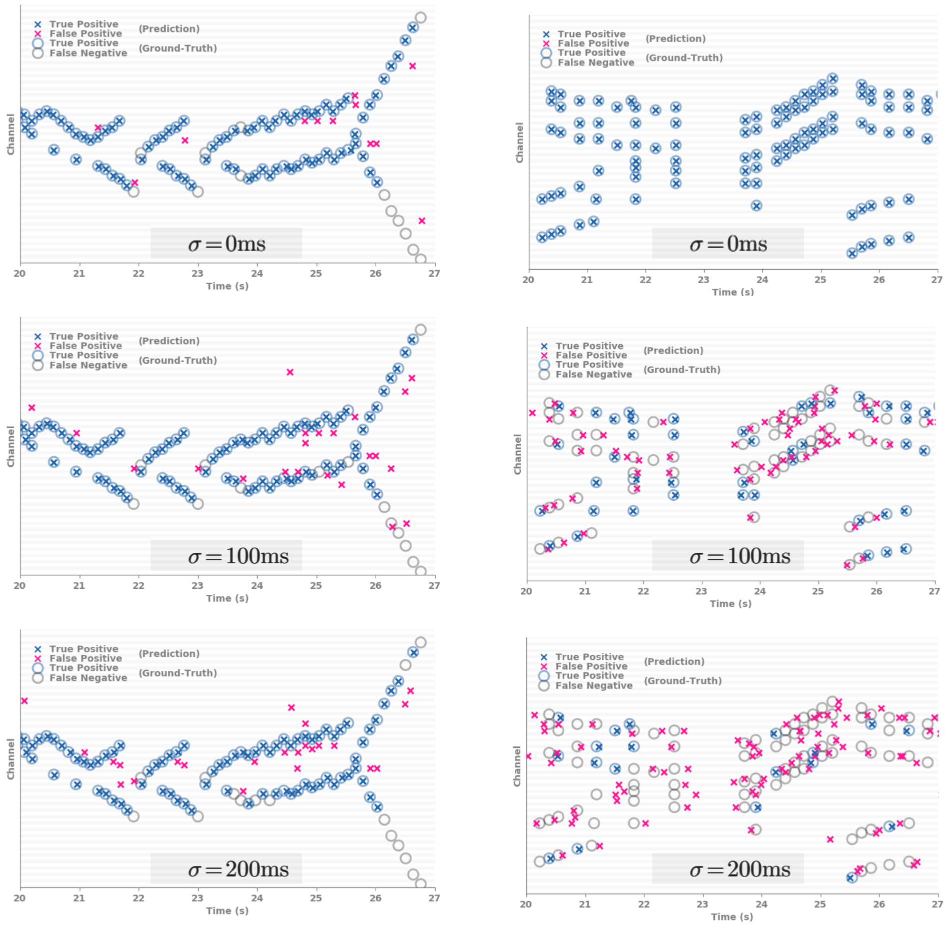
Ablation Study

To assess the usefulness of the different components of $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$, we repeat the above experiments keeping only individual parts of the loss function. Table 5.4 reveals that $\mathcal{L}_{\text{SLL}}$ is the main driver of performance in noise-free settings, while $\mathcal{L}_{\text{PB}}$ ensures stability under increased label misalignment. (A simple threshold-based peak-picking algorithm was implemented to infer localization from the dispersed mass produced by $\mathcal{L}_{\text{SLL}}$.) Overall, while each loss individually yields reasonable predictions, only the combined $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ yields both competitive scores in noise-free settings and strong robustness to temporal misalignment. It is however important to note that the good performance of the counting loss on piano onset detection is a direct consequence of the high discernibility and uniformity of piano onsets. The loss function is unable to effectively learn golf swing sequencing (Section 5.6.1) or time series detection (Section 5.6.2) on its own without any additional localization supervision.

Table 5.4: Ablation Study. Piano onset detection performance of our model trained with loss functions $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ ($\mathcal{S}_M$=100ms), $\mathcal{L}_{\text{SLL}}$ and $\mathcal{L}_{\text{PB}}$ respectively in various noise level settings. The mean F$_1$-score is reported.

	$\sigma = 0$ms	50ms	100ms	150ms	200ms
$\mathcal{L}_{\text{SLL}}$ ($\alpha_\tau = 0$)	76.06	76.00	75.10	66.88	46.91
$\mathcal{L}_{\text{PB}}$ ($\alpha_\tau = 1$)	71.59	73.04	68.69	70.33	67.26
$\mathcal{L}_{\mathcal{S}\text{OFT}\text{LOC}}$	**76.88**	**76.34**	**75.86**	**74.87**	**73.68**

The results obtained when training with $\mathcal{L}_{\text{PB}}$ *alone* are remarkable when compared to the previous experiments, where the loss failed to successfully train the model to localize events in a weakly-supervised manner. This high performance of the weakly-supervised approach can be attributed to several factors. First, piano onset detection is characterized by clear-cut and often easily detectable events since note onsets are bursts of energy that are quite salient on the spectrograms. In addition, the dataset presents a rich variety of different count values—contrasting



(a) Out-of-sample predictions of our $\mathcal{L}_{\text{SoftLoc}}$ model trained on data subject to various noise levels. *(Schubert—Piano Sonata in A minor, D 784, Opus 143, 3. Mov)*

(b) In-sample performance of the noisy training labels themselves (*as predictions*) when compared to the clean ground-truth. *(Liszt—Hungarian Rhapsody No. 10)*

Figure 5.7: Robustness of Predictions to Noisy Labels. Difference between (a) the consistency of the predictions and (b) the quality of provided labels for training across various noise levels σ, ranging from noise-free ($\sigma = 0$ms) to extremely noisy ($\sigma = 200$ms).

with the golf event sequencing dataset in Section 5.6.1 where the count was a constant. The model is thus bound to learn to recognize non-trivial patterns in the data in order to be able to count correctly. These factors make piano onset detection especially suitable for weakly-supervised counting-based learning.

5.6.4 Drum Detection Experiment

The softness $\mathcal{S}_M$ is a defining model hyperparameter. In this section, 210 runs of the same drum detection experiment are conducted with varying noise and softness levels in order to highlight the correlation between this key hyperparameter, label noise, and the final localization performance.

To that end, we modify one of the drum detection experiments proposed by Wu et al. (2018), see Section 4.3.1. More specifically, the experiment is conducted on the D-DTD *Eval Random* drum detection task based on the IDMT-SMT-Drums dataset (Dittmar & Gärtner, 2014). Recall that the specific goal of this task is the correct temporal detection of three different classes of drum hits—hi-hats, kick drums, and snare drums—within a 50ms tolerance window. The network, the training, and the evaluation are similar to that of the weakly-supervised drum detection experiment conducted in Section 4.3.1. Once again, to perform the robustness evaluation, the training labels are artificially perturbed, while the test annotations are kept intact. For each run, the label noise level σ (i.e., $\epsilon_m \sim \mathcal{N}(0, \sigma^2)$) and the softness $\mathcal{S}_M$ are uniformly sampled at random from $[0\text{ms}, 100\text{ms}]$ and $[0\text{ms}, 150\text{ms}]$ respectively. Please refer to the provided implementation for the remaining model and training specifications (e.g., learning rate: 10^{-4}, iterations: 1.5×10^5).

Drum Detection Results

The results of the 210 runs are displayed in Figure 5.8. A Gaussian Nadaraya-Watson kernel regression (Nadaraya, 1964; Watson, 1964) is used to interpolate the F_1-score, offering a detailed view of the model's response to varying label noise levels. This figure not only confirms the model's high robustness to label misalignments, but also reveals that these results are *very robust to changes in the softness level.* Indeed, a wide range of softnesses yield optimal performance (i.e., as long as $\mathcal{S}_M \geq \sigma$). Robustness considerations aside, our $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ model displays an outstanding overall performance with F_1-scores over 95% across all noise

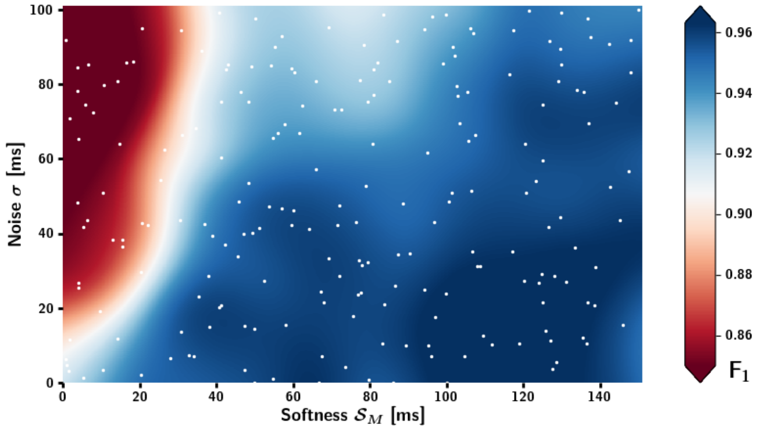


Figure 5.8: Noise Robustness and Hyperparameter Sensitivity in Drum Detection. Drum detection performance with respect to model softness $\mathcal{S}_M$ *(x-axis)* and label noise σ *(y-axis)*. F_1-scores are Gaussian Nadaraya-Watson estimates based on 210 runs (white dots).

levels; the model—even when trained on extremely noisy labels (e.g., $\sigma = 100$ms)—outperforms several standard benchmarks (Wu et al., 2018) which were trained on noise-free training samples ($\sigma = 0$ms).

Noise-free Comparison

In clean settings (i.e., $\sigma = 0$ms), the benchmark models have a clear advantage as they correctly assume noise-free labels. Despite this, our $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ model achieves state-of-the-art performance on three different metrics (KD, HH, overall precision) demonstrating that robustness does not come at the expense of raw localization performance (see results in Table 5.5).

5.7 Conclusion

In this chapter, we introduce a novel loss function that allows for the training of precise temporal localization models even in the presence of poorly aligned annotations. In contrast to the traditional cross-entropy, our loss function does not attempt to strictly mimic the given annotations, but rather relaxes the

Table 5.5: *Noise-free* Drum Detection. Comparison of our $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ model with $s_M = 100$ms and state-of-the-art models evaluated in (Wu et al., 2018) on the clean D-DTD *Eval Random* task ($\sigma = 0$ms). The mean F_1-score is reported.

Method	KD	SD	HH	Pre	Rec	F_1
RNN	97.2	92.9	97.3	95.7	96.9	95.8
tanhB	95.4	93.1	97.3	93.9	97.1	95.3
ReLUts	86.6	93.9	97.7	92.7	95.0	92.7
lstmpB	98.4	**96.7**	97.4	97.7	**97.6**	**97.5**
GRUts	91.4	93.2	96.2	91.8	97.2	93.6
$\mathcal{L}_{\mathcal{S}\text{oftLoc}}$	**98.6**	95.7	**97.8**	**98.3**	97.2	97.4

reliance of the learning on the exact event locations. While a softer learning of event localization is already made possible through classical heuristics (e.g., label smoothing), we show that these approaches inherently suffer from multiple drawbacks (e.g., entanglement and ambiguity of predictions). We solve these issues by directly inferring point predictions which are learned through comparison of the smoothed labels and smoothed predictions and by leveraging *counting* as an additional means of supervision.

We demonstrate the effectiveness of our simple approach in a number of challenging tasks (i.e., video action detection, time series event detection, music onset detection), in which our $\mathcal{L}_{\mathcal{S}\text{oftLoc}}$ loss function exhibits state-of-the-art robustness without compromising performance on clean training data. Experiments further reveal that these results not only are robust to large variations in the main hyper-parameter of the loss function (i.e., softness s_M), but also hold for a wide range of temporal noise distributions. As the proposed loss function is agnostic to the underlying network, it can be used as a loss replacement for the classical stepwise cross-entropy in almost any architecture to increase robustness to temporal noise, thus allowing for a wide array of applications.

Above all, this chapter shows how the sparsity-inducing properties of the Poisson-binomial counting loss—highlighted in Section 3—can be leveraged to enforce *sparsity constraints* in an end-to-end fashion. The integration of prediction sparsity directly in the training process thus alleviates the need for sub-optimal maximum-picking heuristics and, by extension, allows for a fully *end-to-end* robust learning of point event detection.

The use of non-differential operations to obtain sparse predictions is ubiquitous in computer vision applications. Thus, in order to assess further the versatility of the proposed count-based sparsity regularization, the next chapter applies this principle to the difficult task of learning multi-instance sub-pixel point localization in images.

Chapter 6

Learning Multi-Instance Sub-pixel Point Localization

Based on *Learning Multi-Instance Sub-pixel Point Localization*, Schroeter J, Tuytelaars T, Sidorov K, Marshall D, ACCV 2020

The previous chapter demonstrated how the Poisson-binomial counting loss function can be applied to enforce sparsity constraints in an end-to-end manner. This direct integration of prediction sparsity in the learning process was shown to constrain models to directly infer a sparse set of predictions, thereby alleviating the need for any post-processing operations.

In this chapter, the sparsity-inducing ability of Poisson-binomial counting is applied to the domain of computer vision and, more specifically, to the problem of multi-instance sub-pixel point localization (i.e., estimating the coordinates of multiple point objects with precision beyond pixel accuracy) In fact, post-processing operations such as non-maximum suppression (NMS) are a ubiquitous solution to obtain final sparse predictions in many computer vision applications. However, the common inability to integrate these non-differential heuristics directly into the training process (with the notable exception of Henderson & Ferrari (2016) and Hosang et al. (2017), see discussion in Section 7.3) can be highly detrimental to the learning of complex *high-precision* tasks such as sub-pixel localization. In contrast, this chapter highlights once again how instance counting—as a regularization to a novel loss function designed to learn sub-pixel localization—can by itself ensure that models directly infer multiple clear-cut and unambiguous point estimates. This *end-to-end* learning of *sparse* sub-pixel localization alleviates the need for any post-processing, and thus inherently increases the spatial precision of the learned predictions—sometimes far beyond pixel precision—as demonstrated by the experiments on single-molecule localization microscopy, checkerboard corner detection, and even sub-frame event detection in sports videos conducted in Section 6.4.

Sparsity considerations aside, this chapter also shows how regularizing the training with the Poisson-binomial counting loss function can lead to improved convergence speed.

6.1 Introduction

Sub-pixel point localization (e.g., sub-pixel detection of molecule locations in diffraction-limited microscopy images) is a challenging task that is characterized by the discrepancy between the precision required of the point predictions and the granularity of the input image. In this context, the standard paradigm (Tompson et al., 2015; Newell et al., 2016; Wei et al., 2016; Xiao et al., 2018; Merget et al., 2018) of operating directly on the discrete space defined by the pixel locations (e.g., discrete heatmap-matching), and thus of coupling the precision of the detections to the input resolution, is trivially not sufficient to infer point detections beyond pixel precision.

Several methods have thus emerged to extend the classical discrete setup to allow for sub-pixel capabilities (Papandreou et al., 2017; Neumann & Vedaldi, 2018; Nibali et al., 2018; Sun et al., 2018; Fieraru et al., 2018; Graving et al., 2019; Luvizon et al., 2019; Tai et al., 2019; Zhang et al., 2020). The majority of these approaches, however, work on the assumption that there is *exactly one* instance per object class. By restricting the setup to single instance localization, the point location can be inferred, for example, through continuous spatial density estimation (Neumann & Vedaldi, 2018), weighted integration (Nibali et al., 2018; Sun et al., 2018; Luvizon et al., 2019), or displacement field estimation (Papandreou et al., 2017). These methods find direct application in human pose estimation (Tompson et al., 2015; Newell et al., 2016; Wei et al., 2016; Xiao et al., 2018) and facial landmark detection (Merget et al., 2018; Yang et al., 2017), where the single instance assumption is fulfilled through image cropping and assigning each landmark to a different prediction class. However, the uniqueness assumption they rely on is often too constraining in other scenarios, especially in multi-instance sub-pixel localization.

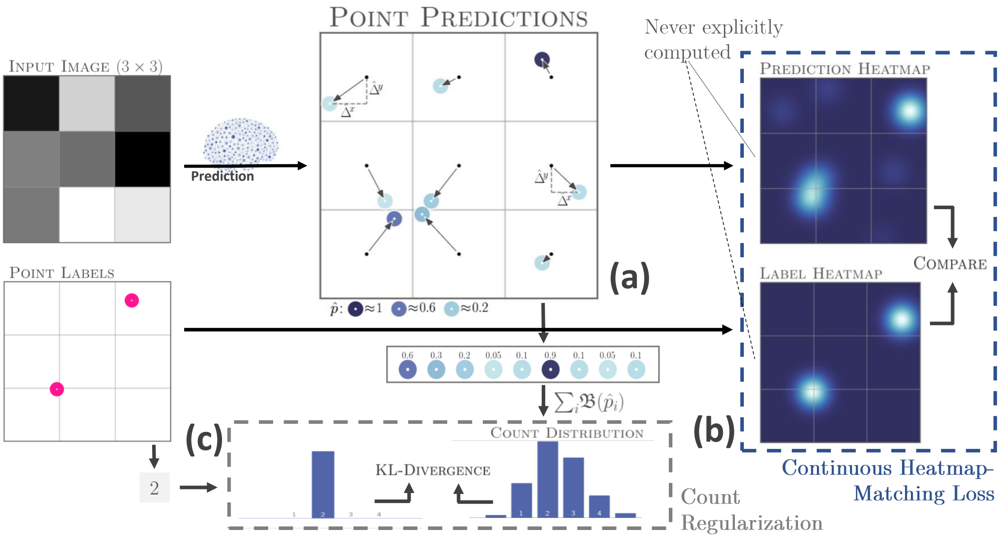


Figure 6.1: Model overview. **(a)** The model infers numerous point predictions through dense offset regression. **(b)** The point estimates are compared to the label locations through *continuous* heatmap-matching. **(c)** The predicted count is compared against the number of labeled objects (count-regularization). As the heatmaps are never explicitly determined, the loss is computed with infinite spatial resolution.

In practice, multi-instance sub-pixel point localization is relevant to various fields. For instance, in single-molecule localization microscopy (Sage et al., 2015; Nehme et al., 2018), a precise and useful account of molecule locations requires sub-pixel localization capabilities, as the resolution of the input image is limited by inherent sensor properties (e.g., diffraction-limited systems (Born & Wolf, 1997)). Additionally, as hundreds of molecules can emit light at the same time, successful models have to be able to detect multiple instances in dense settings (i.e., potentially more than one instance per pixel). In camera calibration, an accurate estimation of the camera parameters requires an extremely precise checkerboard corner detector (Placht et al., 2014; Hu et al., 2019). Thus, the ability to infer multi-instance sub-pixel corner locations is especially relevant to the effective calibration of low-resolution cameras. In these two examples, the instance uniqueness assumption does not hold, and thus the problem calls for the development of models that are able to detect and disentangle, with great precision, the locations of multiple objects (of the same class), which might even lie within the same pixel.

In this chapter, we introduce a novel model that learns—in an end-to-end fashion—to directly output one single clear-cut and spatially precise *point* estimate in $\mathbb{R}^2$ per point label. More precisely, the model infers point localizations through dense offset regression (comparable to (Neumann & Vedaldi, 2018; Papandreou et al., 2017)) and is trained using a novel loss function based on a *continuous* generalization of heatmap-matching, which helps bypass any issue induced by space discretization (see Section 6.3.2). Similar to Section 5.5.2, we further ensure that the model learns to output a unique high probability point estimate per point label through counting-based sparsity regularization (see Section 6.3.3). (See Figure 6.1 for an overview of the model.) Overall, by obviating the need for post-processing operations such as non-maximum suppression (Papandreou et al., 2017) or maxima refinement (Graving et al., 2019) which are set to deteriorate the accuracy of the predictions (see Section 6.3.3) and by inferring spatially unambiguous point predictions, our approach offers an effective solution to the challenging problem of multi-instance sub-pixel localization.

6.2 Related Works

Methods for *sub-pixel* point detection can be classified into three categories: upsampling-based, refinement-based, and regression-based approaches.

Upsampling-based Approaches

The standard paradigm of first transforming the point detection problem into a heatmap prediction problem (e.g., (Tompson et al., 2015; Merget et al., 2018)) before estimating point locations from the maxima of the discrete prediction heatmap (Li et al., 2020; Tompson et al., 2014) is not well-suited for sub-pixel applications. Indeed, the precision of these models is inherently limited to pixel accuracy. Several works achieve sub-pixel accuracy in this setting by simply inferring finer-grained discrete heatmaps through explicit *upsampling*. This artificial increase in resolution can be implemented in several ways ranging from a naïve upsampling of the input image (Nehme et al., 2018) to a sophisticated upsampling of the prediction map itself with a trained refinement network (Hu et al., 2019). While this process enables sub-pixel predictions with respect to the original image resolution, it suffers from two drawbacks: first, the estimates are still constrained to pixel locations in the upsampled space, and thus the precision of the predictions

Table 6.1: Characteristics of related works in multi-instance and sub-pixel point localization. No prior work allows for an end-to-end learning of point localization in dense multi-instance settings without the use of spatial upsampling. **SP**: sub-pixel capabilities, **MI**: multi-instance ability, **DS**: suitable for dense settings, **NP**: no post-processing required, and **NU**: no explicit upsampling needed.

		SP	**MI**	**DS**	**NP**	**NU**
Discrete Heatmap-matching			✓			✓
+ Refinement	(Graving et al., 2019; Zhang et al., 2020; Yang et al., 2017)	✓	✓	(✓)		✓
ChArUcoNet	(Hu et al., 2019)	✓	✓		✓	
Deep-Storm	(Nehme et al., 2018)	✓	✓	(✓)	✓	
Tiny People Pose	(Neumann & Vedaldi, 2018)	✓			✓	✓
Fractional Heatmap Reg.	(Tai et al., 2019)	✓			✓	✓
Global Regression	(Toshev & Szegedy, 2014; Carreira et al., 2016)	✓	(✓)		✓	✓
Offset Regression	(Fieraru et al., 2018; Zhou et al., 2019)	✓	✓	(✓)		✓
G-RMI	(Papandreou et al., 2017)	✓	✓			✓
Integral Pose Reg.	(Nibali et al., 2018; Sun et al., 2018; Luvizon et al., 2019)	✓			✓	✓
Ours		✓	✓	✓	✓	✓

is directly bounded by the amount of upsampling performed; secondly, the explicit upsampling of the visual representations significantly increases the memory requirement. In addition, as these approaches lack the ability to precisely detect multiple instances per pixel, they need to resort to large upsampling factors to deal with dense multi-instance applications such as single-molecule localization microscopy—exacerbating the issue of computational complexity.

Refinement-based Approaches

Instead of resorting to upsampling to obtain finer-grained discrete grids, other works propose first inferring heatmaps on coarser resolutions, before *refining* the estimates of the maxima locations to obtain predictions in $\mathbb{R}^2$ (Graving et al., 2019; Zhang et al., 2020; Yang et al., 2017; Donné et al., 2016). For instance, Graving et al. (2019) use Fourier-based convolutions to align a 2D continuous Gaussian filter with the discrete predicted heatmap, while Zhang et al. (2020) estimate the maxima (in $\mathbb{R}^2$) through log-likelihood optimization. However, while they can be deployed on top of any state-of-the-art discrete models, refinement-based methods introduce a clear disparity between the optimization objective (heatmap estimation) and the overall goal of the pipeline (sub-pixel localization). Consequently, as the refinement operation is not part of the optimization loop (i.e., the backpropagation does not take these operations into account), the learning of sub-pixel localization is not achieved in an end-to-end fashion which leads to suboptimal results.

Regression-based Approaches

In contrast to heatmap-matching, regression models can infer continuous locations without resorting to intermediate discretized representations. The most trivial approach consists in directly regressing the coordinates of the points of interest (Toshev & Szegedy, 2014; Carreira et al., 2016). However, this simple method suffers from several drawbacks (e.g., no translational invariance to the detriment of generalization capabilities and the number of points to detect has to be rigidly set in the model architecture). In contrast, offset regression models (Liu et al., 2016; Redmon et al., 2016) first subdivide the input space into a grid of smaller sub-regions, before inferring relative object coordinates and class probabilities within each region via regression. While originally proposed for object detection, this approach has also seen applications in point detection (Fieraru et al., 2018; Zhou et al., 2019; Vahdat, 2017), with the specificity that classification probabilities are commonly assigned through heatmap-matching. However, despite their ability to infer predictions in the continuous space and to leverage local features more efficiently than their global counterparts, these models often rely on loss functions that are highly discontinuous at the edges of the grid cells ((Vahdat, 2017) is a noticeable exception). Thus, in order to alleviate the discontinuity issues, large grid cells often have to be considered which is reminiscent of global coordinates regression models and their inherent drawbacks. More importantly, these methods often have to rely heavily on NMS to obtain sparse predictions, thus breaking the end-to-end learning of point localization. Both of these features are detrimental to the overall precision of the point estimates, and by extension, to the sub-pixel localization capabilities of these models, especially in multi-instance settings.

In this chapter, we make use of *both* the continuous prediction ability of offset regression and the finer-grained spatial learning capabilities of heatmap-matching-based learning to achieve precise multi-instance sub-pixel point localization.

6.3 Our End-to-end Sub-pixel Point Detection Model

We propose to tackle multi-instance sub-resolution point localization through dense offset prediction, continuous heatmap-matching-based learning, and instance counting regularization. An overview of the model is given in Figure 6.1.

6.3.1 Dense offset prediction

As in standard offset regression (Liu et al., 2016; Redmon et al., 2016), we propose to train a model to infer, for each pixel of the final representation, n tuples $(\hat{\Delta}^x, \hat{\Delta}^y, \hat{\mathbf{p}})$ with coordinate offsets $\hat{\Delta}^x, \hat{\Delta}^y \in [-\frac{1}{2}, \frac{1}{2}]$ and class probabilities $\mathbf{p} \in [0,1]^d$, where d is the number of classes. In contrast to standard approaches, the loss introduced in this chapter (see Equation 6.3) does not present any discontinuity at the sub-regions borders, and thus does not explicitly require the resolution of the input image to be downsampled before the loss computation. As a result, a one-to-one correspondence between the pixels in the final representation and the pixels in the input image can be exploited, which makes it possible to infer a set of n tuples $(\hat{\Delta}^x, \hat{\Delta}^y, \hat{\mathbf{p}})$ for each pixel in the input image—even smaller granularity can be considered. More specifically, the model $\hat{f}_\theta$ maps any given input image $\mathbf{X}$ of size $(w \times h)$ to a *dense* ensemble of $N := n \cdot w \cdot h$ points $(\hat{x}, \hat{y}, \hat{\mathbf{p}})$, where the point coordinates $\hat{x}$ and $\hat{y}$ are equal to the sum of the continuous offsets predictions $\hat{\Delta}^x, \hat{\Delta}^y$ and the respective pixel center locations $(\bar{x}, \bar{y})$, namely

$$\begin{aligned}\hat{f}_\theta(\mathbf{X}) &= \left\{ (\hat{x}, \hat{y}, \hat{\mathbf{p}})_{(i)} \,|\, i \leq N \right\} \\ &= \left\{ \left(\bar{x}_{(j,k)} + \hat{\Delta}^x_{(j,k,l)},\ \bar{y}_{(j,k)} + \hat{\Delta}^y_{(j,k,l)},\ \hat{\mathbf{p}}_{(j,k,l)} \right) \,|\, j \leq w, k \leq h, l \leq n \right\} =: \mathcal{P}_\theta.\end{aligned} \tag{6.1}$$

Overall, this mapping offers a full and fine-grained coverage of the original image space, and thus makes the precise prediction of multiple point locations in $\mathbb{R}^2$ possible, thereby unlocking multi-instance sub-pixel capabilities. Indeed, the object locations $(\hat{x}, \hat{y})$ can lie anywhere in $\mathbb{R}^2$, in contrast to standard point detection models (Tompson et al., 2015; Newell et al., 2016; Yang et al., 2017; Bulat & Tzimiropoulos, 2016; Pfister et al., 2015) where point locations are limited to the discrete grid defined by the input pixels. Similarly, the true point labels are not discretized, i.e., $\mathcal{L} := \{(x,y)_j \in \mathbb{R}^2 \mid j \leq M\}$, with M the number of labels in an image. Since such dense oversampling of point predictions is not suitable for classical offset regression loss functions (Redmon & Farhadi, 2018), a novel flexible loss function has to be introduced.

Remark

The points $(\hat{x}, \hat{y}, \hat{\mathbf{p}}) \in \mathcal{P}_\theta$ outputted by the model correspond to the final point localization estimates (see Section 6.3.3 for details on how the model converges almost all instance probabilities to zero, thus turning the dense set of predictions into a sparse one) and not to intermediate representations that span a density—or a heatmap—as in (Tompson et al., 2015; Neumann & Vedaldi, 2018; Yang et al., 2017; Bulat & Tzimiropoulos, 2016; Pfister et al., 2015) or that require extensive post-processing as in (Papandreou et al., 2017).

6.3.2 Continuous Heatmap-Matching

In order to estimate the model parameters θ through backpropagation, the model predictions $\mathcal{P}_\theta$ and the ground-truth labels $\mathcal{L}$ have to be compared using a sensible and differentiable measure. To that end, we propose a novel *continuous* generalization of the standard discrete heatmap-matching paradigm (Tompson et al., 2015; Yang et al., 2017; Bulat & Tzimiropoulos, 2016) that effectively solves the problems inherent to classical offset regression loss functions while retaining their continuous localization learning ability. First, the point predictions $\mathcal{P}_\theta$ and point labels $\mathcal{L}$ are mapped to continuous heatmaps using a Gaussian kernel K with smoothing parameter λ (similar to Gaussian mixture). Thus, the value of the continuous prediction heatmap (induced by $\mathcal{P}_\theta$) at any given point $(x_0, y_0) \in \mathbb{R}^2$ is equal—up to a normalization factor—to

$$\hat{\mathcal{H}}(x_0, y_0 \,|\, \mathcal{P}_\theta) = \sum_{i}^{N} \hat{p}_i K(\hat{x}_i, \hat{y}_i, x_o, y_o) = \sum_{i} \hat{p}_i \exp\left(-\frac{(\hat{x}_i - x_0)^2}{\lambda^2} - \frac{(\hat{y}_i - y_0)^2}{\lambda^2}\right), \tag{6.2}$$

where, to simplify notation, only a single object class is considered (i.e., $d=1$) as the generalization for $d>1$ is trivial.

Classical models explicitly compute and compare (e.g., through an ℓ_2-loss) the *discrete* label heatmap obtained through the smoothing of the point labels and the *discrete* prediction heatmap inferred by the model. As a result, the heatmap comparison becomes gradually more approximate as lower-resolution inputs are considered, which inevitably has a detrimental effect on the sub-pixel learning capability. In contrast, we propose to directly compute *analytically* the difference between the *continuous* label and prediction heatmaps induced by the point labels and predictions. More precisely, we propose the integrated local squared distance between the two planes as loss function for the learning of point localization:

$$\begin{aligned}\mathscr{L}_{\mathrm{HM}}(\mathcal{P}_\theta, \mathcal{L}) &= \iint_{\mathbb{R}^2} \left[\mathcal{H}(x_0, y_0 \,|\, \mathcal{L}) - \hat{\mathcal{H}}(x_0, y_0 \,|\, \mathcal{P}_\theta)\right]^2 dx_0 dy_0 \\ &= \iint_{\mathbb{R}^2} \Bigg[\sum_j \exp\left(-\frac{(x_j - x_0)^2}{\lambda^2} - \frac{(y_j - y_0)^2}{\lambda^2}\right) \\ &\qquad - \sum_i \hat{p}_i \exp\left(-\frac{(\hat{x}_i - x_0)^2}{\lambda^2} - \frac{(\hat{y}_i - y_0)^2}{\lambda^2}\right)\Bigg]^2 dx_0 dy_0.\end{aligned} \tag{6.3}$$

Performing integration over the entire $\mathbb{R}^2$ space, rather than over the image domain only, helps avoid special treatment of points at image boundaries.

Overall, since the heatmaps are never explicitly computed, their comparison is performed with infinite spatial resolution, thus alleviating the issues arising from space discretization. Moreover, as the computation of the heatmap comparison is exact regardless of the resolution of the input image, the smoothing bandwidth λ can be selected as tight as needed without any loss of information. This allows for a more precise learning of localization, and thus increased sub-pixel detection capabilities.

Closed-form Loss Computation

A closed-form solution of the loss function (Equation 6.3) can be derived (see Appendix C.1) by successively using the distributivity property, Fubini's theorem, and the limits of the Gaussian error function:

$$\begin{aligned}\mathscr{L}_{\mathrm{HM}}(\mathcal{P}, \mathcal{L}) = &\sum_i \sum_j \frac{\pi\lambda^2}{2} \exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\lambda^2}\right) \\ &+ \sum_i \sum_j \hat{p}_i \hat{p}_j \frac{\pi\lambda^2}{2} \exp\left(-\frac{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2}{2\lambda^2}\right) \\ &- 2 \sum_i \sum_j \hat{p}_i \frac{\pi\lambda^2}{2} \exp\left(-\frac{(\hat{x}_i - x_j)^2 + (\hat{y}_i - y_j)^2}{2\lambda^2}\right).\end{aligned} \tag{6.4}$$

This equation enables the straightforward computation of the partial derivatives of the loss function with respect to the class probability predictions and the location estimates used for backpropagation, see Appendix C.1 for formulas and derivations.

Remark

While the use of dense offset regression in conjunction with Gaussian mixtures is reminiscent of (Papandreou et al., 2017; Neumann & Vedaldi, 2018), our model significantly differs in the nature of the predictions it infers. Indeed, previous works have as underlying objective the explicit estimation of prediction heatmaps. For instance, the dense point predictions in (Neumann & Vedaldi, 2018) are used to estimate a continuous density, which in turn is used to infer the final point locations. Thus, similarly to the classical heatmap-matching approaches, the density—or heatmap—is the target of the learning and not the localization itself. In contrast, the points outputted by our model directly correspond to the final point predictions; the heatmaps are not a goal in themselves but are rather used as building blocks of our loss function to assess the quality of the predictions. Consequently, in our framework, the final point predictions are an integral part of the optimization loop which allows for the end-to-end learning of multi-instance sub-pixel point localization.

6.3.3 Detection Sparsity through Counting Regularization

Detection sparsity (i.e., obtaining one clear-cut non-ambiguous point estimate per label) is a critical issue in dense multi-instance sub-pixel localization applications. Indeed, relying on post-processing operations such as NMS to map a set of ambiguous estimates to clear-cut predictions is not suitable in this setting: for instance, in dense setups, two predictions made within the same pixel may correspond to two distinct ground-truth point locations, and thus should not necessarily be merged into a single prediction. Additionally, systematically combining several low-probability predictions into a single high-probability point estimate is far from optimal as it inevitably has a negative impact on the spatial precision of the predictions and, by extension, the model sub-pixel capability.

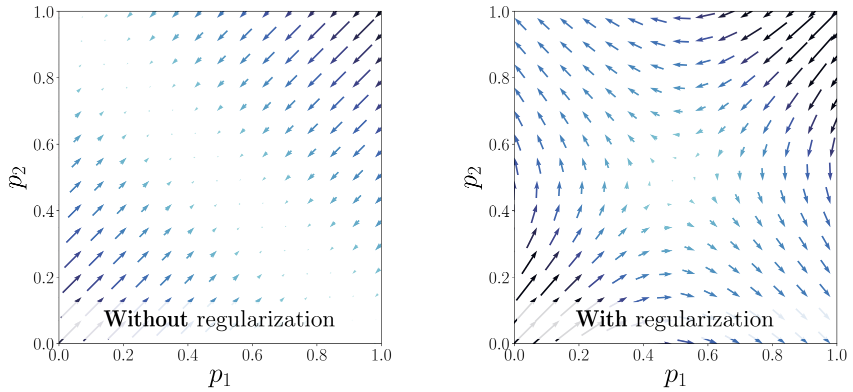


Figure 6.2: Prediction sparsity through counting regularization. Gradients of the loss function with respect to instance probabilities p_1, p_2 for the situations described in the example of Section 6.3.3.

Counting Regularization

The continuous heatmap-matching loss function $\mathscr{L}_{\text{HM}}$ does not guarantee detection sparsity on its own; indeed, splitting a point prediction $(\hat{x}, \hat{y}, \hat{p})$ into two point predictions with half probability each $(\hat{x}, \hat{y}, \hat{p}/2)$ has no effect on the loss. To remedy this issue without resorting to ineffective post-processing operations, we propose—as done in Section 5.5.2—adding the sparsity-inducing Poisson-binomial loss function (Equation 2.9) as a regularizer to the training objective; in this way, clear-cut and precise predictions can be learned and inferred in an end-to-end fashion:

$$\begin{aligned} \mathscr{L}_{\text{Count}}(\theta) &= D_{KL}(\mathbb{1}_c \| \textstyle\sum_i \mathcal{B}(\hat{p}_i)) \\ &= -\log\left(\Pr\left(\hat{\mathscr{C}_\theta} = c \mid \mathbf{X}\right)\right) \\ &= -\log\Bigg(\sum_{A \in F} \prod_{i \in A} \hat{p}_i \prod_{j \in A^c} (1 - \hat{p}_j)\Bigg), \end{aligned} \tag{6.5}$$

where, once again F is the set of all subsets of $\{1, \ldots, |\hat{\mathbf{p}}|\}$ of size $c = |\mathcal{L}|$. Thus, while the heatmap-matching loss $\mathscr{L}_{\text{HM}}$ does not ensure prediction sparsity (e.g., it does not penalize the splitting of predictions into several lower-likelihood ones), this regularizer does (see Chapter 3).

6.3.4 Benefits of Counting Regularization

In this section, we present a few examples that illustrate more precisely the usefulness of the proposed counting regularization on the learning process.

Prediction Sparsity

As an illustration of the sparsity-inducing effect of the regularization, let us consider a unique object at location (x, y) and two point predictions, at the same coordinates: $(x, y, \hat{p}_1)$ and $(x, y, \hat{p}_2)$. This setup describes a situation where the predictions are perfectly aligned spatially with the ground-truth, but where the instance probability still needs to be fine-tuned.

Without Regularization In this scenario, the heatmap-matching loss function is proportional to

$$\begin{aligned}\mathscr{L}_{\mathrm{HM}}(\mathcal{P}_\theta, \mathcal{L}) &\propto 1 + 2\hat{p}_1\hat{p}_2 + \hat{p}_1^2 + \hat{p}_2^2 - 2\hat{p}_1 - 2\hat{p}_2 \\ &= (\hat{p}_1 + \hat{p}_2 - 1)^2.\end{aligned} \tag{6.6}$$

and is thus minimized when if and only if

$$\hat{p}_1 + \hat{p}_2 = 1. \tag{6.7}$$

This result is trivial when considering that the loss is defined as the integrated squared difference between sums of Gaussians. However, it confirms that all combinations of $\hat{p}_1$ and $\hat{p}_2$ that satisfy this condition are stable solutions to the *unregularized* optimization problem.

With Regularization In contrast, when incorporating the counting regularization, the loss function is proportional to

$$\mathscr{L} \propto (\hat{p}_1 + \hat{p}_2 - 1)^2 - \beta \underbrace{\log\left(\hat{p}_1(1 - \hat{p}_2) + \hat{p}_2(1 - \hat{p}_1)\right)}_{\mathscr{L}_{\mathrm{MC}}}. \tag{6.8}$$

Figure 6.2 shows the value of the gradients of the loss function (with and without regularization) with respect to probability estimates $\hat{p}_1$ and $\hat{p}_2$. These results are strongly reminiscent of the ones presented for the simple example derived in Section 3.1.2. Indeed, Figure 6.2 confirms that, without regularization, the optimization problem can have as a stable solution any combination of $\hat{p}_1$ and $\hat{p}_2$ which satisfies $\hat{p}_1 + \hat{p}_2 = 1$. In contrast, these results demonstrate, once again, that the Poisson-binomial counting encourages the convergence of the predictions towards sparsity; indeed, only $(\hat{p}_1, \hat{p}_2) = (1, 0)$ or $(\hat{p}_1, \hat{p}_2) = (0, 1)$ are stable solutions to the *regularized* optimization problem.

Faster Location Convergence

Let us now consider a unique object in *1-dimension* at location (x) and two point predictions $(\hat{x}_1 = x - \Delta, \hat{p})$ and $(\hat{x}_2 = x + \Delta, \hat{p})$. This setup describes a situation where two predictions are equidistant from the ground-truth.

Without Regularization In this scenario, the optimal probability estimate $\hat{p}$—when considering only the heatmap-matching loss function as a training objective—is

$$\begin{aligned}
&\frac{\partial}{\partial \hat{p}} \mathscr{L}_{\text{HM}}(\mathcal{P}_\theta, \mathcal{L}) = \pi\lambda^2 \hat{p}_{opt} \exp\left(-\frac{4\Delta^2}{2\lambda^2}\right) + \pi\lambda^2 \hat{p}_{opt} - \pi\lambda^2 \exp\left(-\frac{\Delta^2}{2\lambda^2}\right) = 0 \\
&\Leftrightarrow \hat{p}_{opt}\left(\exp\left(-\frac{4\Delta^2}{2\lambda^2}\right) + 1\right) = \exp\left(-\frac{\Delta^2}{2\lambda^2}\right) \\
&\Leftrightarrow \hat{p}_{opt} = \exp\left(-\frac{\Delta^2}{2\lambda^2}\right) \Big/ \left(\exp\left(-\frac{4\Delta^2}{2\lambda^2}\right) + 1\right).
\end{aligned} \tag{6.9}$$

With Regularization Similarly, the optimal probability estimate $\hat{p}$—when integrating the counting regularization to the training objective—is given by the following equation:

$$\begin{aligned}
&\frac{\partial}{\partial \hat{p}} \{\mathscr{L}_{\text{HM}}(\mathcal{P}_\theta, \mathcal{L}) + \beta \mathscr{L}_{\text{MC}}(\mathcal{P}_\theta, \mathcal{L})\} \\
&\quad = \pi\lambda^2 \hat{p}_{opt} \exp\left(-\frac{4\Delta^2}{2\lambda^2}\right) + \pi\lambda^2 \hat{p}_{opt} - \pi\lambda^2 \exp\left(-\frac{\Delta^2}{2\lambda^2}\right) - \beta \underbrace{\frac{1 - 2\hat{p}_{opt}}{(\hat{p}_{opt} - 1)\hat{p}_{opt}}}_{\frac{\partial}{\partial \hat{p}} \mathscr{L}_{\text{MC}}} = 0 \\
&\quad \Leftrightarrow \hat{p}_{opt} \exp\left(-\frac{4\Delta^2}{2\lambda^2}\right) + \hat{p}_{opt} - \exp\left(-\frac{\Delta^2}{2\lambda^2}\right) - \tilde{\beta} \frac{1 - 2\hat{p}_{opt}}{(\hat{p}_{opt} - 1)\hat{p}_{opt}} = 0 \\
&\overset{\hat{p}_{opt} \neq 0,1}{\Leftrightarrow} \hat{p}_{opt}(\hat{p}_{opt} - 1)\left[\hat{p}_{opt} \exp\left(-\frac{4\Delta^2}{2\lambda^2}\right) + \hat{p}_{opt} - \exp\left(-\frac{\Delta^2}{2\lambda^2}\right)\right] - \tilde{\beta}(1 - 2\hat{p}_{opt}) = 0,
\end{aligned} \tag{6.10}$$

which can easily be solved using a standard root-finding algorithm (e.g., Brent's method (Brent, 1973)).

In addition, the location gradient—the partial derivative of the loss with respect to the location x, assuming $\hat{p} = \hat{p}_{opt}$, is

$$\frac{\partial}{\partial \hat{x}_2} \mathscr{L}_{\mathrm{HM}}(\mathcal{P}_\theta, \mathcal{L}) = \hat{p}_{opt} \pi \exp\left(-\frac{\Delta^2}{2\lambda^2}\right)(\Delta) - \hat{p}_{opt}^2 \pi \exp\left(-\frac{4\Delta^2}{2\lambda^2}\right)(2\Delta)\,. \tag{6.11}$$

The location gradient for the regularized optimization problem is identical since the value of the counting regularization is independent of the instance locations. Thus, the location gradient, assuming that $\hat{p}$ is set optimally, can be obtained for both training situations (with and without regularization) by plugging Equation 6.9 or the root derived from Equation 6.10 into Equation 6.11.

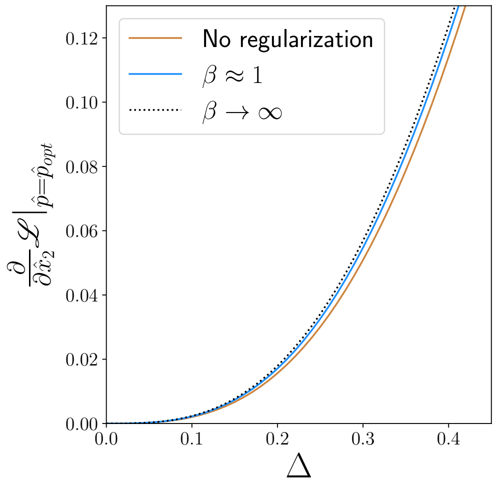
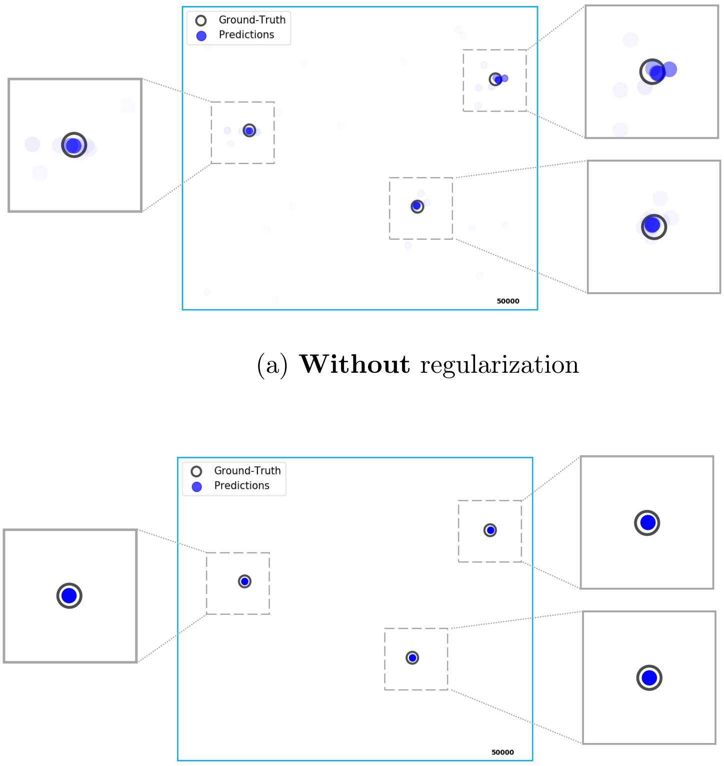


Figure 6.3: Location gradient

Figure 6.3 displays the result as a function of Δ for $\lambda = 1$. Above all, this figure shows that the location gradient becomes slightly larger with increased regularization. This effect suggests that, in this scenario, the regularization acts as a means for faster convergence. Indeed, the counting regularization encourages the learning process to converge the location prediction faster towards the ground-truth object location. While this effect is hard to quantify in higher dimensions (i.e., more than two point predictions), the counting regularization is still expected to improve location convergence in more complex settings.



(a) **Without** regularization

(b) **With** Regularization

Figure 6.4: Impact of counting-based regularization on the convergence of the continuous heatmap-matching loss function. Final point locations and probabilities (color intensity) after 50K gradient descent iterations for the one-pixel example. *(Best seen in video)*

Improved Convergence

In order to assess the impact of the counting regularization on convergence in higher dimensions, we propose a simple simulation-based experiment. More precisely, in a single pixel, we place and sample uniformly at random three ground-truth point locations as well as fifty initial point predictions, where the probability assigned to each prediction is also sampled uniformly at random. The location and the probability of the initial predictions are then iteratively updated through gradient descent using, as cost function, either the heatmap-matching loss function alone or the regularized version.

The resulting convergence videos can be downloaded or directly viewed on the project page[1] (see Figure 6.4 for an illustration of the convergence after 50K iterations). Overall, these videos clearly demonstrate the effect of the counting regularization on the convergence of the point predictions. Indeed, with regularization, not only do the predictions converge faster *spatially* towards the ground-truth locations, but most importantly their probabilities converge significantly faster towards *sparsity*. In fact, in this example, the predictions resulting from a regularized optimization fully converge in less than 30K iterations, while their unregularized counterparts are still scattered around the ground-truth location even after 100K iterations. As the regularized optimization produces only three point predictions with non-zero probability, the inference can be done by simply outputting the location corresponding to these instances, without the need for any post-processing. Overall, the faster convergence of both the location and the probability estimates are expected to have a positive effect on the learning of sub-pixel point localization in more complex settings.

6.4 Experiments

Code for all experiments is publicly available[1].

6.4.1 Single-molecule Localization Microscopy

In this section, we replicate the experiment on molecule localization microscopy proposed by Nehme et al. (2018). The task consists in determining the localization of multiple blinking molecules on diffraction-limited images of fluorescent simulated microtubules. The overall setting is particularly challenging as multiple instances can fall within the same pixel of the input image, thus requiring precise multi-instance sub-pixel localization capabilities.

Model and Benchmarks

The model in (Nehme et al., 2018) achieves sub-pixel localization by explicitly increasing the resolution of each dimension of the input image by a factor 8 (i.e., effectively increasing the number of pixels by $8 \times 8 = 64$) before inferring a single

[1] https://github.com/SchroeterJulien/ACCV-2020-Subpixel-Point-Localization

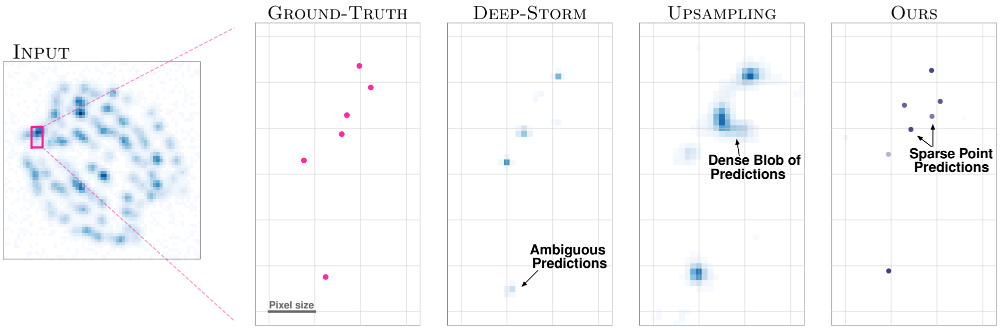


Figure 6.5: Model predictions for multi-instance sub-pixel molecule localization. No non-maximum suppression was performed on our predictions, our model learns to *directly* infer sparse point predictions as a result of the counting regularization. The color intensity of the predictions is proportional to the probability estimate.

localization probability for each pixel of the upsampled input (referred to as Deep-Storm). By keeping the architecture as suggested in (Nehme et al., 2018) and replacing the loss with a classical discrete heatmap-matching approach, we obtain a benchmark reminiscent of upsampling-based heatmap-matching (referred to as Upsampling). As the input image is subject to high levels of upsampling, the model architecture relies on a series of downsampling layers followed by a series of upsampling layers to obtain a wide enough receptive field. In contrast, since our approach decouples the resolution of the input image from the resolution of the predictions and thereby obviates the need for upsampling, these layers are not needed to learn meaningful representations; our method can directly operate on the original images instead and infer $n=2$ points (i.e., n tuples of offsets and probabilities) for each pixel. (In our model, a point with probability greater than 0.3 is considered a detection.)

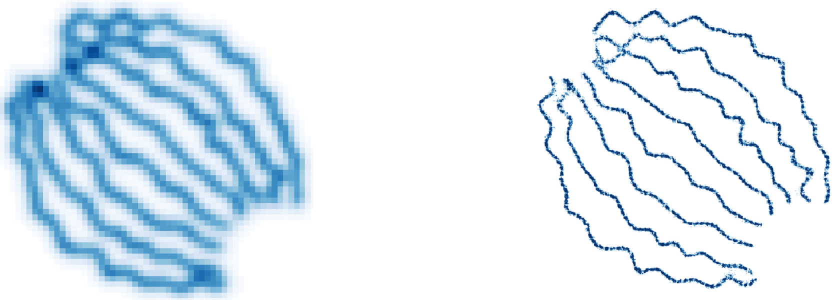
Evaluation and Results

All models are trained with the data provided by Nehme et al. (2018) and tested on the fluorescent simulated microtubules from Sage et al. (2015). The Jaccard index—a standard metric of set similarity—is computed with the tool provided by Sage et al. (2015) using various tolerances τ. Table 6.2 reveals that our approach not only displays the best overall performance on this experiment, but also achieves fast inference as it can perform precise multi-instance sub-pixel

Table 6.2: Single-molecule localization microscopy results. Comparison of various methods on the sub-pixel single-molecule localization experiment (Nehme et al., 2018). The Jaccard index [and F_1 score] is computed with the software provided by Sage et al. (2015).

	Jaccard Index [F_1]		Inference Speed
Method	$\tau = 25$nm	$\tau = 50$nm	time/image
Deep-Storm	0.153 [0.266]	0.416 [0.588]	17.44 ms
Upsampling	0.171 [0.292]	0.448 [0.618]	17.44 ms
Refinement	0.195 [0.326]	0.448 [0.619]	0.76 ms
Ours	**0.221 [0.361]**	**0.482 [0.650]**	0.76 ms

localization using the original input resolution without the need for any explicit upsampling. This outcome can partially be attributed to our approach's ability to infer sparse clear-cut point estimates without requiring any additional post-processing, see Figure 6.5. The overall rendering of a test image of microtubules provided by Sage et al. (Sage et al., 2015) is presented in Figure 6.6, see (Nehme et al., 2018) for details.



(a) Stacked input of diffraction-limited images.

(b) Stacked molecule locations inferred by our model.

Figure 6.6: Rendering of test microtubules using our model.

Ablation Study

We replicate the same experiment with various forms of regularization to assess the impact of count supervision on the performance of our model. Table 6.3 shows

Table 6.3: Regularization ablation study on the single molecule localization microscopy experiment.

	JACCARD	
REGULARIZATION	$\tau = 25$nm	50nm
NONE	0.211	0.456
l_1 (as in (NEHME ET AL., 2018))	0.208	0.454
COUNTING ($\mathscr{L}_{\text{Count}}$)	**0.221**	**0.482**

that the theoretical benefits of count-based regularization directly translate to improved sub-pixel molecule localization capabilities in practice.

Training with the counting-based regularization leads to faster convergence (see Section 6.3.4). Thus, in order to assess whether the performance gap between the different regularization methods is due to the selected number of training steps (i.e., 50k)—and thus can simply be explained by a slower convergence—we evaluate the performance of the model without regularization when training for a longer time.

Overall, training longer does not increase the sub-pixel detection capabilities of the non-regularized model. Indeed the performance is 0.203/0.439, 0.207/0.439, and 0.202/0.435 ($\tau = 25$nm/$\tau = 50$nm) after training for 100k, 250k, and 500k iterations respectively. While this observation is the result of a single run, the number of training iterations does certainly not explain the performance gap on its own. This experiment therefore suggests that counting-based regularization presents more benefits than just convergence speed.

Sensitivity to Smoothing Parameter λ

In order to assess the sensitivity of the results to changes in the model softness parameter λ, we replicate the single-molecule localization microscopy experiments with a wide array of potential λ values. The results—reported in Table 6.4—highlight the remarkable robustness of the model to changes in this key hyperparameter. Indeed, the model achieves highly consistent results with near-constant performance metrics for λ between 0.2 and 0.8. Thus, the model's success is almost independent of the value of this hyperparameter.

Table 6.4: Single-molecule localization microscopy λ-sensitivity analysis based on the experiment proposed in (Nehme et al., 2018). The Jaccard index [and F_1 score] is computed with the software from (Sage et al., 2015). The setting $\lambda = 0.2$ is the one reported—without any hyperparameter optimization—as Ours in this section (see ↓).

Tol.	Jaccard Index	↓							
	λ=0.1	0.2	0.3	0.4	0.5	0.6	0.8	1.2	2.0
$\tau = 25$nm	0.219	**0.234**	0.233	0.229	0.226	**0.234**	**0.234**	0.219	0.208
$\tau = 50$nm	0.464	0.517	0.519	0.519	0.524	**0.531**	0.526	0.453	0.410

This range of suitable smoothing parameters is intuitive. On the one hand, this key hyperparameter has to be set high enough in order to allow for an effective gradient-based optimization of the loss. Indeed, in the most extreme case where $\lambda \to 0$, the loss function is non-differentiable with zero gradients almost everywhere. While any value greater than zero yields theoretically a differentiable loss, selecting a λ that is too small leads to extreme gradients near the point labels and near-zero gradients everywhere else—which is suboptimal for training. In this context, a smoothing parameter smaller than 0.2 times the size of pixel might lead to such optimization issues. On the other hand, λ has to be set small enough in order for the loss function to convey enough information about the exact location of the labels. Indeed, in the most extreme case where $\lambda \to \infty$, all the information about the location of the labels gets lost. Thus, in a context that requires extremely precise sub-pixel location estimates (i.e., error tolerance set to 0.25 times the size of a pixel), a smoothing parameter λ smaller than 1 ensures a fine-grained enough location supervision.

6.4.2 Checkerboard Corner Detection

The precise detection of corners in checkerboards is a key component of camera calibration. This challenging task requires the predictions to lie within a fraction of a pixel of the ground-truth in order to be of practical use. In this section, we compare the sub-pixel localization capabilities of our method and other learning-based approaches with state-of-the-art classical local feature-based methods that

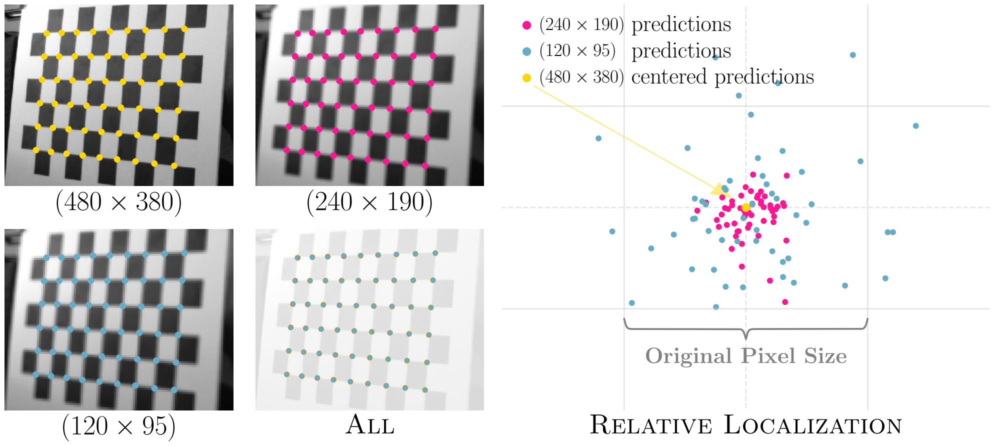


Figure 6.7: Corner detection across different resolutions. The low-resolution location estimates stay well within half a pixel of the original predictions, which corresponds to 1/8 of a pixel in the lowest resolution.

are specifically tailored to the sub-pixel detection of such corners (Placht et al., 2014; Duda & Frese, 2018; Sinzinger, 2008).

Training Data

To train the various learning-based models, we generate a *synthetic* dataset composed of 20k checkerboard images. This not only allows us to automatically simulate numerous transformations (lens distortions, lighting variations, perspective transformations, noise) in a controllable manner, but most importantly gives us an exact account of the ground-truth corner locations, as opposed to human-annotated datasets that are inherently prone to inaccuracies. More details about the dataset generation process are included in Appendix C.2.2.

Model Architecture

In line with previous checkerboard corner detection methods (Donné et al., 2016; Chen et al., 2018), a "shallow" architecture comprised of only three convolutional layers—with 32, 32 and 64 filters respectively—is considered for all learning-based models, including ours. For faster training, two downsampling convolutional layers, with stride 2, are added to our model, after both the first and second convolutional layers. This modification merely enables our model to assign probabilities and

Table 6.5: Full results of localization performance in low-resolution settings on the **GoPro** dataset (Placht et al., 2014). **Consistency**: mean-absolute displacement (and 90th quantile) between predictions on high and low-resolution images downsampled by a factor δ. **Reprojection Error:** corresponding errors in corner reprojection [and number of fully detected boards]. In units of original pixel size.

		Consistency		
	Methods	$\delta = 2$	4	6
Classic	OCamCalib (Scaramuzza et al., 2006)	0.660 (1.12)	1.389 (2.50)	1.989 (3.61)
	Rochade (Placht et al., 2014)	0.380 (0.67)	0.467 (0.81)	1.125 (2.07)
	OpenCV (Bradski, 2000)	**0.111 (0.20)**	**0.179 (0.31)**	0.336 (0.50)
	MATLAB (Geiger et al., 2012)	0.129 (0.20)	0.198 (0.32)	**0.314 (0.50)**
Learn.	DL-Heatmap (sim. (Donné et al., 2016))	0.900 (1.41)	1.629 (2.24)	2.395 (3.61)
	+ Refinement (sim. (Graving et al., 2019))	0.153 (0.28)	0.279 (0.50)	0.428 (0.76)
	OURS	**0.133 (0.24)**	**0.244 (0.43)**	**0.378 (0.66)**

		Reprojection Error		
	Methods	$\delta = 2$	4	6
Classic	OCamCalib (Scaramuzza et al., 2006)	0.107 [100]	0.390 [73]	——— [18]
	Rochade (Placht et al., 2014)	0.085 [100]	0.321 [100]	1.716 [71]
	OpenCV (Bradski, 2000)	**0.045** [100]	0.256 [98]	0.994 [73]
	MATLAB (Geiger et al., 2012)	**0.045** [99]	**0.205** [100]	**0.325** [100]
Learn.	DL-Heatmap (sim. (Donné et al., 2016))	0.146 [100]	0.363 [100]	0.797 [77]
	+ Refinement (sim. (Graving et al., 2019))	0.054 [100]	0.336 [100]	0.531 [100]
	OURS	**0.046** [100]	**0.198** [82]	**0.417** [100]

offsets to bigger regions of 4×4 pixels rather than to each pixel of the original input. In contrast, no downsampling could be performed on all other learning-based benchmarks, as it would only deteriorate the precision of their predictions.

Baselines

The following classical baselines are considered: OCamCalib (Scaramuzza et al., 2006), ROCHADE (Placht et al., 2014), OpenCV (Bradski, 2000), and MATLAB (Geiger et al., 2012). We also include three learning-based benchmarks which use the model architecture described above and are trained on our synthetic dataset: standard discrete heatmap-matching with naïve argmax maximum picking (similar to (Donné et al., 2016; Chen et al., 2018)), heatmap-matching with local refinement through Gaussian distribution fitting (comparable to standard refinement-based approaches (Graving et al., 2019)), and higher resolution heatmap-matching where the input images are explicitly upsampled with a factor 8 (similar to (Nehme et al., 2018)).

Table 6.6: Full results of localization performance in low-resolution settings on the **uEye** dataset (Placht et al., 2014). **Consistency**: mean-absolute displacement (and 90th quantile) between predictions on high and low-resolution images downsampled by a factor δ. **Reprojection Error:** corresponding errors in corner reprojection [and number of fully detected boards]. In units of original pixel size.

		Consistency	
	Methods	$\delta = 2$	4
Classic	OCamCalib (Scaramuzza et al., 2006)	0.783 (1.58)	1.447 (2.92)
	Rochade (Placht et al., 2014)	0.176 (0.29)	0.587 (1.05)
	OpenCV (Bradski, 2000)	0.126 (0.21)	0.889 (2.66)
	MATLAB (Geiger et al., 2012)	**0.090 (0.15)**	**0.174 (0.29)**
Learn.	DL-Heatmap (sim. (Donné et al., 2016))	0.955 (1.41)	1.666 (2.24)
	+ Refinement (sim. (Graving et al., 2019))	**0.077 (0.14)**	0.562 (1.20)
	OURS	0.134 (0.23)	**0.348 (0.64)**

		Reprojection Error	
	Methods	$\delta = 2$	4
Classic	OCamCalib (Scaramuzza et al., 2006)	0.129 [200]	0.197 [114]
	Rochade (Placht et al., 2014)	0.057 [206]	0.107 [197]
	OpenCV (Bradski, 2000)	0.057 [197]	——— [0]
	MATLAB (Geiger et al., 2012)	**0.048** [206]	**0.059** [204]
Learn.	DL-Heatmap (sim. (Donné et al., 2016))	0.126 [206]	0.230 [175]
	+ Refinement (sim. (Graving et al., 2019))	**0.052** [206]	0.086 [162]
	OURS	0.055 [200]	**0.073** [187]

Evaluation and Results

We evaluate the methods on the standard uEye and GoPro datasets (Placht et al., 2014). Since these real-world test datasets do not contain any ground-truth corner positions, we assess the sub-pixel localization capabilities of the different approaches both through prediction consistency across resolutions and through corner reprojection errors. Note that, in these experiments, the upsampling approach yields representations that are far too large to be supported by standard GPUs, especially on the GoPro dataset, which illustrates its limits.

First, we measure prediction consistency by comparing the corner localizations obtained on the original high-resolution images with those obtained on the lower-resolution inputs downsampled by a factor δ. This experiment thus posits that a direct correlation exists between a model's ability to infer consistent sub-pixel locations and its capacity to output consistent predictions across various resolutions. The mean absolute displacement and the 90th quantile reported in

Tables 6.5 and 6.6 show that our approach yields very consistent corner location estimates (see also Figure 6.7). Overall, this performance demonstrates that our model is capable of inferring point locations well beyond pixel accuracy.

Second, we compute the reprojection errors—a standard metric in camera calibration—of the predicted checkerboard corners in low-resolution settings (i.e., input downsampled by a factor δ) after performing camera calibration with the standard OpenCV implementation (Bradski, 2000). Overall, the excellent performance of our approach on this task (see Tables 6.5 and 6.6), much higher than most classical state-of-the-art approaches, reveals once again the high sub-pixel capabilities of our model. (Additional results are included in Appendix C.2.)

In conclusion, our method displays strong overall results for all downsampling levels and outperforms the other deep learning benchmarks on almost all measures. More precisely, our approach is only outperformed by the OpenCV (Bradski, 2000)—on specific metrics—and MATLAB (Geiger et al., 2012) implementation. However, while OpenCV performs slightly better on the consistency measure, its relatively high number of false positives and false negatives have a clear impact on its camera calibration performance (i.e., see reprojection error below). Overall, our approach appears to be extremely suitable for high precision detection and calibration in low-resolution settings. These results are all the more remarkable when considering that the learning-based models are trained solely on synthetic images and that the classical benchmarks are specifically designed for this task *only*—they are not portable to other applications in contrast to our approach.

Regularization Ablation Study

As done for the single-molecule localization microscopy experiment, we perform an ablation study to measure the impact of the counting-based regularization on the performance of the checkerboard corner detection model. The results—summarized in Tables 6.7 and 6.8—are in line with the findings of the single-molecule localization microscopy experiment. Indeed, adding the counting loss as a regularizer to the soft localization learning loss consistently improves the performance of the trained model. For instance, there is only one metric on which the approach without regularization outperforms its regularized counterpart (i.e., reprojection error on the GoPro dataset with downsampling factor $\delta = 4$). However, this unique favorable outcome for the approach without regularization is merely

Table 6.7: Regularization ablation study for the experiment on the **GoPro** dataset (Placht et al., 2014). **Consistency**: mean-absolute displacement (and the 90th quantile) between predictions on high and low-resolution images downsampled by δ. **Reprojection Error:** corresponding errors in corner reprojection [and the number of fully detected boards]. In units of original pixel size.

	CONSISTENCY		
	$\delta = 2$	4	6
WITHOUT COUNT REGULARIZATION	0.199 (0.37)	0.369 (0.66)	0.562 (0.99)
WITH COUNT REGULARIZATION	**0.133 (0.24)**	**0.244 (0.43)**	**0.378 (0.66)**

	REPROJECTION ERROR		
	$\delta = 2$	4	6
WITHOUT COUNT REGULARIZATION	0.052 [100]	**0.172** [48]	0.460 [100]
WITH COUNT REGULARIZATION	**0.046** [100]	0.198 [82]	**0.417** [100]

due to the much lower recall it reports; indeed, the metric is thus computed on the easiest samples only which positively biases the outcome. Overall, counting-based regularization undoubtedly improves the learning of sub-pixel point localization.

6.4.3 Sub-frame Temporal Event Detection in Videos

As mentioned earlier, the precise *temporal* localization of *point* events in sequential data (i.e., answering the question “when do instantaneous events occur?”) is a widespread task with applications in numerous fields from accurate audio-to-score music transcription, to the detection of sports events in videos. In contrast to action extents prediction, a task that is prone to high levels of temporal ambiguity (Sigurdsson et al., 2017), instantaneous event detection is characterized by the sharp temporal accuracy required of the predictions. Indeed, temporal point predictions often have to fall within a narrow margin of the ground-truth location to be of any practical use. In recent years, several works have achieved state-of-the-art results on diverse temporal detection tasks by leveraging the generalization capabilities of deep neural networks (Hawthorne et al., 2019; Wu et al., 2018; McNally et al., 2019).

However, these standard approaches, since they perform dense classification on discrete sequences, limit the potential precision of the prediction to the temporal

Table 6.8: Regularization ablation study for the the **uEye** dataset (Placht et al., 2014). **Consistency**: mean-absolute displacement (and the 90th quantile) between predictions on high and low-resolution images downsampled by a factor δ. **Reprojection Error:** corresponding errors in corner reprojection [and the number of fully detected boards]. In units of original pixel size.

	Consistency		Reprojection Error	
Methods	$\delta = 2$	4	$\delta = 2$	4
Without Count Regularizer	0.192 (0.36)	0.501 (0.90)	0.059 [148]	0.085 [137]
With Count Regularizer	**0.134** (**0.23**)	**0.348** (**0.64**)	**0.055** [200]	**0.073** [187]

resolution (frame rate) of the input. Thus, millisecond precision can only be achieved by using data with millisecond temporal resolutions.

In this section, we show that the loss function introduced in Section 6.3.2 can be leveraged not only for spatial applications, but also for sequential data to achieve *sub-frame* temporal detection. Indeed, by inferring event occurrence times directly in $\mathbb{R}$ rather than on a discrete timeline (Hawthorne et al., 2019; Wu et al., 2018; McNally et al., 2019), our approach decouples the precision of the predictions from the resolution of the input sequence, and can thus output accurate predictions without the need for high temporal resolution inputs. This alleviates the need for high-resolution data collection and processing and significantly reduces the computational burden.

Experiment Specifications

In this section, we modify the previously introduced experiment (Section 5.6.1) proposed by McNally et al. (McNally et al., 2019) on golf swing events detection in videos. In order to evaluate the sub-frame capability of our model and its ability to infer precise localization in low-resolution settings, we downsample the training and testing videos with a temporal decimation rate δ. A wide spectrum of downsampling rates are considered, ranging from the original experiment ($\delta = 1$) to highly downsampled settings where only 1 out of 16 frames of the video samples are kept ($\delta = 16$). Since the tolerance within which a prediction is considered correct (i.e., ±1 frame of the original resolution) is kept unchanged across all experiments, the task becomes progressively more challenging as the downsampling rate δ increases. Indeed, even though the downsampled sequences retain less

Table 6.9: Golf swing event detection accuracy (within a ±1 frame tolerance) as a function of the downsampling factor δ. Averages and standard deviations (in brackets) are reported over 4 folds. The model architecture is from (McNally et al., 2019). The temporal upsampling is performed using the state-of-the-art frame interpolation method proposed by (Bao et al., 2019).

	$\delta = 1$ frame	2 frames	4 frames	8 frames	16 frames
Naïve upsampling	67.6 (0.8)	68.5 (0.7)	59.8 (1.3)	44.7 (1.0)	23.9 (0.5)
Frame interpolation	— " —	67.4 (0.6)	67.1 (0.6)	60.5 (1.3)	41.6 (1.9)
Prediction upsampling	— " —	69.6 (0.6)	69.9 (0.6)	66.3 (1.1)	57.8 (1.2)
Ours	**70.9** (1.4)	**70.4** (1.2)	**70.7** (1.3)	**69.8** (1.4)	**60.6** (1.6)

and less information, predictions are expected to remain as precise as in higher resolution settings. (The code from (McNally et al., 2019) was used as is, without any fine-tuning in all experiments.)

Our Approach

The continuous heatmap-matching loss function (Equation 6.3) can be adapted for 1-dimensional applications simply by dropping all dependence on y. Thus, the model is trained to infer, for each timestep in the sequence, temporal offsets $\Delta^x \in [0,1]$ and event occurrence probabilities $\mathbf{p} \in [0,1]^d$ for each event class. Since our loss is agnostic to the underlying model, it can be directly applied in conjunction with the architecture proposed in the original paper (McNally et al., 2019). Once again, we leverage the properties of the counting-based regularization to achieve prediction sparsity (see Section 6.3.3).

Benchmarks

McNally et al. (2019) leverage the widely used (e.g., (Hawthorne et al., 2019; Wu et al., 2018)) standard average stepwise cross-entropy as loss function. As this loss function requires the predictions to be set on a discrete grid, we consider two different video temporal upsampling regimes to augment the original model with sub-frame detection capabilities. The first one consists in duplicating each frame of the input δ times in order to match the original ($\delta = 1$) sequence resolution (*Naïve upsampling*), while the second leverages the state-of-the-art frame interpolation

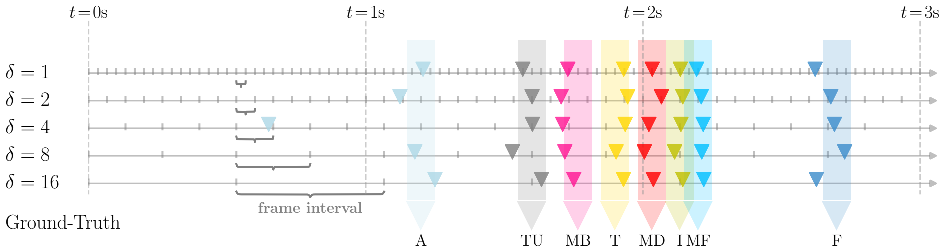


Figure 6.8: Consistency of our temporal point predictions across all resolutions. The colored triangles represent the predictions for each of the 8 different classes, while the patches illustrate the tolerance window of 1 frame around the ground-truth.

method proposed by (Bao et al., 2019) to estimate the $\delta-1$ missing frames (*Frame interpolation*). We also consider an additional benchmark that operates on the downsampled resolution without any explicit input upsampling: instead of inferring only one event probability per timestep, the model infers δ probabilities, one for the current timestep and one for each of the $\delta-1$ missing timesteps in an effort to match the original resolution of the predictions (*Prediction upsampling*). This final benchmark is reminiscent of the upsampling-based approach used in (Nehme et al., 2018; Hu et al., 2019).

Experiment Results

Table 6.9 shows that our approach outperforms the traditional ones for all downsampling factors δ; the performance gap becomes even more apparent as the downsampling rate is increased. For instance, our loss function allows for the training of a very competitive golf event detector using only 1 out of 8 frames of the original video (i.e., $\delta=8$). This prediction consistency across the various downsampling rates for a given test sequence is depicted in Figure 6.8. This figure confirms, once again, that our approach yields both consistent event location predictions across all downsampling rates and that, in low-resolution videos, the model achieves a temporal precision way beyond frame accuracy.

These results overall demonstrate that our proposed approach does not only achieve precise multi-instance sub-pixel detection accuracy in spatial applications, but can also be effective for sub-frame temporal event detection. (Note that additional results with detailed per event class metrics can be found in

Table 6.10: Ablation study. Golf swing event detection accuracy (within a ±1 frame tolerance) as a function of the decimation factor δ. Averages and standard deviations (in brackets) are reported over 4 folds. The architecture and other experiment specifications are from (McNally et al., 2019).

	$\delta = 1$ frame	2 frames	4 frames	8 frames	16 frames
Without Count Regularizer	69.7 (1.9)	69.3 (1.8)	69.7 (1.0)	68.0 (1.6)	59.3 (0.8)
With Count Regularizer	**70.9** (1.4)	**70.4** (1.2)	**70.7** (1.3)	**69.8** (1.4)	**60.6** (1.6)
Diff.	+1.2	+1.1	+1.0	+1.8	+1.6

Appendix C.3.) Additionally, by being able to operate on lower resolution inputs without any significant performance deterioration, our approach allows for both a more efficient training and a faster inference, which is key for low-resource and real-time applications, especially on mobile and embedded devices.

Ablation Study

Table 6.10 reveals that the counting regularization consistently improves the detection performance of our model on the golf swing event sequencing experiment. Indeed, training with this regularizer yields at least a 1% improvement in accuracy on all downsampling levels, when compared to the results obtained when training with $\mathscr{L}_{\text{HM}}$ alone. This increase is all the more substantial since the particular task introduced by McNally et al. (2019) does not necessarily require complex post-processing operations to be solved successfully. Indeed, the sequences in the dataset contain exactly one occurrence for each event class (see Section 5.6.1 for more details about the characteristics of the golf video sequencing dataset), and thus a simple global maximum-picking operation can be used to achieve detection sparsity since multi-instance detection capabilities are not essential. Overall, this result highlights that the incorporation of prediction sparsity directly in the learning process through count-based regularization can further improve the model even in settings where prediction sparsity is easily achievable.

6.5 Conclusion

In this chapter, we propose dense offset regression, continuous heatmap-matching-based learning, and instance counting regularization to improve multi-instance sub-pixel localization accuracy. The novel localization learning loss function introduced to that effect—which allows for the end-to-end learning of sub-pixel point localization—is derived as a continuous generalization of standard heatmap-matching approaches. The model demonstrates strong performance on molecule localization microscopy, checkerboard corner detection, and we show how this paradigm can also be leveraged for sub-frame temporal video event detection.

Most importantly, the experiments and examples of this chapter highlight, once again, the usefulness of the Poisson-binomial loss function as a means of improving *convergence* and prediction *sparsity.* Indeed, similar to the observations made in the previous chapter, the use of the Poisson-binomial counting-based regularization—and by extension its unique sparsity-inducing properties—alleviates the need for sub-optimal post-processing operations, and thus allows for the fully *end-to-end* learning of sub-pixel point detection—leading to improved spatial precision. In addition to sparsity consideration, the regularization is shown to improve the spatial convergence of the prediction towards the ground-truth location—inducing further improvements in the spatial precision of the predictions. Finally, various ablation studies confirm that incorporating the prediction sparsity objective into the learning process systematically improves sub-pixel localization capabilities.

Chapter 7

Avenues for Future Research

The previous chapters showed how Poisson-binomial counting-based learning can be leveraged, among other applications, for weakly-supervised temporal localization, robust temporal point event detection, and for improved multi-instance sub-pixel point detection. This list does, however, not constitute a full account of the potential applications of the proposed framework. Therefore, in order to better capture the versatility of Poisson-binomial counting, this chapter presents several additional avenues for future research.

7.1 Semi-Supervised Learning through Counting-based Consistency Regularization

Unlabeled data, while being easily collectible, are often disregarded in standard learning pipelines. However, even without containing any labels, unannotated samples can carry a vast amount of information. For instance, real-life images are not unstructured compositions of random pixels, but instead are comprised of coherent visual patterns. Learning these underlying visual structures can already be a key step towards a finer-grained understanding of image content (Pathak et al., 2016; Noroozi et al., 2017; Caron et al., 2018; Gidaris et al., 2018).

In contrast to fully-supervised learning—which discards all unlabeled samples—and unsupervised learning—which only relies on unlabeled data for training, *semi-supervised learning* (Chapelle et al., 2006) offers an in-between solution that not only takes advantage of labeled samples as a fine-grained learning input, but also utilizes unlabeled data as a rich additional source of information. These models thus effectively reduce the need for costly annotations, which often constitute the main bottleneck in the dataset creation process, by allowing to integrate parts of the unfathomable amounts of unannotated data that are being generated every day into the learning process.

This section shows how Poisson-binomial counting can be leveraged, in some settings, to alleviate some of the weaknesses of consistency regularization—a state-of-the-art semi-supervised learning approach.

7.1.1 Consistency Regularization

Among the vast literature on semi-supervised learning (Joachims, 2003; Zhu et al., 2003; Grandvalet & Bengio, 2005; Lee, 2013; Kingma et al., 2014; Oliver et al., 2018; Berthelot et al., 2019), consistency regularization (Sajjadi et al., 2016; Laine & Aila, 2017; Oliver et al., 2018; Berthelot et al., 2019) offers a simple, yet effective, way of integrating unlabeled samples into the training process. Indeed, by merely requiring the inclusion of a regularization term to the loss function, this architecture-agnostic approach can augment almost any standard fully-supervised models with semi-supervised learning capabilities.

In a nutshell, consistency regularization ensures that models output *consistent* predictions when stochastic label-invariant perturbations are applied to the input data. First, in the simplest form of the paradigm (Berthelot et al., 2019), each sample $\mathbf{X}$ is transformed twice independently using a stochastic augmentation function Ω on it. The most important aspect of the transformation function is that it should theoretically not affect the predictions of the model (i.e., $\hat{\mathbf{f}}_\theta(\Omega(\mathbf{X})) \approx \hat{\mathbf{f}}_\theta(\mathbf{X})$). For instance, in image classification, the collection of transformations could include image rotation, color perturbations, or slight cropping since these functions do not impact the class semantic. Indeed, an upside-down or a grayed-out image of a cat still represents a cat. As the two perturbed images are known per design to have the same underlying label, the predictions of the two inputs can then be compared and used as a consistency regularization. For instance, in classification, the following consistency loss function is defined (Grandvalet & Bengio, 2005; Lee, 2013):

$$\left\| \hat{\mathbf{f}}_\theta(\Omega(\mathbf{X})) - \hat{\mathbf{f}}_\theta(\Omega(\mathbf{X})) \right\|_2^2 \tag{7.1}$$

In contrast to what one might initially assume, this equation does not cancel out since the transformation Ω is assumed to be stochastic, and thus the perturbed input $\Omega(\mathbf{X})$ is also stochastic. Of course, while variations of this approach exist—

e.g., (Tarvainen & Valpola, 2017; Miyato et al., 2018), the underlying principle remains the same.

Consistency regularization methods, therefore, rely on the prior knowledge that the predictions (e.g., class assignments in classification) are invariant to input perturbations to incorporate unlabeled data into the learning process. Indeed, Equation 7.1 does not require any labels to be computed, and thus can be leveraged for unannotated samples as well. Overall, these approaches seek to improve the learning of meaningful representations by encouraging the models to learn these known underlying invariances from unlabeled samples. Of course, labeled samples are still extremely important as they ensure that the predictions are correct in addition to being consistent.

7.1.2 Some Limitations of Consistency Regularization

Consistency regularization relies on the existence of a balanced perturbation function that produces rich enough input transformations, while retraining the semantic information that is needed to make a correct prediction. Indeed, on the one hand, leveraging only mild augmentations (i.e., $\Omega \approx id$) does not teach any relevant invariances to the model, and thus makes the use of unlabeled samples almost pointless. On the other hand, perturbing the input to the point of affecting the underlying ground-truth can only negatively impact the learning of meaningful representations.

Depending on the application, defining an effective and rich task-specific augmentation function Ω is challenging. Indeed, while many class-invariant transformations exist in the visual domain (e.g., rotation, shifting, and cropping), other finer-grained tasks present a lower level of invariance. For instance, in object detection, the position of the bounding-box labels is not invariant to spatial transformations. The same observation can also be made about the even finer-grained pixel-level annotations in image segmentation. While heuristics and other color-based perturbations could be leveraged in these examples, the nature of lower-level tasks often limits the use of a wide range of transformations, and thus makes the consistency regularization less effective in these finer-grained settings.

Another challenge consists in defining a consistency measure that accurately reflects the consistency between different predictions. Indeed, while Equation 7.1 is a common choice in classification and regression tasks, other tasks have a more complex definition of consistency. For example, in object detection, models often output several bounding-boxes surrounding the same object. Thus, the consistency metric should be almost unaffected by which one of these similar alternatives is ultimately selected as detection. Such a function can, however, be highly cumbersome to design efficiently. In fact, this feature is often overlooked in object detection where the standard approach consists in over-sampling the space with high-likelihood bounding-boxes (Girshick, 2015; Ren et al., 2015; Redmon et al., 2016; Liu et al., 2016), before selecting a sub-sample of them through non-maximum suppression or other post-processing operations. The fact that a two-step process is often preferred over single loss optimization (e.g., optimize a MAP-based measure (Henderson & Ferrari, 2016)) when performing object detection highlights the difficulty of implementing an effective consistency measure in this setup. The definition of a consistency measure is also not straightforward in temporal event detection tasks (e.g., video event detection and music transcription). Indeed, designing a measure that takes into account the consistency of both the existence probabilities and the temporal localization, while being robust to small temporal shifts in the predictions, is challenging.

Overall, consistency regularization can be difficult to implement effectively and efficiently, especially in finer-grained tasks such as object detection and temporal event detection.

7.1.3 Counting-based Consistency Regularization

The previous chapters showed how instance *counting* can be applied both as a weaker form of supervision in some settings (Chapter 4) and as a loss regularization that is inherently more robust to noisy annotations than more targeted supervision (Chapter 5). The same approach of operating on a weaker level of supervision can be used to alleviate the issues encountered when applying consistency regularization to tasks with finer-grained annotations. In this section, we thus propose counting-based consistency regularization as an effective alternative for applications dealing with countable instances (e.g., object detection and temporal event localization).

Using higher-level annotations—such as counts—implicitly alleviates the issues highlighted in the previous section. First, weaker labels are inherently invariant to a wide range of transformations. For example, in object detection, while the location of the bounding-boxes changes when the input image is subject to rotation or cropping, the number of instances remains invariant to almost any spatial transformation. The same observation holds in music transcription, where the number of notes played is preserved when the input sequence is slowed down or shifted temporally, whereas the temporal location of the instances are affected by these perturbations. Thus, counting-based consistency regularization allows to effectively leverage a wider range of augmentations, consequently teaching a rich array of semantic invariances to the model and thus better utilizing the information contained in the unlabeled data.

In addition, defining a counting-based consistency measure is straightforward, even for complex tasks such as object detection or temporal event localization. Indeed, assuming that the model outputs class (or existence) probabilities for each instance—in addition to other predictions such as, for example, object positions or bounding-box dimensions, i.e., $\{\mathbf{p}_1, \ldots, \mathbf{p}_N\} = \hat{\mathbf{f}}_\theta(\mathbf{X}) \in [0,1]^N$, the consistency between two counts can be measured using a simple squared distance:

$$\left\| \textstyle\sum_i \hat{\mathbf{f}}_{\theta,i}(\Omega(\mathbf{X})) - \sum_i \hat{\mathbf{f}}_{\theta,i}(\Omega(\mathbf{X})) \right\|_2^2 \tag{7.2}$$

Counting-based supervision thus offers an intuitive alternative to standard consistency regularization that is both invariant to a broader range of perturbations and presents a simple consistency measure.

7.1.4 Poisson-binomial Counting-based Consistency Regularization

Once again, if the ground-truth instance probabilities are known to be *sparse* and if the supervised learning loss function used in conjunction with the consistency regularization supports sparse predictions, then a more tailored learning can be achieved by modeling counts per class as Poisson-binomial distribution rather than scalars, i.e.,

$$\hat{\mathscr{C}}_\theta(\Omega(\mathbf{X})) := \textstyle\sum_i \mathcal{B}\Big(\hat{\mathbf{f}}_{\theta,i}(\Omega(\mathbf{X}))\Big)\,. \tag{7.3}$$

In this stochastic setup, a consistency measure can be defined as the log-likelihood of observing an identical count for the two perturbed inputs:

$$\begin{aligned}
&\log\Big(\Pr\Big(\hat{\mathscr{C}}_\theta(\Omega(\mathbf{X})) = \hat{\mathscr{C}}_\theta(\Omega(\mathbf{X}))\Big)\Big)\\
&\quad= \log\Big(\textstyle\sum_{c=0}^{\infty} \Pr\Big((\hat{\mathscr{C}}_{\theta}(\Omega(\mathbf{X}))=c) \wedge (\hat{\mathscr{C}}_\theta(\Omega(\mathbf{X}))=c)\Big)\Big)\\
&\quad= \log\Big(\textstyle\sum_{c=0}^{\infty} \Pr\Big(\hat{\mathscr{C}}_\theta(\Omega(\mathbf{X}))=c\Big) \cdot \Pr\Big(\hat{\mathscr{C}}_\theta(\Omega(\mathbf{X}))=c\Big)\Big)\\
&\quad= \log\Big(\textstyle\sum_{c=0}^{\infty} \Pr\Big(\sum_i \mathcal{B}\Big(\hat{\mathbf{f}}_{\theta,i}(\Omega(\mathbf{X}))\Big)=c\Big) \cdot \Pr\Big(\sum_i \mathcal{B}\Big(\hat{\mathbf{f}}_{\theta,i}(\Omega(\mathbf{X}))\Big)=c\Big)\Big).
\end{aligned} \tag{7.4}$$

For readability, the consistency measure is presented for a single class since the extension to higher dimensions is trivial.

Since the modeling of counts as a distribution rather than a scalar does not affect their invariance to augmentations, Equation 7.4 offers a more tailored supervision than Equation 7.2 without limiting the range of usable transformations. Indeed, similar sparsity convergence properties to that of Chapter 3 can be derived for the proposed regularization (e.g., the loss function is minimized if and only if the predictions are sparse). Thus, in addition to teaching transformation invariances to the model, this proposed Poisson-binomial counting-based consistency regularization implicitly teaches the model to infer sparse predictions.

Future Work

The rigorous empirical evaluation of the proposed approach still needs to be done. However, based on the experiments and conclusions of the previous chapters, the more tailored supervision offered by Equation 7.4 is expected to be more effective at leveraging unlabeled data in fine-grained tasks that are known to have sparse ground-truth instance probability assignments.

7.2 Sparse Adversarial Attack on Image Classification

While deep learning models have demonstrated remarkable performance in image classification (Krizhevsky et al., 2012), they have also been shown to be extremely vulnerable to targeted alterations of the input image (Szegedy et al., 2013). Indeed, small image-specific perturbations that are almost undetectable to the human eye can often be crafted to fool a trained neural network (Szegedy et al., 2013; Goodfellow et al., 2014; Papernot et al., 2016; Moosavi-Dezfooli et al., 2016). Moosavi et al. (2017) further proved that highly effective universal (i.e., image-agnostic) adversarial perturbations often exist. Overall, the development of such adversarial attack methods is not only useful for assessing the robustness of existing models, but more importantly is key for obtaining a better understanding of deep learning and ultimately designing models that are more effective (e.g., adversarial training (Goodfellow et al., 2014; Huang et al., 2015)).

7.2.1 One Pixel Adversarial Attack

Su et al. (2019) showed that image classification models are prone to misclassification even when perturbing only one pixel of the input image is perturbed. To that effect, they propose performing a d-pixel targeted-attack by solving the following optimization problem:

$$\max_{\pi} \hat{f}_{\theta,i_{adv}}(\mathbf{X}+\pi(\mathbf{X})) \quad \text{s.t.} \quad \|\pi(\mathbf{X})\|_0 \leq d, \tag{7.5}$$

where $\hat{f}_\theta$ is the trained classification function that has to be fooled, the index i_{adv} is the desired output class of the model after the perturbation $\pi(\mathbf{X})$ is applied to the input image $\mathbf{X}$, and d is the selected maximum number of pixels that can be perturbed for the adversarial attack. In a nutshell, successfully maximizing this ℓ_0-constrained equation helps to find the d-pixel perturbation $\pi(\cdot)$ that, when applied to the input image $\mathbf{X}$, maximizes the model's misclassification for the wrong class i_{adv}. The objective of this attack is thus simply to modify the input image $\mathbf{X}$ in such a way that the model classifies the slightly perturbed input $\mathbf{X}+\pi(\mathbf{X})$ as an image of class i_{adv}, instead of the true class i_{true}. A d-pixel untargeted-attack can be performed in a similar manner by minimizing the probability that the model infers for the correct class i_{true}. Overall, Su et al. (2019) demonstrate that altering a single pixel (i.e, $d = 1$) is often enough to deceive classical models.

This d-pixel approach strongly contrasts with classical methods, which perturb every pixel in the input image while ensuring that the total amount of alterations remains of small magnitude in order to be as unnoticeable as possible, e.g.,

$$\max_{\pi} \hat{f}_{\theta,i_{adv}}(\mathbf{X}+\pi(\mathbf{X})) \quad \text{s.t.} \quad \ell_1(\pi(\mathbf{X})) \leq L, \tag{7.6}$$

where L represents an arbitrary upper-bound for the magnitude of the image perturbation $\pi(\mathbf{X})$. While the two maximization problems might appear similar at first sight, the change from an ℓ_1 to an ℓ_0 constraint has a significant impact on the optimization process. Indeed, while the latter maximization problem (Equation 7.5) is often solved through simple backpropagation (Szegedy et al., 2013; Goodfellow et al., 2014), the non-differentiability of the ℓ_0-norm makes this simple approach significantly less compelling for the optimization of Equation 7.6. Therefore, Su et al. (2019) propose solving the ℓ_0-constrained maximization problem through differential evolution (Storn & Price, 1997; Das & Suganthan, 2010), a specific form of evolutionary algorithm.

In the next section, we will show how the sparsity-inducing ability of the Poisson-binomial counting loss function can be leveraged to nevertheless optimize Equation 7.6 through backpropagation.

7.2.2 Sparse Attack with Poisson-Binomial Counting

As shown in Section 5.5.2, the optimization of a function under constraints can be done through backpropagation using a *differentiable* penalized objective function.

Attack Maximization

The main objective of an adversarial attack consists in optimizing the perturbations π in such a way that the classification model $\hat{f}_\theta$ is deceived as much as possible. More specifically, a targeted attack attempts to maximize the probability the model infers for a target adversarial class i_{adv}:

$$\max_{\pi} \hat{f}_{\theta,i_{adv}}(\mathbf{X} + \pi(\mathbf{X})). \tag{7.7}$$

This maximization problem is, however, similar to a standard classification task, but with the wrong label assignment. In this scenario, the cross-entropy is, therefore, the natural choice of loss function to optimize the perturbation through backpropagation, i.e.,

$$\max_{\pi} \hat{f}_{\theta,i_{adv}}(\mathbf{X} + \pi(\mathbf{X})) = \min_{\pi}\{-\log(\hat{f}_{\theta,i_{adv}}(\mathbf{X} + \pi(\mathbf{X}))\} =: \min_{\pi} \mathcal{L}_{\mathrm{TAR}}(\pi). \tag{7.8}$$

Similarly, the loss function for an untargeted attack, which seeks to reduce the probability that the model infers for the true class i_{true}, can be defined as a *log*-likelihood function:

$$\min_{\pi} \hat{f}_{\theta,i_{true}}(\mathbf{X} + \pi(\mathbf{X})) = \min_{\pi}\{-\log(1 - \hat{f}_{\theta,i_{true}}(\mathbf{X} + \pi(\mathbf{X}))\} := \min_{\pi} \mathcal{L}_{\text{UNT}}(\pi). \tag{7.9}$$

While minimizing these loss functions alone yields effective attacks, the resulting perturbations they produce do certainly not fulfill the d-pixel limitation set by the optimization problem (Equation 7.5). The next section, therefore, shows how the Poisson-binomial counting loss function can be utilized as a sparsity regularization to impose this strict sparsity constraint.

Counting Sparsity Constraint

The Poisson-binomial counting loss function offers a unique way of encouraging—and even sometimes forcing—the model to output, for each training sample, a strict and given number of non-zero instances. This controllable sparsity is highlighted by Theorem 3.1, which state that, for all $c \in \mathbb{N}$

$$\mathcal{L}_{\text{PB}}(\hat{\mathbf{p}} \mid c) := D_{KL}(\mathbb{1}_c \| \textstyle\sum_i \mathcal{B}(\hat{p}_i)) = 0 \iff (\|\hat{\mathbf{p}}\|_0 = c) \wedge (\hat{\mathbf{p}} \in \{0,1\}^N). \tag{7.10}$$

Indeed, this equation indicates that, in some settings, the Poisson-binomial counting loss function can be used as a differentiable replacement for the constraint $\|\cdot\|_0 = c$. Thus, in contrast to sparse activation functions (Martins & Astudillo, 2016; Martins & Kreutzer, 2017; Malaviya et al., 2018), which produce sparsity with only limited control on the number c of non-zero instances, the Poisson-binomial loss function can be leveraged to teach models to output *exactly* c non-zero instances. Several applications of this principle have been presented in Chapter 5 and Chapter 6.

In this section, the differentiability of the Poisson-binomial loss function and most importantly its unique controllable sparsity-inducing abilities are used to optimize the d-pixel targeted-attack problem using backpropagation (Equation 7.5).

First, one of the main assumptions underlying the sparsity regularization through Poisson-binomial counting is that the optimal non-zero instance predictions have to be exactly equal to 1 as shown by the second conditional in Equation 7.10

(i.e., $\hat{\mathbf{p}}_\theta \in \{0{,}1\}^T$). In the context of the d-pixel adversarial attack, the instances that are regularized through the ℓ_0-constraint are the elements of the perturbation $\pi(\mathbf{X})$. However, the optimal perturbation is not necessarily contained in $\{0{,}1\}^{N\times M}$—where $N \times M$ is the dimension of the input image $\mathbf{X}$—and thus the Poisson-binomial loss function cannot be directly applied to enforce the constraint $\|\pi(\mathbf{X})\|_0 \leq d$ from Equation 7.5. To that effect, we propose to first decompose the perturbation π as a product of masking probabilities $\mathbf{p}$ and perturbation magnitudes $\mathbf{m}$:

$$\pi(\mathbf{X}) = \mathbf{p}(\mathbf{X}) \cdot \mathbf{m}(\mathbf{X}). \tag{7.11}$$

In a nutshell, for each pixel in the original image, its corresponding masking probability $\mathbf{p}_i$ indicates the likelihood of it being altered by the perturbation, while $\mathbf{m}_i$ indicates the magnitude of the potential alterations. After the decomposition, given the setting of the d-pixel attack problem (Equation 7.5), the optimal $\mathbf{p}$ now fulfill both $\|\hat{\mathbf{p}}\|_0 \leq c$ and $\hat{\mathbf{p}} \in \{0{,}1\}^{N\times M}$. Further, the constraint $\|\hat{\mathbf{p}}\|_0 \leq c$ can intuitively be modified to $\|\hat{\mathbf{p}}\|_0 = c$ since reducing the number of perturbed pixels can only decrease the effectiveness of the adversarial attack. Since both conditionals on the right-hand side of Equation 7.10 are now fulfilled by the optimal masking probabilities $\mathbf{p}$, these optimal $\mathbf{p}$ are conversely those that minimize the Poisson-binomial loss function with $c := d$. Therefore, we propose to use Poisson-binomial sparsity regularization $\mathbf{p}$ to enforce the sparsity constraint on the perturbation.

Penalized Objective Function

Thus, as already demonstrated in Section 5.5.2, an unconstrained version of the d-pixel constrained optimization problem (Equation 7.5) can be derived by integrating the sparsity constraint as a penalty function to the attack-maximization objective function, e.g.,

$$\min_{\mathbf{p}(\cdot),\mathbf{m}(\cdot)} (1-\lambda) \cdot \mathcal{L}_{\mathrm{TAR}}(\mathbf{p}(\mathbf{X}) \cdot \mathbf{m}(\mathbf{X})) + \lambda \cdot \mathcal{L}_{\mathrm{PB}}(\hat{\mathbf{p}}(\mathbf{X}) \mid d) \tag{7.12}$$

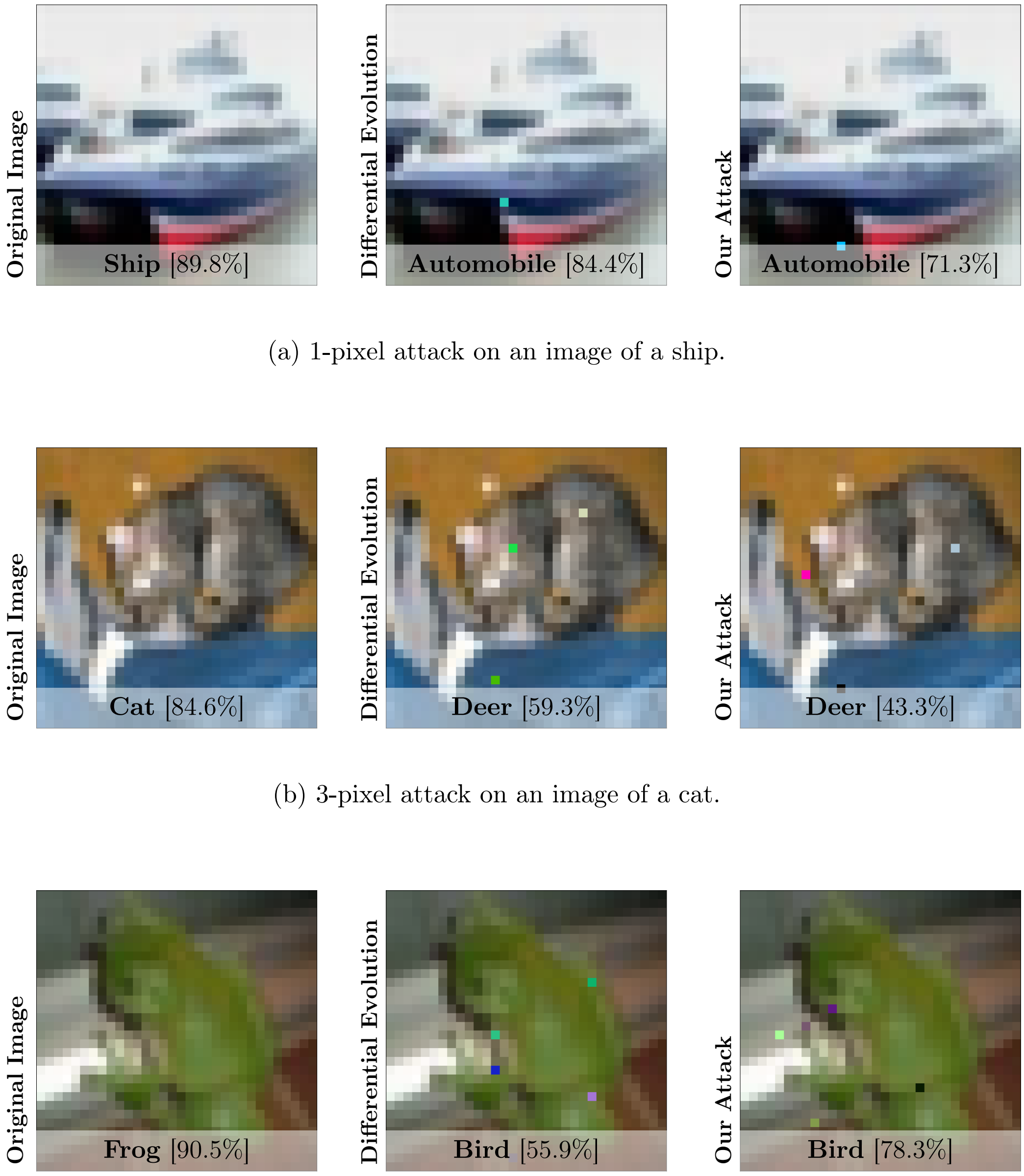
where the weight λ is gradually increased during the training to progressively enforce the constraint. The loss function for the untargeted attack is analogous. Overall, since all elements of the loss function are continuously differentiable, the learning of the perturbation $\pi(\mathbf{X}) = \mathbf{p}(\mathbf{X}) \cdot \mathbf{m}(\mathbf{X})$ can be performed through backpropagation. However, it has to be noted that, while the existence of a differentiable loss function for the learning of the attack (e.g., Equation 7.12) allows the training to be done through backpropagation, this approach does not guarantee a convergence towards the global optimum since the optimization problem is highly *non-convex*. This is discussed in more detail below.

Preliminary Experiment

In this section, the proposed backpropagation-based optimization of the d-pixel perturbation is applied to fool the LeNet architecture (LeCun et al., 1989) on some samples of the CIFAR-10 dataset (Krizhevsky et al., 2009). More specifically, for each sample $\mathbf{X}$, the pre-sigmoid masking probabilities $\mathbf{p}(\mathbf{X})$ and perturbation magnitudes $\mathbf{m}(\mathbf{X})$ are first initialized using a normal distribution with standard deviations 0.1 and 0.5 and means $\log(0.8^{2/32\times 32} - 1)$ and 0 respectively—see Section 2.3.3 for more precision about the bias required for a sound initialization—and 0 respectively. These initial values are then iteratively updated to minimize Equation 7.12 through 5000 timesteps of gradient descent optimization. A large learning rate of 100 is chosen in order to compensate for the tiny gradients the model produces. Finally, in this experiment, a fixed λ (c.f. Equation 7.12) of 0.9 was shown to be sufficient to achieve prediction sparsity.

Figure 7.1 presents a few examples of successful adversarial attacks using the proposed backpropagation-based algorithm. For instance, in Figure 7.1a, the approach was able to modify the class inferred by the LetNet model from *ship* (with a probability of 89.9%) to *automobile* (with a probability of 71.3%) by altering the color of a single pixel in the input image. Remarkably, in these examples, while performing the untargeted attack through deferential evolution (Su et al., 2019) or backpropagation produces different perturbations, the wrong object class towards which the predictions converge is identical. Overall, these examples not only highlight the high sensitivity of the LetNet model to a few altered pixels, but most importantly show that backpropagation can be leveraged for d-pixel attacks on image classification.

Finally, it has to be emphasized that the proposed backpropagation-based approach optimizes the sparsity of the perturbations in an end-to-end manner. Indeed, Figure 7.1 presents the raw perturbed inputs produces by the model, without any additional post-processing heuristic. This underlines once again the unique

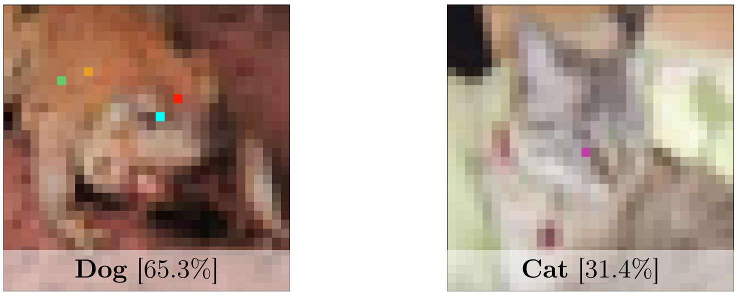


(a) 1-pixel attack on an image of a ship.

(b) 3-pixel attack on an image of a cat.

(c) 5-pixel attack on an image of a frog.

Figure 7.1: Examples of successful untargeted adversarial attacks using the proposed backpropagation-based algorithm. The object name and the percentage stand for the most likely object class inferred by LeNet and its corresponding probability. The sparsity of our model is learned in an end-to-end manner, and thus is achieved without any post-processing operation.



(a) 3-pixel attack.

(b) 1-pixel attack.

Figure 7.2: Examples of samples where the proposed backpropagation-based algorithm failed the untargeted *d*-pixel adversarial attacks, while the differential evolution approach (Su et al., 2019) succeeded: (a) the 3-pixel constraint is violated since 4 pixels where altered, and (b) the attack—while reducing the probability assigned to the correct object class—does not modify the class the LeNet model infers.

ability of the Poisson-binomial counting loss function to enforce a theoretically non-differential operation in an end-to-end manner.

Limitations and Future Work

As briefly mentioned above, the *d*-pixel optimization problem (Equation 7.5) is highly non-convex. Thus, while the proposed differentiable penalized objective function (Equation 7.12) is minimized if and only if the optimization problem is solved, its optimization through backpropagation does not necessarily guarantee the convergence of the perturbation towards the global maximum. Figure 7.2 depicts some examples where the optimization got stuck in a local maximum, and thus where the proposed learning procedure failed. For instance, in Figure 7.2b, the resulting 1-pixel adversarial attack reduces only slightly the probability assigned to the correct class, while the differential evolution-based approach (Su et al., 2019) successfully manages to mislead the model. These sub-optimal perturbations clearly demonstrate that gradient-based learning alone is not sufficient in such highly non-convex optimization problems.

Future work could focus on combining the proposed backpropagation-based method with population-based learning. Indeed, while the latter offers a more dense coverage of the solution space, and thus allows for a more complete exploration of the space, the former could be utilized to perform a fine-grained gradient-based update of the population which could better take advantage of the local structure of the optimization problem. Overall, this hybrid optimization approach could increase the likelihood of converging towards a good local optimum perturbation, if not the global optimum.

7.3 Learning Non-maximum Suppression in Object Detection

As already mentioned, object detection is often a two-step process that consists in over-sampling the space with numerous high-quality bounding-boxes before refining the selection through non-differential operations such as non-maximum suppression (Girshick, 2015; Ren et al., 2015; Redmon et al., 2016; Liu et al., 2016). However, while they have proven to be effective, these approaches split the object detection process into a trainable component and fixed heuristic and, as a result, optimize the model parameters on a sub-task only (i.e., outputting a large collection of meaningful bounding-boxes) rather than on the final detection objective. By not performing the learning of the task in a fully end-to-end manner—a paradigm that has been at the core of the advent of deep learning (Krizhevsky et al., 2012; Sutskever et al., 2014; Long et al., 2015; Levine et al., 2016), the models introduce a potentially sub-optimal dichotomy between training objective and task objective. Indeed, for instance, as demonstrated in Chapter 5 and Chapter 6, effectively replacing untrainable post-processing stages with tailored training objectives often leads to improved performance since the model directly learns the mapping between the input data and the objective of the task without relying on uncontrollable operations for inference.

7.3.1 End-to-end Learning with Non-maximum Suppression

Several alternatives have been proposed for integrating the non-maximum suppression operation into the training process, and thus learning object detection in an end-to-end manner. The first approach consists in including the non-differentiable operation in the forward path, but then replacing the ill-defined gradients with pseudo gradients during the backpropagation. For example, Henderson & Ferrari (2016) directly optimize the mean-average precision (i.e., the standard performance metric in object detection) by computing pseudo partial derivatives for piecewise-constant functions through either single-sided finite difference or linear envelope estimation. Thus, they replace the gradients of the combined non-maximum suppression and metric computation layer with smoothed pseudo gradients in the backward path of the gradient-based optimization algorithm. While offering a fully end-to-end supervision of object detection, this method yields however sparse gradients that affect the effectiveness of the learning process. For instance, since not every change in the detection score of some bounding-boxes necessarily impacts the overall mean-average precision, their approach implicitly does not take all predictions into account for the optimization; thus, some instances are indirectly removed from the learning process. Further, Song et al. (2016) introduce as an alternative a dynamic programming algorithm that enables the training of models using non-differentiable loss functions through loss-augmented inference. While they explicitly discard non-maximum suppression from models in their experiments, the general principle could be leveraged to train models that include this specific non-differentiable operation in their forward path. Finally, Wan et al. (2015) propose a custom structured loss function for learning with a variant of the standard non-maximum suppression operation. The loss function not only incorporates information about suppressed bounding-boxes, but also requires the knowledge of which bounding-box was selected instead (i.e., which detection was responsible for the suppression). As this approach is tailored to the chosen non-maximum suppression operation, it only finds limited application outside of this framework.

7.3.2 Learning Non-maximum Suppression

Learning non-maximum suppression is another way of indirectly incorporating this non-differential operation into the decision process. Indeed, instead of explicitly including this operation as a layer in the architecture, the model can be trained to implicitly mimic its sparsity-inducing effect. Based on that idea, Hosang et al. (2017) propose a learnable rescoring scheme that updates the original detection probabilities assigned to each bounding-box with more extreme ones (i.e., closer to 0 or 1). However, while it is designed to output a sparser probability assignment, the model is unable to mimic the hard selection process induced by the NMS operation on its own. Similar to its classical counterparts, this approach still relies on an untrainable heuristic—here, thresholding—to achieve clear-cut predictions. Nevertheless, it has to be noted that the sparser rescoring of the bounding-box predictions makes this approach much less reliant on the post-processing stage than the traditional object detection models. Indeed, while the final predictions still depend on the selected thresholding factor, the results are by design quite robust to changes in that hyperparameter.

7.3.3 Learning NMS through Poisson-Binomial Counting

This work presents several applications in which the Poisson-binomial counting loss function (see Chapter 3) can be leveraged to replace non-differential sparsity-inducing operations. For instance, Chapter 5 demonstrates how learning to count instances can alleviate the need for peak-picking or thresholding-based heuristics in precise temporal event detection such as piano onset detection or video event detection. Similarly, Chapter 6 shows how counting-based regularization can replace non-maximum suppression in *point* object detection tasks such as single-molecule localization microscopy or checkerboard corner detection. In all these applications, the sparsity of the predictions is not achieved through an explicit sparsity-inducing operation, but is rather implicitly learned by the model as a byproduct of learning to count instances.

The same versatile counting-based approach to prediction sparsity could be applied to object detection. For instance, the Poisson-binomial loss function could be leveraged as part of a bounding-box rescoring scheme that aims at outputting a sparse selection of detections. However, in contrast to the work of Hosang et al. (2017) that still relies on thresholding to obtain prediction sparsity, the Poisson-binomial loss function can enforce sparsity in a totally end-to-end manner. Indeed, for example, recall Figure 4.3 (digit detection) or Figure 6.6 (single-molecule localization microscopy) which display raw model outputs without any post-processing. The counting-based loss function could thus act as a concrete trainable replacement for the non-maximum suppression operation, and thus make the inference independent from any thresholding factor.

Further, replacing the non-maximum suppression operation with Poisson-binomial counting-based learning could potentially also improve the method proposed by Henderson & Ferrari (2016). In their work, the authors acknowledge that the sparsity of the proposed gradients (i.e., not all bounding-boxes contribute to the loss computation, and thus to the optimization) can negatively affect the effectiveness of the approach. In contrast, learning sparsity through counting-based supervision would couple all the bounding-box predictions to the loss computation; indeed, any variation in the existence probability of any instance has a direct effect on the count and on the gradient computation. Every bounding-box would thus contribute to the learning process by propagating information back.

Future Work

Of course, the effectiveness of counting-based supervision as a trainable replacement for the non-maximum suppression in object detection still has to be evaluated. The main expected challenge resides in the development of a loss function that can both accurately assess the consistency between the set of predicted and labeled bounding-boxes, and that works properly with the counting-based sparsity regularization. To that end, the loss has to be able to handle sparse predictions, unlike standard loss functions (Girshick, 2015; Ren et al., 2015; Redmon et al., 2016; Liu et al., 2016). The first efforts could be devoted to combining counting-based learning with direct MAP-based training (Henderson & Ferrari, 2016).

Chapter 8

Conclusion

Not Everything That Counts Can Be Counted

This work demonstrates how instance counting can be leveraged to learn prediction sparsity in an end-to-end manner. The novel Poisson-binomial counting loss function, introduced to that effect, was indeed shown to present unique convergence properties. In a nutshell, if a model accurately learns to count using the Poisson-binomial loss function, the instance probabilities it infers will inexorably converge towards clear-cut values (i.e., either towards 0 or 1); sparsity is not learned, but rather emerges implicitly as a byproduct of learning to count these instances. This claim is not only supported theoretically by several convergence theorems (Chapter 3), but is most importantly demonstrated experimentally throughout the work.

In practice, the Poisson-binomial loss function helps, above all, incorporate the implicit prior knowledge that predictions are sparse directly into the training without harming the end-to-end learning process. The unique sparsity-inducing ability of the loss thus finds a direct use in the many applications where the optimal probability assignments are known to be sparse. This includes the detection of instantaneous events in videos or audio sequences (e.g., piano onset detection), the localization of point objects in images (e.g., single-molecule localization microscopy), and the detection of objects in images (e.g., bounding-box prediction). Overall, this work shows that the proposed cost function can not only be successfully leveraged as a standalone loss function in certain settings (Chapter 4) but can also be used as a sparsity regularization in conjunction with other more targeted loss functions to enforce sparsity constraints in an end-to-end fashion (Chapter 5–7). Among other applications, the Poisson-binomial loss function allows for improved single-molecule localization microscopy, for efficient checkerboard corner detection, and for competitive weakly-supervised learning of piano onset detection, while its invariance to label misalignment allows for a more robust video point event sequencing and a more reliable learning of wearable sensors time series detection.

In conclusion, this work not only proposes a novel loss function, but most importantly shows how careful, yet simple, loss modeling can sometimes replace complex architectural designs or can alleviate the need for ineffective heuristics.

Bibliography

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S. YouTube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

Acharya, M., Kafle, K., and Kanan, C. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8076–8084, 2019.

Adams, R. and Marlin, B. Learning Time Series Detection Models from Temporally Imprecise Labels. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 54, pp. 157–165. PMLR, 2017.

Adavanne, S. and Virtanen, T. Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network. In *Proceedings of DCASE Workshop*, pp. 12–16, 2017.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., and Parikh, D. VQA: Visual question answering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2425–2433, 2015.

Arteta, C., Lempitsky, V., Noble, J. A., and Zisserman, A. Interactive object counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 504–518. Springer, 2014.

Arteta, C., Lempitsky, V., and Zisserman, A. Counting in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 483–498. Springer, 2016.

Azadi, S., Feng, J., Jegelka, S., and Darrell, T. Auxiliary image regularization for deep CNNs with noisy labels. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2016.

Bain, M., Nagrani, A., Schofield, D., and Zisserman, A. Count, crop and recognise: Fine-grained recognition in the wild. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

Banko, M. and Brill, E. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 26–33. Association for Computational Linguistics, 2001.

Bao, W., Lai, W.-S., Ma, C., Zhang, X., Gao, Z., and Yang, M.-H. Depth-aware video frame interpolation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3703–3712, 2019.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 5049–5059, 2019.

Bilen, H. and Vedaldi, A. Weakly supervised deep detection networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2846–2854, 2016.

Böck, S., Schlüter, J., and Widmer, G. Enhanced peak picking for onset detection with recurrent neural networks. In *Proceedings of the 6th International Workshop on Machine Learning and Music (MML)*, pp. 15–18, 2013.

Bojanowski, P., Lajugie, R., Bach, F., Laptev, I., Ponce, J., Schmid, C., and Sivic, J. Weakly supervised action labeling in videos under ordering constraints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 628–643. Springer, 2014.

Boominathan, L., Kruthiventi, S. S., and Babu, R. V. Crowdnet: A deep convolutional network for dense crowd counting. In *Proceedings of the 24th ACM international conference on Multimedia*, pp. 640–644, 2016.

Born, M. and Wolf, E. *Principles of Optics.* Cambridge University Press, 1997.

Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 120: 122–125, 2000.

Brent, R. P. *Algorithms for minimization without derivatives.* Prentice-Hall, 1973.

Bulat, A. and Tzimiropoulos, G. Human pose estimation via convolutional part heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 717–732. Springer, 2016.

Caicedo, J. C. and Lazebnik, S. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2488–2496, 2015.

Camos, V. and Tillmann, B. Discontinuity in the enumeration of sequentially presented auditory and visual stimuli. *Cognition*, 107(3):1135–1143, 2008.

Cao, X., Wang, Z., Zhao, Y., and Su, F. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 734–750, 2018.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

Carreira, J., Agrawal, P., Fragkiadaki, K., and Malik, J. Human pose estimation with iterative error feedback. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4733–4742, 2016.

Chan, A. B. and Vasconcelos, N. Bayesian poisson regression for crowd counting. In *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 545–551, 2009.

Chapelle, O., Scholkopf, B., and Zien, A. *Semi-supervised learning*. MIT Press, 2006.

Chattopadhyay, P., Vedantam, R., Selvaraju, R. R., Batra, D., and Parikh, D. Counting everyday objects in everyday scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1135–1144, 2017.

Chen, B., Xiong, C., and Zhang, Q. CCDN: Checkerboard corner detection network for robust camera calibration. In *International Conference on Intelligent Robotics and Applications*, pp. 324–334. Springer, 2018.

Chen, S. X. and Liu, J. S. Statistical applications of the Poisson-binomial and conditional Bernoulli distributions. *Statistica Sinica*, pp. 875–892, 1997.

Chen, X.-H., Dempster, A. P., and Liu, J. S. Weighted finite population sampling to maximize entropy. *Biometrika*, 81(3):457–469, 1994.

Cheng, Z.-Q., Li, J.-X., Dai, Q., Wu, X., and Hauptmann, A. G. Learning spatial awareness to improve crowd counting. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Cogliati, A., Duan, Z., and Wohlberg, B. Context-dependent piano music transcription with convolutional sparse coding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12):2218–2230, 2016.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9268–9277, 2019.

Das, S. and Suganthan, P. N. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation*, 15(1):4–31, 2010.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. IEEE, 2009a.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. IEEE, 2009b.

Dittmar, C. and Gärtner, D. Real-time transcription and separation of drum recordings based on NMF decomposition. In *Proceedings of International Conference on Digital Audio Effects (DAFx)*, 2014.

Donné, S., De Vylder, J., Goossens, B., and Philips, W. MATE: Machine learning for adaptive calibration template detection. *Sensors*, 16(11):1858, 2016.

Duchenne, O., Laptev, I., Sivic, J., Bach, F., and Ponce, J. Automatic annotation of human actions in video. In *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 1491–1498, 2009.

Duda, A. and Frese, U. Accurate detection and localization of checkerboard corners for calibration. In *British Machine Vision Conference (BMVC)*, 2018.

Emiya, V., Badeau, R., and David, B. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1643–1654, 2010.

Fallah, N., Gu, H., Mohammad, K., Seyyedsalehi, S. A., Nourijelyani, K., and Eshraghian, M. R. Nonlinear poisson regression using neural networks: a simulation study. *Neural Computing and Applications*, 18(8):939, 2009.

Fergus, R., Perona, P., and Zisserman, A. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 264–271, 2003.

Fernández, M. and Williams, S. Closed-form expression for the Poisson-binomial probability density function. *IEEE Transactions on Aerospace and Electronic Systems*, 46(2):803–817, 2010.

Fieraru, M., Khoreva, A., Pishchulin, L., and Schiele, B. Learning to refine human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 205–214, 2018.

Frénay, B. and Verleysen, M. Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014.

Gail, M. H., Lubin, J. H., and Rubinstein, L. V. Likelihood calculations for matched case-control studies and survival studies with tied death times. *Biometrika*, 68(3):703–707, 1981.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 1050–1059, 2016.

Gan, C., Wang, N., Yang, Y., Yeung, D.-Y., and Hauptmann, A. G. Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2568–2577. IEEE, 2015.

Gao, M., Li, A., Yu, R., Morariu, V. I., and Davis, L. S. C-WSL: Count-guided weakly supervised localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 152–168, 2018.

Geiger, A., Moosmann, F., Car, Ö., and Schuster, B. Automatic camera and range sensor calibration using a single shot. In *2012 IEEE International Conference on Robotics and Automation*, pp. 3936–3943. IEEE, 2012.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Gillet, O. and Richard, G. ENST-drums: an extensive audio-visual database for drum signals processing. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, pp. 156–159, 2006.

Girshick, R. Fast r-cnn. In *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

Giuffrida, M. V., Minervini, M., and Tsaftaris, S. A. Learning to count leaves in rosette plants. In *Workshop on Computer Vision Problems in Plant Phenotyping (CVPPP)*. BMVA press, 2016.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning.* MIT Press, 2016.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 529–536, 2005.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 369–376. ACM, 2006.

Graving, J. M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B. R., and Couzin, I. D. DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife*, 8:e47994, 2019.

Halevy, A., Norvig, P., and Pereira, F. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 8536–8546, 2018.

Hawthorne, C., Elsen, E., Song, J., Roberts, A., Simon, I., Raffel, C., Engel, J., Oore, S., and Eck, D. Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*, 2017.

Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, A., Dieleman, S., Elsen, E., Engel, J., and Eck, D. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. IEEE, 2016.

Henderson, P. and Ferrari, V. End-to-end training of object class detectors for mean average precision. In *Asian Conference on Computer Vision (ACCV)*, pp. 198–213. Springer, 2016.

Hilbert, D. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Mathematische Annalen*, 38:459–460, 1891.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hosang, J., Benenson, R., and Schiele, B. Learning non-maximum suppression. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4507–4515, 2017.

Howard, S. Discussion on Professor Cox's paper. *Journal of the Royal Statistical Society*, 34B(2):210–211, 1972.

Hsieh, M.-R., Lin, Y.-L., and Hsu, W. H. Drone-based object counting by spatially regularized regional proposal network. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Hu, D., DeTone, D., and Malisiewicz, T. Deep ChArUco: Dark ChArUco marker pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8436–8444, 2019.

Hu, Y., Chang, H., Nian, F., Wang, Y., and Li, T. Dense crowd counting from still images with convolutional neural networks. *Journal of Visual Communication and Image Representation*, 38:530–539, 2016.

Huang, C., Li, Y., Change Loy, C., and Tang, X. Learning deep representation for imbalanced classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5375–5384, 2016a.

Huang, D.-A., Fei-Fei, L., and Niebles, J. C. Connectionist temporal modeling for weakly supervised action labeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 137–153. Springer, 2016b.

Huang, R., Xu, B., Schuurmans, D., and Szepesvári, C. Learning with a strong adversary. *arXiv preprint arXiv:1511.03034*, 2015.

Huang, S., Li, X., Zhang, Z., Wu, F., Gao, S., Ji, R., and Han, J. Body structure aware deep crowd counting. *IEEE Transactions on Image Processing*, 27(3): 1049–1059, 2017.

Huber, P. J. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pp. 73–101, 1964.

Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., and Shah, M. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–546, 2018.

Jang, E., Gu, S., and Poole, B. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*, 2017.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 2309–2318, 2018.

Jiang, L., Zhang, J., and Deng, B. Robust rgb-d face recognition using attribute-aware loss. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Joachims, T. Transductive learning via spectral graph partitioning. In *Proceedings of International Conference on Machine Learning (ICML)*, 2003.

Kaufman, E. L., Lord, M. W., Reese, T. W., and Volkmann, J. The discrimination of visual number. *The American journal of psychology*, 62(4):498–525, 1949.

Kelz, R., Dorfer, M., Korzeniowski, F., Böck, S., Arzt, A., and Widmer, G. On the potential of simple framewise approaches to piano transcription. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2016.

Khan, A., Gould, S., and Salzmann, M. Deep convolutional neural networks for human embryonic cell counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 339–348. Springer, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 27, pp. 3581–3589, 2014.

Kong, D., Gray, D., and Tao, H. A viewpoint invariant approach for crowd counting. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pp. 1187–1190. IEEE, 2006.

Kong, Q., Xu, Y., Wang, W., and Plumbley, M. D. A joint detection-classification model for audio tagging of weakly labelled data. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 641–645. IEEE, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105, 2012.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Kumar, A. and Raj, B. Audio event detection using weakly labeled data. In *Proceedings of International Conference on Multimedia*, pp. 1038–1047, 2016.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., and Ferrari, V. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

Laradji, I. H., Rostamzadeh, N., Pinheiro, P. O., Vazquez, D., and Schmidt, M. Where are the blobs: Counting by localization with point supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 547–562, 2018.

Le Cam, L. An approximation theorem for the Poisson binomial distribution. *Pacific Journal of Mathematics*, 10(4):1181–1197, 1960.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. Temporal convolutional networks for action segmentation and detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 156–165. IEEE, 2017.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten ZIP code recognition. *Neural computation*, 1(4):541–551, 1989.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436, 2015.

Lee, D., Lee, S., Han, Y., and Lee, K. Ensemble of convolutional neural networks for weakly-supervised sound event detection using multiple scale input. *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

Lee, D.-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013.

Lempitsky, V. and Zisserman, A. Learning to count objects in images. In *Advances in neural information processing systems*, pp. 1324–1332, 2010.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Li, J., Su, W., and Wang, Z. Simple pose: Rethinking and improving a bottom-up approach for multi-person pose estimation. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 740–755. Springer, 2014.

Liu, J.-Y. and Yang, Y.-H. Event localization in music auto-tagging. In *Proceedings of International Conference on Multimedia*, pp. 1048–1057. ACM, 2016.

Liu, T. and Tao, D. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2016.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 21–37, 2016.

Liu, X., van de Weijer, J., and Bagdanov, A. D. Leveraging unlabeled data for crowd counting by learning to rank. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015.

Luvizon, D. C., Tabia, H., and Picard, D. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 85: 15–22, 2019.

Ma, Z., Wei, X., Hong, X., and Gong, Y. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6142–6151, 2019.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Maddison, C., Mnih, A., and Teh, Y. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR 2017)*, 2017.

Malaviya, C., Ferreira, P., and Martins, A. F. Sparse and constrained attention for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 370–376, 2018.

Mandler, G. and Shebo, B. J. Subitizing: an analysis of its component processes. *Journal of Experimental Psychology: General*, 111(1):1, 1982.

Martins, A. and Astudillo, R. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pp. 1614–1623, 2016.

Martins, A. F. and Kreutzer, J. Learning what's easy: Fully differentiable neural easy-first taggers. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 349–362, 2017.

Mathe, S., Pirinen, A., and Sminchisescu, C. Reinforcement learning for visual object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2894–2902, 2016.

McNally, W., Vats, K., Pinto, T., Dulhanty, C., McPhee, J., and Wong, A. GolfDB: A video database for golf swing sequencing. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019.

Merget, D., Rock, M., and Rigoll, G. Robust facial landmark detection via a fully-convolutional local-global context network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 781–790, 2018.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Mnih, V. and Hinton, G. E. Learning to label aerial images from noisy data. In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 567–574, 2012.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, 2016.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. Universal adversarial perturbations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1765–1773, 2017.

Moranduzzo, T. and Melgani, F. Automatic car counting method for unmanned aerial vehicle images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(3):1635–1647, 2013.

Mundhenk, T. N., Konjevod, G., Sakla, W. A., and Boakye, K. A large contextual dataset for classification, detection and counting of cars with deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 785–800. Springer, 2016.

Nadaraya, E. A. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

Narayan, S., Cholakkal, H., Khan, F. S., and Shao, L. 3c-net: Category count and center loss for weakly-supervised action localization. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1196–1204, 2013.

Nehme, E., Weiss, L. E., Michaeli, T., and Shechtman, Y. Deep-storm: super-resolution single-molecule microscopy by deep learning. *Optica*, 5(4):458–464, 2018.

Nettleton, D. F., Orriols-Puig, A., and Fornells, A. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275–306, 2010.

Neumann, L. and Vedaldi, A. Tiny people pose. In *Asian Conference on Computer Vision (ACCV)*, pp. 558–574. Springer, 2018.

Newell, A., Yang, K., and Deng, J. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 483–499. Springer, 2016.

Nguyen, M. H., Torresani, L., De La Torre, F., and Rother, C. Weakly supervised discriminative localization and classification: a joint learning process. In *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 1925–1932. IEEE, 2009.

Nguyen, P., Liu, T., Prasad, G., and Han, B. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6752–6761. IEEE, 2018.

Nibali, A., He, Z., Morgan, S., and Prendergast, L. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018.

Niebles, J. C., Wang, H., and Fei-Fei, L. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

Niebles, J. C., Chen, C.-W., and Fei-Fei, L. Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 392–405. Springer, 2010.

Noroozi, M., Pirsiavash, H., and Favaro, P. Representation learning by learning to count. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Oliver, A., Odena, A., Raffel, C. A., Cubuk, E. D., and Goodfellow, I. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3235–3246, 2018.

Onoro-Rubio, D. and López-Sastre, R. J. Towards perspective-free object counting with deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 615–629. Springer, 2016.

Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., and Murphy, K. Towards accurate multi-person pose estimation in the wild. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4903–4911, 2017.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE, 2016.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1944–1952, 2017.

Paul Cohen, J., Boucher, G., Glastonbury, C. A., Lo, H. Z., and Bengio, Y. Count-ception: Counting by fully convolutional redundant counting. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

Peano, G. Sur une courbe, qui remplit toute une aire plane. *Mathematische Annalen*, 36(1):157–160, 1890.

Petrov, V. V. On lower bounds for tail probabilities. *Journal of Statistical Planning and Inference*, 137(8):2703–2705, 2007.

Pfister, T., Charles, J., and Zisserman, A. Flowing convnets for human pose estimation in videos. In *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 1913–1921, 2015.

Placht, S., Fürsattel, P., Mengue, E. A., Hofmann, H., Schaller, C., Balda, M., and Angelopoulou, E. ROCHADE: Robust checkerboard advanced detection for camera calibration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 766–779. Springer, 2014.

Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and Ellis, D. P. `mir_eval`: A transparent implementation of common MIR metrics, 2014.

Rahnemoonfar, M. and Sheppard, C. Deep count: fruit counting based on deep simulated learning. *Sensors*, 17(4):905, 2017.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., and Moy, L. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr): 1297–1322, 2010.

Redmon, J. and Farhadi, A. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Rezatofighi, S. H., Milan, A., Abbasnejad, E., Dick, A., Reid, I., et al. Deepsetnet: Predicting sets with deep neural networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5257–5266. IEEE, 2017.

Richard, A., Kuehne, H., and Gall, J. Weakly supervised action learning with RNN based fine-to-coarse modeling. In *Proceedings of IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR), volume 1, pp. 754–763. IEEE, 2017.

Riggs, K. J., Ferrand, L., Lancelin, D., Fryziel, L., Dumur, G., and Simpson, A. Subitizing in tactile perception. *Psychological Science*, 2006.

Roos, B. Binomial approximation to the Poisson binomial distribution: The Krawtchouk expansion. *Theory of Probability & Its Applications*, 45(2):258–272, 2001.

Ryan, D., Denman, S., Fookes, C., and Sridharan, S. Crowd counting using multiple local features. In *2009 Digital Image Computing: Techniques and Applications*, pp. 81–88. IEEE, 2009.

Sage, D., Kirshner, H., Pengo, T., Stuurman, N., Min, J., Manley, S., and Unser, M. Quantitative evaluation of software packages for single-molecule localization microscopy. *Nature methods*, 12(8):717–724, 2015.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 29, pp. 1163–1171, 2016.

Saleheen, N., Ali, A. A., Hossain, S. M., Sarker, H., Chatterjee, S., Marlin, B., Ertin, E., Al'Absi, M., and Kumar, S. puffmarker: a multi-sensor approach for pinpointing the timing of first lapse in smoking cessation. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 999–1010. ACM, 2015.

Sam, D. B. and Babu, R. V. Top-down feedback for crowd counting convolutional neural network. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Scaramuzza, D., Martinelli, A., and Siegwart, R. A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5695–5701. IEEE, 2006.

Schlüter, J. and Böck, S. Improved musical onset detection with convolutional neural networks. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6979–6983. IEEE, 2014.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

Schroeter, J., Sidorov, K., and Marshall, D. Weakly-supervised temporal localization via occurrence count learning. In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 5649–5659, 2019.

Schroeter, J., Sidorov, K., and Marshall, D. Robust temporal point event localization through smoothing and counting. In *ICML Workshop on Uncertainty & Robustness in Deep Learning (UDL)*, 2020a.

Schroeter, J., Tuytelaars, T., Sidorov, K., and Marshall, D. Learning multi-instance sub-pixel point localization. In *Asian Conference on Computer Vision (ACCV)*, 2020b.

Schroeter, J., Sidorov, K., and Marshall, D. Learning precise temporal point event detection with misaligned labels. In *AAAI*, 2021.

Seguí, S., Pujol, O., and Vitria, J. Learning to count with deep object features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 90–96, 2015.

Shah, B. *American Statistician*, 27(3):123–124, 1973.

Shang, C., Ai, H., and Bai, B. End-to-end crowd counting via joint learning local and global count. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 1215–1219. IEEE, 2016.

Shou, Z., Gao, H., Zhang, L., Miyazawa, K., and Chang, S.-F. AutoLoc: Weakly supervised temporal action localization in untrimmed videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 154–171, 2018.

Sigtia, S., Benetos, E., and Dixon, S. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, 2016.

Sigurdsson, G. A., Russakovsky, O., and Gupta, A. What actions are needed for understanding human actions in videos? In *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 2137–2146, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Singh, K. K. and Lee, Y. J. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 3544–3553. IEEE, 2017.

Sinzinger, E. D. A model-based approach to junction detection using radial energy. *Pattern Recognition*, 41(2):494–505, 2008.

Song, Y., Schwing, A., Urtasun, R., et al. Training deep neural networks via direct loss minimization. In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 2169–2177, 2016.

Song, Z. and Qiu, Q. Learn to classify and count: A unified framework for object classification and counting. In *Proceedings of the 2018 International Conference on Image and Graphics Processing*, pp. 110–114, 2018.

Southall, C., Stables, R., and Hockman, J. Automatic drum transcription using bi-directional recurrent neural networks. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, pp. 591–597, 2016.

Southall, C., Stables, R., and Hockman, J. Automatic drum transcription for polyphonic recordings using soft attention mechanisms and convolutional neural networks. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, pp. 606–612, 2017.

Stevens, S. S., Volkmann, J., and Newman, E. B. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

Storn, R. and Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11 (4):341–359, 1997.

Stöter, F.-R., Chakrabarty, S., Edler, B., and Habets, E. A. Classification vs. regression in supervised learning for single channel speaker count estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 436–440. IEEE, 2018.

Su, J., Vargas, D. V., and Sakurai, K. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

Sun, X., Xiao, B., Wei, F., Liang, S., and Wei, Y. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 529–545, 2018.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3104–3112, 2014.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tai, Y., Liang, Y., Liu, X., Duan, L., Li, J., Wang, C., Huang, F., and Chen, Y. Towards highly accurate and stable face alignment for high-resolution videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8893–8900, 2019.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1195–1204, 2017.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., and Bregler, C. Efficient object localization using convolutional networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 648–656, 2015.

Tompson, J. J., Jain, A., LeCun, Y., and Bregler, C. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1799–1807, 2014.

Toshev, A. and Szegedy, C. DeepPose: Human pose estimation via deep neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1653–1660, 2014.

Trott, A., Xiong, C., and Socher, R. Interpretable counting for visual question answering. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

Vahdat, A. Toward robustness against label noise in training deep discriminative neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 5596–5605, 2017.

Vogl, R., Dorfer, M., and Knees, P. Recurrent neural networks for drum transcription. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, pp. 730–736, 2016.

Vogl, R., Dorfer, M., and Knees, P. Drum transcription from polyphonic music with recurrent neural networks. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 201–205, 2017.

Walach, E. and Wolf, L. Learning to count with cnn boosting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 660–676. Springer, 2016.

Wan, J. and Chan, A. Adaptive density map generation for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1130–1139, 2019.

Wan, L., Eigen, D., and Fergus, R. End-to-end integration of a convolution network, deformable parts model and non-maximum suppression. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 851–859, 2015.

Wang, L., Xiong, Y., Lin, D., and Van Gool, L. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4325–4334, 2017.

Wang, Y., Liu, W., Ma, X., Bailey, J., Zha, H., Song, L., and Xia, S.-T. Iterative learning with open-set noisy labels. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8688–8696. IEEE, 2018.

Watson, G. S. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372, 1964.

Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. Convolutional pose machines. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4732, 2016.

Wolfram Alpha. URL `https://www.wolframalpha.com/`.

Wu, C.-W., Dittmar, C., Southall, C., Vogl, R., Widmer, G., Hockman, J., Muller, M., and Lerch, A. A review of automatic drum transcription. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 26(9):1457–1483, 2018.

Xiao, B., Wu, H., and Wei, Y. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 466–481, 2018.

Xie, W., Noble, J. A., and Zisserman, A. Microscopy cell counting and detection with fully convolutional regression networks. *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, 6(3):283–292, 2018.

Xu, Y., Kong, Q., Huang, Q., Wang, W., and Plumbley, M. D. Attention and localization based on a deep convolutional recurrent model for weakly supervised audio tagging. *arXiv preprint arXiv:1703.06052*, 2017.

Xu, Y., Kong, Q., Wang, W., and Plumbley, M. D. Large-scale weakly supervised audio classification using gated convolutional neural network. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 121–125. IEEE, 2018.

Yadati, K., Larson, M., Liem, C. C., and Hanjalic, A. Detecting socially significant music events using temporally noisy labels. *IEEE Transactions on Multimedia*, 20(9):2526–2540, 2018.

Yang, J., Liu, Q., and Zhang, K. Stacked hourglass network for robust facial landmark localisation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 79–87, 2017.

Zhang, C., Li, H., Wang, X., and Yang, X. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015a.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2017a.

Zhang, F., Zhu, X., Dai, H., Ye, M., and Zhu, C. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7093–7102, 2020.

Zhang, J., Ma, S., Sameki, M., Sclaroff, S., Betke, M., Lin, Z., Shen, X., Price, B., and Mech, R. Salient object subitizing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4045–4054, 2015b.

Zhang, S., Wu, G., Costeira, J. P., and Moura, J. M. F. Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017b.

Zhang, Y., Zhou, D., Chen, S., Gao, S., and Ma, Y. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 589–597, 2016.

Zhao, Z., Li, H., Zhao, R., and Wang, X. Crossing-line crowd counting with two-phase deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 712–726. Springer, 2016.

Zhou, X., Wang, D., and Krähenbühl, P. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

Zhu, X., Ghahramani, Z., and Lafferty, J. D. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of International Conference on Machine Learning (ICML)*, 2003.

Appendix A

Local Minima Proof of Chapter 3

The identification of the local minima of the loss function with respect to the instance probabilities $\mathbf{p}=\{p_1,\ldots,p_N\}\in[0,1]^N$ plays an important role in understanding how well a gradient-based optimization converges towards one of the global optima. In this regard, Theorem 3.2 is key to proving that optimizing the *non-convex* Poisson-binomial loss function leads to the sparsity of the individual instances since it shows that the only local minima of the loss function are the global minima themselves, recall

Theorem (Local Minima). Let $l(\mathbf{x}) := D_{KL}(\mathbb{1}_c \| \sum_i \mathcal{B}(x_i))$, then $\forall c \le N$

$$\Big\{\mathbf{p} = \{p_1,\ldots,p_N\} \in [0,1]^N \mid \mathbf{p} \text{ is a local minimum of } l(\mathbf{x})\Big\} \\ \equiv \Big\{\mathbf{p} = \{p_1,\ldots,p_N\} \in [0,1]^N \mid l(\mathbf{p}) = \underbrace{D_{KL}(\mathbb{1}_c \| \textstyle\sum_i \mathcal{B}(p_i)) = 0}_{\text{Global Minimum}}\Big\}. \tag{A.1}$$

This chapter presents a comprehensive proof of this important theorem. Overall, since any global minimum is per definition also a local minimum, it only remains to show that all local minima of the Poisson-binomial loss function are in fact global minima in order to prove the full equivalence.

In order to simplify notation, we investigate throughout this chapter the local maxima of the function

$$h(\mathbf{p}) := \Pr\Big(\textstyle\sum_i \mathcal{B}(p_i) = c\Big) \tag{A.2}$$

whose local maxima are identical to the local minima of the loss function $l(\mathbf{p})$ (see Equation 2.9).

Definition of Local Minima

As an introduction, it is useful to formally define what a local minimum of a function is:

Definition A.1 (Local Minimum)**.** In the Euclidean space, $\mathbf{x}$ is a local minimum of the function $f : X \to \mathbb{R}$ if $\exists \epsilon > 0$ such that $\forall \tilde{\mathbf{x}} \in X$,

$$d(x_i, \tilde{x}_i) = \sqrt{\textstyle\sum_i (x_i - \tilde{x}_i)^2} < \epsilon \implies f(\mathbf{x}) \leq f(\tilde{\mathbf{x}}) \tag{A.3}$$

The definition of local maximality is analogous. In a nutshell, a point $\mathbf{p}$ is a local maximum of the function $h(x)$ if there exists a neighborhood Ω around p such that, for any $\tilde{p} \in \Omega$, the value at that point is smaller than $h(\mathbf{p})$ (i.e., $h(\mathbf{p}) \geq h(\tilde{\mathbf{p}})$).

No Local Minima in $(0,1)^N$

We first demonstrate that $h(\mathbf{p})$ has no local maxima on the interval $(0,1)^N$, and thus prove by extension that the Poisson-binomial loss function $l(\mathbf{p})$ has no local minima on that plane. This can be achieved by proving that the Hessian $\mathbf{H}h(\mathbf{p})$ is not negative-definite at any point $\mathbf{p} \in (0,1)^N$ for all values of $c \in \mathbb{N}$.

The Hessian matrix $\mathbf{H}h(\mathbf{p})$ of $h(\mathbf{p})$ can easily be computed using the distribution of the Poisson-binomial distribution (Property 2.1) and its recursion property (Property 2.4). Indeed, for $c > 1$ (the case $c = 0$ is analogous), the function $h(\mathbf{p})$ is equivalent to

$$\begin{aligned} h(\mathbf{p}) &:= \Pr\Big(\textstyle\sum_i \mathcal{B}(p_i) = c\Big) \\ &\overset{(2.4)}{=} \sum_{S \in F_c} \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j) \\ &\overset{(2.11)}{=} (1 - p_n) \sum_{S \in F_c^{\setminus n}} \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j) + p_n \sum_{S \in F_{c-1}^{\setminus n}} \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j), \end{aligned} \tag{A.4}$$

where $F_c^{\setminus n}$ is the set of all subsets of $P(\{1, \ldots, N\} \setminus \{n\})$ of size c. The last equality holds for any $n \leq N$ since the indices of the instance probabilities can be reordered due to the independence assumption of the Bernoulli distributions.

Thus, the second derivative of $h(\mathbf{p})$ with respect to p_n is equal to zero:

$$\begin{aligned}\frac{\partial^2}{\partial p_n^2} h(\mathbf{p}) &= \frac{\partial^2}{\partial p_n^2}(1-p_n) \sum_{S\in F_c^{\setminus n}} \prod_{i\in S} p_i \prod_{j\in S^c} (1-p_j) + \frac{\partial^2}{\partial p_n^2} p_n \sum_{S\in F_{c-1}^{\setminus n}} \prod_{i\in S} p_i \prod_{j\in S^c} (1-p_j) \\ &= -\frac{\partial}{\partial p_n} \sum_{S\in F_c^{\setminus n}} \prod_{i\in S} p_i \prod_{j\in S^c} (1-p_j) + \frac{\partial}{\partial p_n} \sum_{S\in F_{c-1}^{\setminus n}} \prod_{i\in S} p_i \prod_{j\in S^c} (1-p_j) \\ &= 0. \end{aligned} \tag{A.5}$$

This property implies that the diagonal entries of the Hessian matrix are all equal to zero, and thus its trace, which is defined as the sum of the entries on the diagonal, is equal to zero. Therefore, as it is well known that the trace of a matrix is equal to the sum of eigenvalues λ_i, the sum of the eigenvalues of the Hessian is equal to zero, i.e.,

$$\text{trace}(\mathbf{H}h(\mathbf{p})) = \textstyle\sum_n \frac{\partial^2}{\partial p_n^2} h(\mathbf{p}) = 0 = \sum_i \lambda_i \tag{A.6}$$

In addition, since the function $h(\mathbf{p})$ fulfils all conditions of Schwarz's theorem, the Hessian matrix $\mathbf{H}h(\mathbf{p})$ is symmetric, i.e.,

$$\frac{\partial}{\partial p_i} \frac{\partial}{\partial p_j} h(\mathbf{p}) = \frac{\partial}{\partial p_j} \frac{\partial}{\partial p_i} h(\mathbf{p}). \tag{A.7}$$

From linear algebra, it thus follows that the matrix $\mathbf{H}h(\mathbf{p})$ has exactly N real-valued eigenvalues. By combining this result with Equation A.6, it can be concluded that either the eigenvalues of the Hessian $\mathbf{H}h(\mathbf{p})$ are all equal to zero (in fact only possible for $N = 1$) or that some eigenvalues are negative and some others are positive. Consequently, the Hessian $\mathbf{H}h(\mathbf{p})$ is certainly not negative-definite (i.e., all eigenvalues are not all strictly negative) at any point $\mathbf{p} \in (0, 1)^N$, and thus there are no local minima of the loss function in that interval.

Local Minima are only possible for $p \in \{0, 1\}^N$

As no local minima can be found in the interval $(0, 1)^N$, the only points where local minima might arise lie on the boundaries of the [0,1] interval. It can further be shown that these local minima can only be found at the corners of the hypercube, i.e., $\mathbf{p} \in \{0, 1\}^N$.

This more restricting statement can be proven by contradiction. Indeed, let us assume that a point $\mathbf{p} \notin \{0,1\}^N$ is a local maximum of the function $h(\mathbf{p})$. Then, since $\mathbf{p}$ cannot be in the interval $(0,1)^N$—as shown above—there exists an index $\zeta_1 \leq N$ such that $p_{\zeta_1} \in \{0,1\}$. Before proceeding further, it is worth mentioning that c has to be larger than zero since otherwise the only unique local minimum would trivially be $\mathbf{p} = \mathbf{0}$, which would contradict the claim that $\mathbf{p} \notin \{0,1\}^N$.

Using the recursion properly (Property 2.4) as done above, the instance p_{ζ_1} can be factorized as follow:

$$\begin{aligned} h(\mathbf{p}) &:= \Pr\Big(\textstyle\sum_i \mathcal{B}(p_i) = c \Big) \\ &= \quad (1 - p_{\zeta_1}) \sum_{S \in F_c^{\setminus \zeta_1}} \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j) + p_{\zeta_1} \sum_{S \in F_{c-1}^{\setminus \zeta_1}} \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j). \end{aligned} \tag{A.8}$$

Thus, in the case $p_{\zeta_1} = 0$, the function simplifies to

$$h(\mathbf{p}) = \sum_{S \in F_c^{\setminus \zeta_1}} \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j), \tag{A.9}$$

which correspond to a Poisson-binomial loss function with one less instance probability (*i.e.*, $N - 1$).

The case where $p_{\zeta_1} = 1$ is almost equivalent

$$h(\mathbf{p}) = \sum_{S \in F_{c-1}^{\setminus \zeta_1}} \prod_{i \in S} p_i \prod_{j \in S^c} (1 - p_j), \tag{A.10}$$

except c is decreased by one. In the case where c reaches zero, the assumption that $\mathbf{p} \notin \{0,1\}^N$ is violated and, thus, the statement that the minima of the loss function can only exist in $\{0,1\}^N$ is proven by contradiction.

Consequently, either the assumption is violated (i.e., $c = 0$) or either the resulting function is a lower-dimension Poisson-binomial loss function with one less instance probability. However, the remaining probabilities (i.e., $p_i \neq p_{\zeta_1}$) cannot all be in $(0,1)$ since it was shown above that there cannot be a local minimum of the $(N-1)$-dimensional Poisson-binomial loss function in $(0,1)^{N-1}$. Thus, there has to be an addition index ζ_2 (i.e., $\zeta_2 \neq \zeta_1$) that satisfies $p_{\zeta_2} \in \{0,1\}$. This process repeats itself until all probability instances are in $\{0,1\}$, thus proving by contradiction that $\mathbf{p}\in\{0,1\}^N$ if $\mathbf{p}$ is a local minimum of the loss function. (This claim could be proven more formally by induction.)

A complete proof also requires to consider the case where, during the recursion, the remaining label count to be accounted for (i.e., $c-\sum p_{\zeta_i}$) becomes larger than the number of remaining instances (i.e., $N-|\boldsymbol{\zeta}|$). In this scenario, the function is equal to zero (i.e., $h(x)=0$) regardless of the value assigned to the remaining instance probabilities. Indeed, each instance probability can only increment the predicted count by 1, thus counts higher than the number of instances N are unattainable and have no probability mass assigned to them. Let us denote one of these points as $\mathbf{p}_\kappa$. In order to show that such a point is not local maxima of $h(x)$, let us define the point $\mathbf{p}_\delta := (1-2\delta)\mathbf{p}_\kappa$. In a nutshell, this operation linearly maps all zeros of p_κ to δ and all ones to $1-\delta$. From the definition of $h(x)$, it can easily be observed that $h(\mathbf{p}_\delta)>0$ for any value $0<\delta<1$ since products and sums of strictly positive numbers yield strictly positive results. Thus, for any $0<\delta<1$, we obtain the following inequality:

$$h(\mathbf{p}_\kappa) = 0 < h(\mathbf{p}_\delta) \tag{A.11}$$

Since δ can be selected arbitrarily small, a neighborhood Ω of $\mathbf{p}_\kappa$ such that $h(\mathbf{p}) \geq h(\tilde{p}), \forall \tilde{p} \in \Omega$ cannot be constructed (i.e., one can always find a δ such that $\mathbf{p}_\delta \in \Omega$). Thus, $\mathbf{p}$ cannot be a local maximum of h by definition (see Definition A.1).

Final Identification of Local Minima

So far, we have shown that local minima of the loss function can only be found at the corners of the domain of the loss function (i.e., $\mathbf{p} \overset{!}{\in} \{0,1\}^N$). Thus, it only remains to demonstrate that any point $\mathbf{p} \in \{0,1\}^N$ that does not satisfy the global minimality criterion (Theorem 3.1) is not a local minimum of the loss function.

First, since $\mathbf{p} \in \{0,1\}^N$, it can intuitively be shown, using Property 2.3, that the resulting Poisson-binomial count is a scalar in this case and that it takes integer values, i.e., $\sum_i p_i \in \mathbb{N}$. Thus, the sum $\sum_i p_i$ is either equal to the correct label count c or not. However, any $\mathbf{p}$ that satisfies $\sum_i p_i = c)$ is obviously a global minimum of the loss function since it maximizes the probability assigned to the correct count c., i.e., $h(x) = 1$. Thus, it remains to show that any $\mathbf{p}$ that satisfies $\sum_i p_i \neq c$ is not a local maxima of h.

To that end, let us once again define the point $\mathbf{p}_\delta := (1-2\delta)\mathbf{p}$. In a nutshell, as explained above, given a point $\mathbf{p} \in \{0,1\}^N$, $\mathbf{p}_\delta$ is simply a copy of it with all the zeros mapped to δ and all the ones mapped to $1-\delta$. Recall that from the definition of $h(x)$, it can easily be seen that $h(\mathbf{p}_\delta) > 0$ for any value $0 < \delta < 1$ since products and sums of strictly positive numbers yield strictly positive results. Thus, for any $0<\delta<1$ and any $\mathbf{p}$ that satisfies $\sum_i p_i \neq c$, we obtain the following inequality:

$$h(\mathbf{p}) = \Pr\Big(\sum_i p_i = c \Big) = 0 < h(\mathbf{p}_\delta) \tag{A.12}$$

The same argument presented previously for the point $\mathbf{p}_\kappa$ can be made for $\mathbf{p}$: since δ can be selected arbitrarily small, a neighborhood Ω of $\mathbf{p}$ such that $h(\mathbf{p}) \geq h(\tilde{p}), \forall \tilde{p} \in \Omega$ cannot be constructed (i.e., one can always find an δ such that $\mathbf{p}_\delta \in \Omega$). Thus, $\mathbf{p}$ cannot be a local maximum of h by definition (see Definition A.1), which concludes the proof that any $\mathbf{p}$ that satisfies $\sum_i p_i \neq c$ is not a local maximum of h.

In conclusion, the set of local maxima of $h(\mathbf{p})$ corresponds to the set of global maxima of $h(\mathbf{p})$. By extension, the set of local minima of the loss function is identical to the set of global minima of the loss function.

Appendix B

Convergence Proofs of Example from Chapter 5

B.1 Proof of Example: $\mathcal{L}_{\mathrm{CE}}$

The minimization problem presented in the example of Section 5.4.1 can be simplified using, among others, the definition of $\mathcal{L}_{\mathrm{CE}}$ and $\mathbf{y}^{(i)}$:

$$\begin{aligned}
&\arg\min_\phi \textstyle\sum_i \mathcal{L}_{\mathrm{CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \\
&= \arg\min_\phi \textstyle\sum_i \frac{1}{N}\mathcal{L}_{\mathrm{CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \\
&\overset{N\text{ large}}{\approx} \arg\min_\phi \mathbb{E}_{\epsilon\sim E}[\mathcal{L}_{\mathrm{CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)})] \\
&= -\arg\min_\phi \mathbb{E}_{\epsilon\sim E}[\textstyle\sum_t y_t \log((\phi*\mathbf{x}^{(i)})_t) \\
&\qquad\qquad\qquad + (1-y_t)\log(1-(\phi*\mathbf{x}^{(i)})_t)] \\
&= \arg\max_\phi \mathbb{E}_{\epsilon\sim E}[\textstyle\sum_t y_t \log((\phi*\mathbf{x}^{(i)})_t) \\
&\qquad\qquad\qquad + (1-y_t)\log(1-(\phi*\mathbf{x}^{(i)})_t)] \\
&= \arg\max_\phi \mathbb{E}_{\epsilon^{(i)}\sim E}[\textstyle\sum_t \mathbb{1}_{[t=t^{(i)}+\epsilon^{(i)}]} \log((\phi*\mathbf{x}^{(i)})_t) \\
&\qquad\qquad\qquad + (1-\mathbb{1}_{[t=t^{(i)}+\epsilon^{(i)}]})\log(1-(\phi*\mathbf{x}^{(i)})_t)] \\
&= \arg\max_\phi \mathbb{E}_{\epsilon\sim E}[\log((\phi * \mathbf{x}^{(i)})_{t^{(i)}+\epsilon^{(i)}}) \\
&\qquad\qquad\qquad + \textstyle\sum_{t\neq t^{(i)}+\epsilon^{(i)}} \log(1-(\phi * \mathbf{x}^{(i)})_t] \\
&= \arg\max_\phi \textstyle\sum_k P(E=k)[\log((\phi * \mathbf{x}^{(i)})_{t^{(i)}+k}) \\
&\qquad\qquad\qquad + \textstyle\sum_{t\notin t^{(i)}+k} \log(1-(\phi * \mathbf{x}^{(i)})_t] \\
&= \arg\max_\phi \textstyle\sum_k P(E+t^{(i)}=k)\log((\phi * \mathbf{x}^{(i)})_k) \\
&\qquad\qquad\qquad + (1-P(E+t^{(i)}=k))\log(1-(\phi * \mathbf{x}^{(i)})_k)
\end{aligned} \tag{B.1}$$

Using the definition of $\mathbf{x}^{(i)}$ and of the convolution operation, the expression can be simplified further:

$$\begin{aligned}
&\arg\max_\phi \textstyle\sum_k P(E{+}t^{(i)}{=}k)\log((\phi * \mathbf{x}^{(i)})_k) \\
&\qquad\qquad + (1{-}P(E{+}t^{(i)}{=}k))\log(1{-}(\phi * \mathbf{x}^{(i)})_k) \\
&= \arg\max_\phi \textstyle\sum_k P(E{+}t^{(i)}{=}k)\log(\sum_t \phi(t)\cdot x^{(i)}_{k-t}) \\
&\qquad + (1{-}P(E{+}t^{(i)}{=}k))\log(1{-}\sum_t \phi(t)\cdot x^{(i)}_{k-t} \\
&= \arg\max_\phi \textstyle\sum_k P(E{+}t^{(i)}{=}k)\log(\phi(k{-}t^{(i)})) \\
&\qquad + (1{-}P(E{+}t^{(i)}{=}k))\log(1{-}\phi(k{-}t^{(i)})) \\
&= \arg\max_\phi \textstyle\sum_k P(E={k}{-}t^{(i)})\log(\phi(k{-}t^{(i)})) \\
&\qquad + (1{-}P(E={k}{-}t^{(i)}))\log(1{-}\phi(k{-}t^{(i)})) \\
&= \arg\max_\phi \textstyle\sum_k P(E=k)\log(\phi(k)) \\
&\qquad + (1{-}P(E=k))\log(1{-}\phi(k)).
\end{aligned} \tag{B.2}$$

Since the value of the different timesteps (k) are mutually independent from one another in the optimization problem, each bin of the convolution filter ϕ can be optimized separately, i.e.,

$$\begin{aligned}
&\arg\max_{\phi(k)} P(E=k)\log(\phi(k)) \\
&\qquad\qquad + (1{-}P(E=k))\log(1{-}\phi(k)).
\end{aligned} \tag{B.3}$$

Each individual optimization problem can be expressed as

$$\arg\max_x \alpha\log(x) + (1-\alpha)\log(1-x), \tag{B.4}$$

which has the following closed-form solution:

$$\begin{aligned}
&\frac{\partial}{\partial x}\alpha\log(x) + (1-\alpha)\log(1-x) \overset{!}{=} 0 \\
&\quad\Longleftrightarrow\ \frac{\alpha}{x} - \frac{1-\alpha}{1-x} \overset{!}{=} 0 \\
&\quad\overset{x\neq 0,1}{\Longleftrightarrow}\ \alpha(1-x) \overset{!}{=} (1-\alpha)x \\
&\quad\Longleftrightarrow\ x \overset{!}{=} \alpha.
\end{aligned} \tag{B.5}$$

Thus, combining Equation (B.3) and Equation (B.5), we obtain the result reported in Equation (5.5) from the main text:

$$\begin{aligned} &\phi^* = \arg\min_\phi \textstyle\sum_i \mathcal{L}_{\mathrm{CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \\ &\iff \phi^*(\tau) \approx P(E = \tau), \forall \tau. \end{aligned} \tag{B.6}$$

B.2 Proof of Example: $\mathcal{L}_{\mathrm{LS|CE}}$

The derivation of the result from Equation (5.10) is similar to the one presented in Section B.1; thus, a summarized version of the proof is given instead:

$$\begin{aligned}
&\arg\min_\phi \textstyle\sum_i \mathcal{L}_{\mathrm{LS|CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi) \\
&= \arg\min_\phi \tfrac{1}{N} \textstyle\sum_i \mathcal{L}_{\mathrm{LS|CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi) \\
&\overset{N \text{ large}}{\approx} \arg\min_\phi \mathbb{E}_{\epsilon\sim E}[\mathcal{L}_{\mathrm{LS|CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi)] \\
&= \arg\min_\phi \mathbb{E}_{\epsilon\sim E}[-\textstyle\sum_t \Big((\Phi * \mathbf{y}^{(i)})_t \log((\phi * \mathbf{x}^{(i)})_t) \\
&\qquad\qquad +(1-(\Phi * \mathbf{y}^{(i)})_t \log(1-(\phi * \mathbf{x}^{(i)})_t)\Big)] \\
&= \arg\max_\phi \mathbb{E}_{\epsilon\sim E}[\textstyle\sum_t \Big((\Phi * \mathbf{y}^{(i)})_t \log((\phi * \mathbf{x}^{(i)})_t) \\
&\qquad\qquad +(1-(\Phi * \mathbf{y}^{(i)})_t \log(1-(\phi * \mathbf{x}^{(i)})_t)\Big)] \\
&= \arg\max_\phi \mathbb{E}_{\epsilon\sim E}[\textstyle\sum_t \Big(\sum_{\tau=0}^{T} y_{i,\tau} \Phi(t-\tau) \log((\phi * \mathbf{x}^{(i)})_t) \\
&\qquad\qquad +(1-\textstyle\sum_{\tau=0}^{T} y_{i,\tau} \Phi(t-\tau)) \log(1-(\phi * \mathbf{x}^{(i)})_t)\Big)] \\
&= \arg\max_\phi \mathbb{E}_{\epsilon\sim E}[\textstyle\sum_t \Big(\Phi(t-t^{(i)} - \epsilon^{(i)}) \log((\phi * \mathbf{x}^{(i)})_t) \\
&\qquad\qquad +(1-\Phi(t-t^{(i)} - \epsilon^{(i)})) \log(1-(\phi * \mathbf{x}^{(i)})_t)\Big)] \\
&= \arg\max_\phi \textstyle\sum_k P(E = k) \sum_t \Big(\Phi(t-t^{(i)} - \epsilon^{(i)}) \log((\phi * \mathbf{x}^{(i)})_t) \\
&\qquad\qquad +(1-\Phi(t-t^{(i)} - \epsilon^{(i)})) \log(1-(\phi * \mathbf{x}^{(i)})_t)\Big) \\
&= \arg\max_\phi \textstyle\sum_t \sum_k P(E = k) \Big(\Phi(t-t^{(i)}-\epsilon^{(i)}) \log((\phi * \mathbf{x}^{(i)})_t) \\
&\qquad\qquad +(1-\Phi(t-t^{(i)} - \epsilon^{(i)})) \log(1-(\phi * \mathbf{x}^{(i)})_t)\Big)] \\
&= \arg\max_\phi \textstyle\sum_t (E * \Phi)_{t-t^{(i)}} \log((\phi * \mathbf{x}^{(i)})_t) \\
&\qquad\qquad +(E * (1-\Phi))_{t-t^{(i)}} \log(1-(\phi * \mathbf{x}^{(i)})_t) \\
&= \arg\max_\phi \textstyle\sum_t (E * \Phi)_{t-t^{(i)}} \log((\phi * \mathbf{x}^{(i)})_t) \\
&\qquad\qquad +(1-(E * \Phi)_{t-t^{(i)}}) \log(1-\hat{p}_{\theta,t})
\end{aligned} \tag{B.7}$$

Thus, using Equation (B.5) and the same argument as in Section B.1, we obtain the final result (Equation (5.9) in the main text):

$$\begin{aligned}
&\phi^* = \arg\min_\phi \textstyle\sum_i \mathcal{L}_{\text{LS|CE}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi) \\
&\iff \phi^*(\tau) \approx (E * \Phi)_\tau = \textstyle\sum_i P(E = i)\Phi(\tau - i), \forall \tau.
\end{aligned} \tag{B.8}$$

B.3 Proof of Example: $\mathcal{L}_{\text{SLL}}$

The beginning of the derivation is done as in previous sections:

$$\begin{aligned}
&\arg\min_\phi \textstyle\sum_i \mathcal{L}_{\text{SLL}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi, \mathcal{E}) \\
&= \arg\min_\phi \tfrac{1}{N} \textstyle\sum_i \mathcal{L}_{\text{SLL}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi, \mathcal{E}) \\
&\overset{N \text{ large}}{\approx} \arg\min_\phi \mathbb{E}_{\epsilon \sim E}[\mathcal{L}_{\text{SLL}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi, \mathcal{E})] \\
&= \arg\min_\phi \mathbb{E}_{\epsilon \sim E}[\textstyle\sum_t \Big((\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t - (\Phi * \mathbf{y}^{(i)})_t\Big)^2] \\
&= \arg\min_\phi \mathbb{E}_{\epsilon \sim E}[\textstyle\sum_t (\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t^2 \\
&\qquad\qquad +(\Phi * \mathbf{y}^{(i)})_t^2 \\
&\qquad\qquad -2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot (\Phi * \mathbf{y}^{(i)})_t] \\
&= \arg\min_\phi \mathbb{E}_{\epsilon^{(i)} \sim E}[\textstyle\sum_t (\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t^2 \\
&\qquad\qquad +\Phi(t - t^{(i)} - \epsilon^{(i)})^2 \\
&\qquad\qquad -2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot \Phi(t - t^{(i)} - \epsilon^{(i)})] \\
&= \arg\min_\phi \textstyle\sum_t (\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t^2 \\
&\qquad +\mathbb{E}_{\epsilon^{(i)} \sim E}[\textstyle\sum_t \Phi(t - t^{(i)} - \epsilon^{(i)})^2 \\
&\qquad\qquad -2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot \Phi(t - t^{(i)} - \epsilon^{(i)})] \\
&= \arg\min_\phi \textstyle\sum_t (\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t^2 \\
&\qquad +\textstyle\sum_k P(E = k)[\textstyle\sum_t \Phi(t - t^{(i)} - k)^2 \\
&\qquad\qquad -2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot \Phi(t - t^{(i)} - k)] \\
&= \arg\min_\phi \textstyle\sum_t (\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t^2 \\
&\qquad +\textstyle\sum_{k,t} P(E = k)\Phi(t - t^{(i)} - k)^2 \\
&\qquad -2\textstyle\sum_{k,t} P(E = k)(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot \Phi(t - t^{(i)} - k) \\
&= \arg\min_\phi \textstyle\sum_t (\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t^2 \\
&\qquad\quad +\textstyle\sum_t (\Phi^2 * E)_{t - t^{(i)}} \\
&\qquad\quad -\textstyle\sum_t 2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot (\Phi * E)_{t - t^{(i)}}
\end{aligned} \tag{B.9}$$

The second term does not depend on ϕ, thus

$$\begin{aligned}
&\arg\min_\phi \textstyle\sum_i \mathcal{L}_{\text{SLL}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi, \mathcal{E}) \\
&\approx\ \arg\min_\phi \textstyle\sum_t (\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t^2 \\
&\qquad\qquad -2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot (\Phi * E)_{t-t^{(i)}}
\end{aligned} \tag{B.10}$$

Differentiating by ϕ yields,

$$\begin{aligned}
&\frac{\partial}{\partial\phi} \textstyle\sum_t (\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t^2 - 2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot (\Phi * E)_{t-t^{(i)}} \\
&\quad= \textstyle\sum_t 2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot (\mathcal{E} * \Phi * 1 * \mathbf{x}^{(i)})_t \\
&\qquad\qquad -2(\mathcal{E} * \Phi * 1 * \mathbf{x}^{(i)})_t \cdot (\Phi * E)_{t-t^{(i)}} \\
&\quad\overset{1*\Phi=1}{=}\ \textstyle\sum_t 2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot (\mathcal{E} * 1 * \mathbf{x}^{(i)})_t \\
&\qquad\qquad -2(\mathcal{E} * 1 * \mathbf{x}^{(i)})_t \cdot (\Phi * E)_{t-t^{(i)}} \\
&\quad\overset{1*\mathcal{E}=1}{=}\ \textstyle\sum_t 2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t \cdot (1 * \mathbf{x}^{(i)})_t \\
&\qquad\qquad -2(1 * \mathbf{x}^{(i)})_t \cdot (\Phi * E)_{t-t^{(i)}} \\
&\quad\overset{1*\mathcal{E}=1}{=}\ \textstyle\sum_t 2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t - 2(\Phi * E)_{t-t^{(i)}}
\end{aligned} \tag{B.11}$$

Using the definition of $\mathbf{x}^{(i)}$

$$\begin{aligned}
&\textstyle\sum_t 2(\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t - 2(\Phi * E)_{t-t^{(i)}} = 0 \\
&\quad\iff \textstyle\sum_t (\mathcal{E} * \Phi * \phi * \mathbf{x}^{(i)})_t - (\Phi * E)_{t-t^{(i)}} = 0 \\
&\quad\iff \textstyle\sum_t (\mathcal{E} * \Phi * \mathbf{p}^{*\,(i)})_t - (\Phi * E)_{t-t^{(i)}} = 0 \\
&\quad\iff \textstyle\sum_t (\mathcal{E} * \mathbf{p}^{*\,(i)})_t - (E)_{t-t^{(i)}} = 0 \\
&\quad\iff \textstyle\sum_t (\mathcal{E} * \mathbf{p}^{*\,(i)})_t - (E*\mathbf{g}^{(i)})_t = 0
\end{aligned} \tag{B.12}$$

Thus, we obtain the final result presented in Equation (5.13), which states that the optimal prediction $\mathbf{p}^{*\,(i)}$ that minimizes the loss $\sum_i \mathcal{L}_{\mathbf{SLL}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \,|\, \Phi, \mathcal{E})$ has the form:

$$(\mathcal{E}*\mathbf{p}^{*\,(i)})_\tau \approx (E*\mathbf{g}^{(i)})_\tau \tag{B.13}$$

B.4 Proof of Example: $\mathcal{L}_{\mathcal{S}\mathbf{oftLoc}}$

As done previously,

$$\begin{aligned} &\arg\min_\phi \textstyle\sum_i \mathcal{L}_{\mathcal{S}\text{oftLoc}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \\ &\quad = \arg\min_\phi \tfrac{1}{N} \textstyle\sum_i \mathcal{L}_{\mathcal{S}\text{oftLoc}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \\ &\quad \overset{N\text{ large}}{\approx} \arg\min_\phi \mathbb{E}_{\epsilon\sim E}[\mathcal{L}_{\mathcal{S}\text{oftLoc}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)})] \end{aligned} \tag{B.14}$$

As the training progresses (i.e., $\alpha_\tau \to 1$), the counting-based constraint becomes more predominant and ensures that only one timestep is assigned all the mass for the event of interest. Thus, given the definition of $\mathbf{x}^{(i)}$ and ϕ, the smoothed prediction is of the form

$$(\Phi * \phi * \mathbf{x}^{(i)})_t = \Phi(t - t^{(i)} - \beta), \tag{B.15}$$

where $\beta \in \mathbb{N}$ is a model constant. Thus, as the counting-based constraint already ensure that exactly one timestep has probability 1 (i.e., $t^{(i)}+\beta$), while all other are assigned zero probability, it simply remains to show that the model bias β is equal to zero. Indeed, it would apply that the predictions are perfectly aligned with the ground-truth without any bias.

Plugging Equation (B.14) into Equation (B.15),

$$\begin{aligned} &\arg\min_\phi \textstyle\sum_i \mathcal{L}_{\mathcal{S}\text{oftLoc}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \\ &\overset{\alpha_\tau\to 1}{\approx} \arg\min_\beta \mathbb{E}_{\epsilon\sim E}[\textstyle\sum_t \Big(\Phi(t-t^{(i)}-\beta)-\Phi(t-t^{(i)}-\epsilon)\Big)^2] \\ &= \arg\min_\beta \mathbb{E}_{\epsilon\sim E}[\textstyle\sum_t \Big(\Phi(t-\beta)-\Phi(t-\epsilon)\Big)^2] \end{aligned} \tag{B.16}$$

Using the properties of the filter Φ, it can be shown that the sum inside the expectation of Equation (B.16) is only a function of the distance between the prediction and the misaligned label (i.e., $|\epsilon-\beta|$):

$$\begin{aligned} \textstyle\sum_t \Big(\Phi(t-\beta)-\Phi(t-\epsilon)\Big)^2 &= \begin{cases} \sum_{\bar{t}} \Big(\Phi(\bar{t})-\Phi(t-(\epsilon-\beta))\Big)^2 \\ \sum_{\bar{t}} \Big(\Phi(\bar{t}-(\beta-\epsilon))-\Phi(t)\Big)^2 \end{cases} \\ \implies \textstyle\sum_t \Big(\Phi(t-\beta)-\Phi(t-\epsilon)\Big)^2 &= \textstyle\sum_t \Big(\Phi(t)-\Phi(t-|\epsilon-\beta|)\Big)^2. \end{aligned} \tag{B.17}$$

Thus, by setting

$$\gamma(x) := \textstyle\sum_t \Big(\Phi(t)-\Phi(t-|x|)\Big)^2, \tag{B.18}$$

Equation (B.16) becomes

$$\begin{aligned}
&\arg\min_{\phi} \textstyle\sum_i \mathcal{L}_{\mathcal{S}\text{oftLoc}}(\phi * \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \\
&= \arg\min_{\beta} \mathbb{E}_{\epsilon\sim E}[\textstyle\sum_t \Big(\Phi(t-\beta)-\Phi(t-\epsilon)\Big)^2] \\
&= \arg\min_{\beta} \mathbb{E}_{\epsilon\sim E}[\gamma(\epsilon-\beta)] \\
&= \arg\min_{\beta} \textstyle\sum_k P(E{=}k)\cdot\gamma(k-\beta),
\end{aligned} \tag{B.19}$$

where $\gamma(x)$ is a positive and symmetric (around 0) function, which is monotonically increasing in $[0,\infty]$—and thus monotonically decreasing in $[-\infty,0]$ by symmetry.

The main results can thus be proven by showing that

$$\textstyle\sum_k P(E{=}k)\cdot\gamma(k-\beta) \geq \sum_k P(E{=}k)\cdot\gamma(k-0), \forall\beta\in\mathbb{N}. \tag{B.20}$$

Let us assume that β is *odd*, then the sum can be reordered as follows:

$$\begin{aligned}
&\textstyle\sum_k P(E{=}k)\cdot\gamma(k-\beta) \\
&\quad= \textstyle\sum_{t\geq0} \Big(P(E{=}\lceil\beta/2\rceil{+}t)\cdot\gamma(\lceil\beta/2\rceil{+}t{-}\beta) \\
&\qquad\qquad + P(E{=}\lfloor\beta/2\rfloor{-}t)\cdot\gamma(\lfloor\beta/2\rfloor{-}t{-}\beta)\Big)
\end{aligned}. \tag{B.21}$$

Using the fact that $\gamma(x)$ is monotonically increasing in $[0,\infty]$ and monotonically decreasing in $[-\infty,0]$ and that the noise distribution is well-behaved (i.e., $P(E{=}k){\leq}P(E{=}\tilde{k}) \iff |k|\geq|\tilde{k}|$) and symmetric, the following inequality holds:

$$\begin{aligned}
&\textstyle\sum_k P(E{=}k)\cdot\gamma(k-\beta) \\
&\quad= \textstyle\sum_{t\geq0} \Big(P(E{=}\lceil\beta/2\rceil{+}t)\cdot\gamma(\lceil\beta/2\rceil{+}t{-}\beta) \\
&\qquad\qquad + P(E{=}\lfloor\beta/2\rfloor{-}t)\cdot\gamma(\lfloor\beta/2\rfloor{-}t{-}\beta)\Big) \\
&\quad\geq \textstyle\sum_{t\geq0} \Big(P(E{=}\lceil\beta/2\rceil{+}t)\cdot\gamma(\lfloor\beta/2\rfloor{-}t{-}\beta)\Big) \\
&\qquad\qquad + P(E{=}\lfloor\beta/2\rfloor{-}t)\cdot\gamma(\lceil\beta/2\rceil{+}t{-}\beta)
\end{aligned} \tag{B.22}$$

The equation can be simplified further using the definition of the rounding operator:

$$\begin{aligned} &\textstyle\sum_k P(E{=}k)\cdot\gamma(k-\beta) \\ &\quad\geq \textstyle\sum_{t\geq 0}\Big(P(E{=}\lceil\beta/2\rceil{+}t)\cdot\gamma(\lfloor\beta/2\rfloor{-}t{-}\beta)\Big) \\ &\qquad\qquad\qquad + P(E{=}\lfloor\beta/2\rfloor{-}t)\cdot\gamma(\lceil\beta/2\rceil{+}t{-}\beta) \\ &\quad= \textstyle\sum_{t\geq 0}\Big(P(E{=}\lceil\beta/2\rceil{+}t)\cdot\gamma(-\lceil\beta/2\rceil{-}t)\Big) \\ &\qquad\qquad\qquad + P(E{=}\lfloor\beta/2\rfloor{-}t)\cdot\gamma(-\lfloor\beta/2\rfloor{+}t). \end{aligned} \tag{B.23}$$

Finally, the final inequality is obtained by both using the symmetry—around zero—of the function $\gamma(x)$ and performing a reordering of the sum:

$$\begin{aligned} &\textstyle\sum_k P(E{=}k)\cdot\gamma(k-\beta) \\ &\quad\geq \textstyle\sum_{t\geq 0}\Big(P(E{=}\lceil\beta/2\rceil{+}t)\cdot\gamma(-\lceil\beta/2\rceil{-}t)\Big) \\ &\qquad\qquad\qquad + P(E{=}\lfloor\beta/2\rfloor{-}t)\cdot\gamma(-\lfloor\beta/2\rfloor{+}t) \\ &\quad= \textstyle\sum_{t\geq 0}\Big(P(E{=}\lceil\beta/2\rceil{+}t)\cdot\gamma(\lceil\beta/2\rceil{+}t)\Big) \\ &\qquad\qquad\qquad + P(E{=}\lfloor\beta/2\rfloor{-}t)\cdot\gamma(\lfloor\beta/2\rfloor{-}t) \\ &\quad= \textstyle\sum_k P(E{=}k)\cdot\gamma(k). \end{aligned} \tag{B.24}$$

The derivation for β *even* is analogous.

In conclusion, since

$$\textstyle\sum_k P(E{=}k)\cdot\gamma(k-\beta)\geq\sum_k P(E{=}k)\cdot\gamma(k-0), \forall\beta\in\mathbb{N}, \tag{B.25}$$

then the main result

$$\begin{aligned} \beta^* = &\arg\min_\beta \textstyle\sum_{t,k} P(E{=}k)\Big(\Phi(t{-}\beta){-}\Phi(t{-}k)\Big)^2 \\ &\implies \beta^* = 0 \end{aligned} \tag{B.26}$$

follows from Equation (B.19). Of course, stronger statements—with weaker assumptions—could be derived if the noise distribution E was explicitly known (e.g., $E = N(0,\sigma^2)$).

Appendix C

Supplement for Chapter 6

C.1 Derivation of the Loss Function and its Gradients

In this section, we present a step-by-step derivation of the closed-form continuous heatmap-matching loss $\mathscr{L}_{\mathrm{HM}}$ proposed in Section 6.3.2 and its partial derivatives. The final equations allow for an straight-forward implementation of the loss function and its gradients; both a Tensorflow and a PyToch implementation are available[1].

C.1.1 Loss Derivation

The definition of the loss function contains improper integrals of squared differences of summations that do not allow for its direct computation:

$$\begin{aligned}
&\mathscr{L}_{\mathrm{HM}}(\mathcal{P}_\theta, \mathcal{L}) \\
&= \iint_{\mathbb{R}^2} D(x_0, y_0 \mid \mathcal{P}_\theta, \mathcal{L}) dx_0 dy_0 \\
&= \iint_{\mathbb{R}^2} \left[S(x_0, y_0 \mid \mathcal{L}) - \hat{S}(x_0, y_0 \mid \mathcal{P}_\theta)\right]^2 dx_0 dy_0 \\
&= \iint_{\mathbb{R}^2} \left[\sum_j \exp\left(-\frac{(x_j - x_0)^2}{\lambda^2} - \frac{(y_j - y_0)^2}{\lambda^2}\right)\right. \\
&\qquad\qquad \left. - \sum_i \hat{p}_i \exp\left(-\frac{(\hat{x}_i - x_0)^2}{\lambda^2} - \frac{(\hat{y}_i - y_0)^2}{\lambda^2}\right)\right]^2 dx_0 dy_0.
\end{aligned} \tag{C.1}$$

[1] https://github.com/SchroeterJulien/ACCV-2020-Subpixel-Point-Localization

Overall, the key idea behind the simplification of this formula is to swap the integrations and summations using Fubini's theorem (or Tonelli's theorem). Afterwards, the loss function can be computed analytically.

> **Distributivity** $(a-b)^2 = a^2 + b^2 - 2ab$

$$\begin{aligned}
&= \iint_{\mathbb{R}^2} \left[\sum_j \exp\left(-\frac{(x_j - x_0)^2 + (y_j - y_0)^2}{\lambda^2}\right)\right]^2 dx_0 dy_0 \\
&\quad + \iint_{\mathbb{R}^2} \left[\sum_i \hat{p}_i \exp\left(-\frac{(\hat{x}_i - x_0)^2 + (\hat{y}_i - y_0)^2}{\lambda^2}\right)\right]^2 dx_0 dy_0 \\
&\quad - 2 \iint_{\mathbb{R}^2} \left[\sum_j \exp\left(-\frac{(x_j - x_0)^2 + (y_j - y_0)^2}{\lambda^2}\right)\right] \\
&\qquad \cdot \left[\sum_i \hat{p}_i \exp\left(-\frac{(\hat{x}_i - x_0)^2 + (\hat{y}_i - y_0)^2}{\lambda^2}\right)\right] dx_0 dy_0
\end{aligned} \tag{C.2}$$

> **Distributivity 2** $\left(\sum_i a_i\right) \cdot \left(\sum_j b_j\right) = \sum_i \sum_j a_i \cdot b_j$

$$\begin{aligned}
&= \iint_{\mathbb{R}^2} \sum_i \sum_j \exp\left(-\frac{(x_i - x_0)^2 + (y_j - y_0)^2 + (x_j - x_0)^2 + (y_j - y_0)^2}{\lambda^2}\right) dx_0 dy_0 \\
&\quad + \iint_{\mathbb{R}^2} \sum_i \sum_j \hat{p}_i \hat{p}_j \exp\left(-\frac{(\hat{x}_i - x_0)^2 + (\hat{y}_i - y_0)^2 + (\hat{x}_j - x_0)^2 + (\hat{y}_j - y_0)^2}{\lambda^2}\right) dx_0 dy_0 \\
&\quad - 2 \iint_{\mathbb{R}^2} \sum_i \sum_j \hat{p}_i \exp\left(-\frac{(x_j - x_0)^2 + (y_j - y_0)^2 + (\hat{x}_i - x_0)^2 + (\hat{y}_i - y_0)^2}{\lambda^2}\right) dx_0 dy_0
\end{aligned} \tag{C.3}$$

> **Fubini's Theorem** If $\sum_i \int |f_i(x) dx| < \infty$ and $\int \sum_i |f_i(x) dx| < \infty$, then $\sum_i \int f_i(x) dx = \int \sum_i f_i(x) dx$

$$
\begin{aligned}
&= \sum_i \sum_j \iint_{\mathbb{R}^2} \exp\left(-\frac{(x_i - x_0)^2 + (y_i - y_0)^2 + (x_j - x_0)^2 + (y_j - y_0)^2}{\lambda^2}\right) dx_0 dy_0 \\
&\quad + \sum_i \sum_j \iint_{\mathbb{R}^2} \hat{p}_i \hat{p}_j \exp\left(-\frac{(\hat{x}_i - x_0)^2 + (\hat{y}_i - y_0)^2 + (\hat{x}_j - x_0)^2 + (\hat{y}_j - y_0)^2}{\lambda^2}\right) dx_0 dy_0 \\
&\quad -2 \sum_i \sum_j \hat{p}_i \iint_{\mathbb{R}^2} \exp\left(-\frac{(x_j - x_0)^2 + (y_j - y_0)^2 + (\hat{x}_i - x_0)^2 + (\hat{y}_i - y_0)^2}{\lambda^2}\right) dx_0 dy_0
\end{aligned}
\tag{C.4}
$$

Integration

$$
\begin{aligned}
&\iint_{\mathbb{R}^2} \exp\left(-\frac{(a - x_0)^2 + (b - y_0)^2 + (c - x_0)^2 + (d - y_0)^2}{\lambda^2}\right) dx_0 dy_0 \\
&\overset{(WolframAlpha)}{=} \int_{\mathbb{R}} -\sqrt{\frac{\pi\lambda^2}{8}} \operatorname{erf}\left(\frac{a + c - 2x_0}{\sqrt{2\lambda^2}}\right) \\
&\qquad \cdot \exp\left(-\frac{a^2 - 2ac + 2b^2 - 4by_0 + c^2 + 2d^2 - 4dy_0 + 4y_0^2}{2\lambda^2}\right) \Bigg|_{x_0=-\infty}^{\infty} dy_0 \\
&= \int_{\mathbb{R}} 2\sqrt{\frac{\pi\lambda^2}{8}} \exp\left(-\frac{a^2 - 2ac + 2b^2 - 4by_0 + c^2 + 2d^2 - 4dy_0 + 4y_0^2}{2\lambda^2}\right) dy_0 \\
&= \sqrt{\frac{\pi\lambda^2}{2}} \exp\left(-\frac{a^2 - 2ac + 2b^2 + c^2 + 2d^2}{2\lambda^2}\right) \int_{\mathbb{R}} \exp\left(-\frac{-4by_0 - 4dy_0 + 4y_0^2}{2\lambda^2}\right) dy_0 \\
&\overset{(WolframAlpha)}{=} \sqrt{\frac{\pi\lambda^2}{2}} \exp\left(-\frac{a^2 - 2ac + 2b^2 + c^2 + 2d^2}{2\lambda^2}\right) \\
&\qquad \cdot \left(-\sqrt{\frac{\pi\lambda^2}{8}} \exp\left(\frac{(b + d)^2}{2\lambda^2}\right) \operatorname{erf}\left(\frac{b + d - 2y_0}{\sqrt{2\lambda^2}}\right)\right) \Bigg|_{y_0=-\infty}^{\infty} \\
&= \sqrt{\frac{\pi\lambda^2}{2}} \exp\left(-\frac{a^2 - 2ac + 2b^2 + c^2 + 2d^2}{2\lambda^2}\right) \left(2\sqrt{\frac{\pi\lambda^2}{8}} \exp\left(\frac{(b + d)^2}{2\lambda^2}\right)\right) \\
&= \frac{\pi\lambda^2}{2} \exp\left(-\frac{a^2 - 2ac + 2b^2 + c^2 + 2d^2 - b^2 - d^2 - 2bd}{2\lambda^2}\right) \\
&= \frac{\pi\lambda^2}{2} \exp\left(-\frac{(a - c)^2 + (b - d)^2}{2\lambda^2}\right).
\end{aligned}
\tag{C.5}
$$

Plugging Equation C.5 into Equation C.4

$$\boxed{\begin{aligned}\mathscr{L}_{\text{HM}}(\mathcal{P}_\theta, \mathcal{L}) = \sum_i \sum_j \frac{\pi\lambda^2}{2} \exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\lambda^2}\right) \\ + \sum_i \sum_j \hat{p}_i \hat{p}_j \frac{\pi\lambda^2}{2} \exp\left(-\frac{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2}{2\lambda^2}\right) \\ - 2 \sum_i \sum_j \hat{p}_i \frac{\pi\lambda^2}{2} \exp\left(-\frac{(\hat{x}_i - x_j)^2 + (\hat{y}_i - y_j)^2}{2\lambda^2}\right)\end{aligned}} \tag{C.6}$$

C.1.2 Derivative of the Loss by $\hat{p}_k$

$$\begin{aligned}\frac{\partial}{\partial \hat{p}_k} \mathscr{L}_{\text{HM}}(\mathcal{P}_\theta, \mathcal{L}) &= \frac{\partial}{\partial \hat{p}_k} \sum_i \sum_j \frac{\pi\lambda^2}{2} \exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\lambda^2}\right) \\ &\quad + \frac{\partial}{\partial \hat{p}_k} \sum_i \sum_j \hat{p}_i \hat{p}_j \frac{\pi\lambda^2}{2} \exp\left(-\frac{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2}{2\lambda^2}\right) \\ &\quad - \frac{\partial}{\partial \hat{p}_k} 2 \sum_i \sum_j \hat{p}_i \frac{\pi\lambda^2}{2} \exp\left(-\frac{(\hat{x}_i - x_j)^2 + (\hat{y}_i - y_j)^2}{2\lambda^2}\right) \\ &= 0 + \sum_i \sum_j \frac{\partial}{\partial \hat{p}_k} \hat{p}_i \hat{p}_j \frac{\pi\lambda^2}{2} \exp\left(-\frac{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2}{2\lambda^2}\right) \\ &\quad - 2 \sum_i \sum_j \frac{\partial}{\partial \hat{p}_k} \hat{p}_i \frac{\pi\lambda^2}{2} \exp\left(-\frac{(\hat{x}_i - x_j)^2 + (\hat{y}_i - y_j)^2}{2\lambda^2}\right) \\ &= 0 + \sum_i \sum_j \frac{\partial}{\partial \hat{p}_k} \hat{p}_i \hat{p}_j \frac{\pi\lambda^2}{2} \exp\left(-\frac{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2}{2\lambda^2}\right) \\ &\quad - 2 \sum_i \sum_j \frac{\partial}{\partial \hat{p}_k} \hat{p}_i \frac{\pi\lambda^2}{2} \exp\left(-\frac{(\hat{x}_i - x_j)^2 + (\hat{y}_i - y_j)^2}{2\lambda^2}\right) \\ &= \boxed{\begin{aligned}\pi\lambda^2 \sum_i \hat{p}_i \exp\left(-\frac{(\hat{x}_i - \hat{x}_k)^2 + (\hat{y}_i - \hat{y}_k)^2}{2\lambda^2}\right) \\ - \pi\lambda^2 \sum_j \exp\left(-\frac{(\hat{x}_k - x_j)^2 + (\hat{y}_k - y_j)^2}{2\lambda^2}\right)\end{aligned}}\end{aligned} \tag{C.7}$$

C.1.3 Derivative of the Loss by $\hat{x}_k$

$$\frac{\partial}{\partial \hat{x}_k}\mathscr{L}_{\mathrm{HM}}(\mathcal{P}_\theta, \mathcal{L}) = \frac{\partial}{\partial \hat{x}_k}\sum_i \sum_j \frac{\pi\lambda^2}{2}\exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\lambda^2}\right) \tag{C.8}$$

$$+ \frac{\partial}{\partial \hat{x}_k}\sum_i \sum_j \hat{p}_i \hat{p}_j \frac{\pi\lambda^2}{2}\exp\left(-\frac{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2}{2\lambda^2}\right) \tag{C.9}$$

$$- \frac{\partial}{\partial \hat{x}_k} 2\sum_i \sum_j \hat{p}_i \frac{\pi\lambda^2}{2}\exp\left(-\frac{(\hat{x}_i - x_j)^2 + (\hat{y}_i - y_j)^2}{2\lambda^2}\right) \tag{C.10}$$

$$= 0 + \sum_{i \neq k} \hat{p}_i \hat{p}_k \frac{\pi\lambda^2}{2}\exp\left(-\frac{(\hat{x}_i - \hat{x}_k)^2 + (\hat{y}_i - \hat{y}_k)^2}{2\lambda^2}\right)\left(\frac{2\left(\hat{x}_i - \hat{x}_k\right)}{2\lambda^2}\right) \tag{C.11}$$

$$+ \sum_{j \neq k} \hat{p}_k \hat{p}_j \frac{\pi\lambda^2}{2}\exp\left(-\frac{(\hat{x}_k - \hat{x}_j)^2 + (\hat{y}_k - \hat{y}_j)^2}{2\lambda^2}\right)\left(\frac{-2\left(\hat{x}_k - \hat{x}_j\right)}{2\lambda^2}\right) \tag{C.12}$$

$$- 2\sum_j \hat{p}_k \frac{\pi\lambda^2}{2}\exp\left(-\frac{(\hat{x}_k - x_j)^2 + (\hat{y}_k - y_j)^2}{2\lambda^2}\right)\left(\frac{-2\left(\hat{x}_k - x_j\right)}{2\lambda^2}\right) \tag{C.13}$$

$$= \boxed{\begin{aligned} &\hat{p}_k \pi \sum_j \exp\left(-\frac{(\hat{x}_k - x_j)^2 + (\hat{y}_k - y_j)^2}{2\lambda^2}\right)\left(\hat{x}_k - x_j\right) \\ &- \hat{p}_k \pi \sum_i \hat{p}_i \exp\left(-\frac{(\hat{x}_i - \hat{x}_k)^2 + (\hat{y}_i - \hat{y}_k)^2}{2\lambda^2}\right)\left(\hat{x}_k - \hat{x}_i\right) \end{aligned}} \tag{C.14}$$

$$\tag{C.15}$$

C.2 Checkerboard Corner Detection Experiment

C.2.1 Additional Results

Additional experiments have been conducted for the checkerboard corner localization experiment.

Table C.1: Corner localization performance on our synthetic test set. The mean-absolute deviation (MAD) from ground-truth (in units of original pixel size) as well as precision, recall, and F_1-scores (with a tolerance of 3 pixels) are reported.

	Methods	MAD	Rec.	Prec.	F_1
Classic	OCamCalib (Scaramuzza et al., 2006)	0.362	**97.0**	99.8	**98.4**
	Rochade (Placht et al., 2014)	0.147	59.1	**99.9**	74.3
	OpenCV (Bradski, 2000)	0.137	45.6	89.5	60.4
	MATLAB (Geiger et al., 2012)	**0.086**	65.8	96.4	78.2
Learn.	DL-Heatmap (sim. (Donné et al., 2016))	0.488	98.1	99.7	98.9
	+ Refinement (sim. (Graving et al., 2019))	0.130	98.1	99.7	98.9
	OURS	**0.105**	**99.3**	**99.9**	**99.6**

Experiment: Sub-Pixel Accuracy on Synthetic Test Data

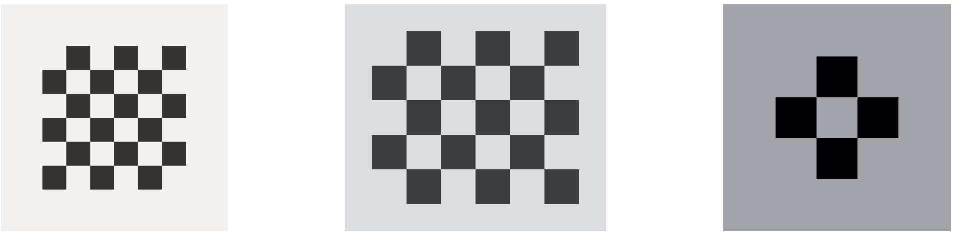
To evaluate the absolute sub-pixel accuracy of our method, we test it on a synthetic test dataset generated analogously to the training dataset described in the main text. (Appendix C.2.2 illustrates the variety of the data generated.) The exact ground-truth corner locations are thus known by construction. For all benchmarks, 1000 synthetic images are used for testing. The results are summarized in Table C.1. Overall, our method consistently outperforms state-of-the-art algorithms both in terms of absolute spatial precision—with typical errors in the order of $\approx 1/10$th of a pixel—as well as in terms of detection rates. Especially noteworthy is the fact that the excellent spatial precision of our approach is not a result of a low recall rate. In contrast, the lower recall values achieved by other methods, show that they often fail to detect challenging corners (due to distortions, noise, low contrast), and hence only the most easily detectable ones are taken into account when computing the spatial error for these methods. For instance, the remarkable mean absolute error of 0.086 pixels achieved by MATLAB (Geiger et al., 2012) results from only 65.8% of the corners it was able to detect in the first place. Finally, our method does not require additional information about the grid structure of the calibration board and its size—in contrast to (Bradski, 2000; Placht et al., 2014; Scaramuzza et al., 2006) which leverage this information to refine the predictions.

Further, we note that the above state-of-the-art methods, with the exception of deep learning-based ones, rely on traditional image processing techniques that have been hand-crafted specifically for this task. Hence, they do not generalize well to other applications. In contrast, we hypothesize that our approach can be straightforwardly applied to any sub-pixel localization task, such as the accurate analysis of medical images.

Finally, in terms of inference efficiency, our point prediction approach is significantly faster (3.43 images/second on a 4.3 GHz CPU, and 100+ with an NVidia TitanXp GPU) than OpenCV (2.04 images/s), the saddle point-based ROCHADE (1.10 images/s), and the heatmap-based approach (1.15 images/s) which is slowed down by the corner-refinement step and the lack of spatial downsampling.

C.2.2 Synthetic Dataset

The code for the generation of the synthetic checkerboard dataset is provided[2] (c.f. `create_dataset.py`).



(6x6) board

(6x5) board

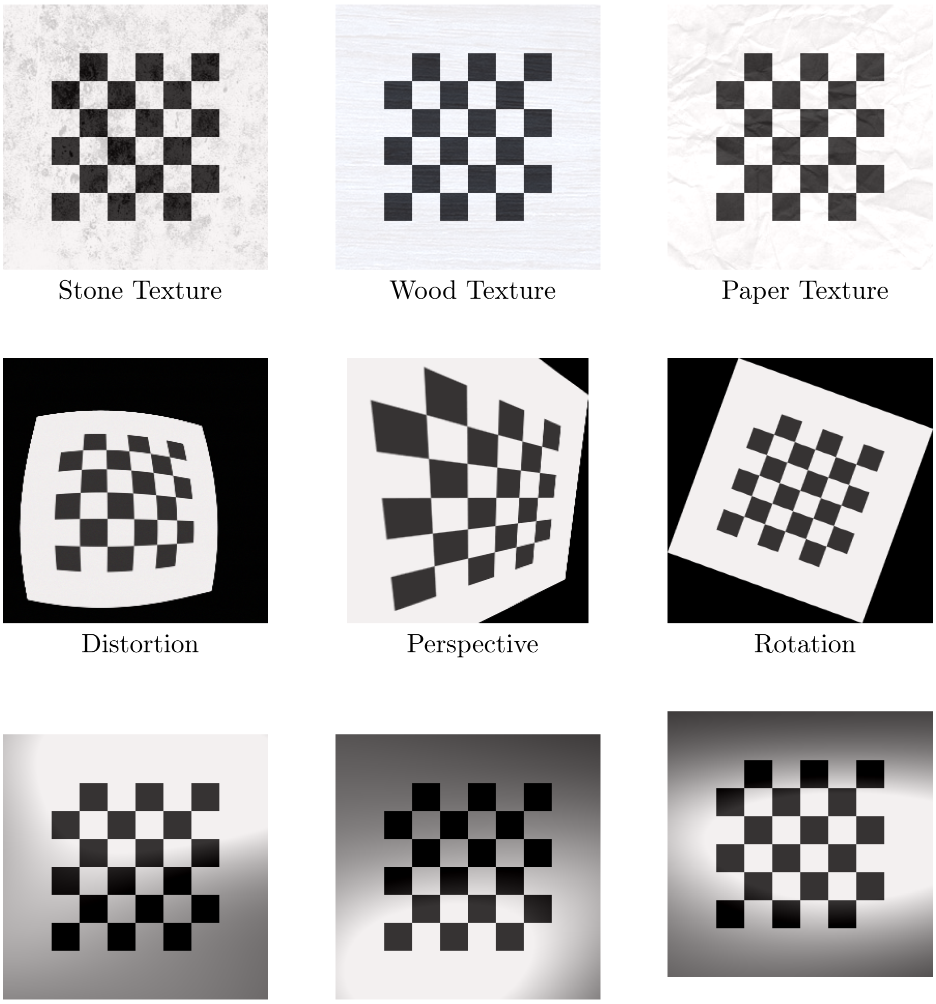
(3x3) board

Figure C.1: Initial checkerboard of various size, shape and coloration

Overall, initial checkerboards are first generated by randomly sampling their size, shape, and coloration (see Figure C.1). Then, some of these checkerboards are projected onto textures such as wood, paper, and stone (see the first row of Figure C.2). Finally, between one and eight (sampled uniformly at random) of the following eight transformations are applied to these checkerboards: blurring, lighting, sharpening, contrast change, scaling, distortion, perspective transform, and rotation (see Figure C.2 for examples). All of these transformations have hyperparameters that are also sampled at random, such as the level of distortion or the angle of the rotation.

While this process allows for a rich variety of checkerboards to be generated (see Figure C.3), most importantly it allows to track the location of the corners with high levels of precision. Indeed, spatial transformations (e.g., rotation and

[2] https://github.com/SchroeterJulien/ACCV-2020-Subpixel-Point-Localization



Stone Texture

Wood Texture

Paper Texture

Distortion

Perspective

Rotation

Figure C.2: Examples of transformations applied to the initial checkerboard images

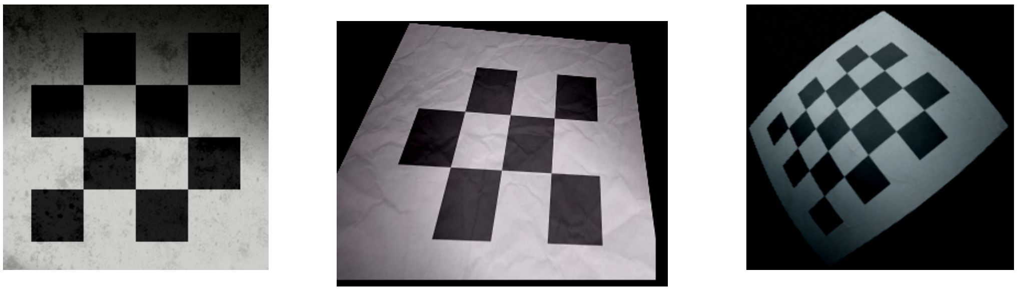


Figure C.3: Example checkerboard training images from our synthetic dataset

perspective transform) can be applied to both checkerboard images and corner locations without any significant approximation. Thus, we are able to leverage these precise labels as the ground truth to train our sub-pixel precision model.

C.3 Golf Swing Event Localization Experiment

C.3.1 Additional Results

Table 5 (in the main text) reports the mean golf swing event detection accuracy over all event classes, i.e., address (A), toe-up (TU), mid-backswing (MB), top (T), mid-downswing (MD), impact (I), mid-follow-through (MFT), and finish (F) (McNally et al., 2019). However, each of these classes differs drastically from one another, especially in terms of temporal ambiguity and detection difficulty. Therefore, in order to assess whether the performance improvement achieved by our model can be attributed to a few event classes only or whether the improvement is consistent across all classes, we provide a detailed report of per class detection accuracy in Table C.2.

Overall, our method displays consistent improvement on most event classes and decimation rates. (Given the relatively moderate size of the testing splits and the stochastic nature of the learning process, a few outliers are to be expected.) Our approach not only improves the detection accuracy of temporally ambiguous events (e.g., A and F) but also pushes further the detection capabilities on more easily detectable classes (e.g., MD and MFT).

Table C.2: Golf swing event detection accuracy (within a ±1 frame tolerance) per class as a function of decimation factor δ. Averages are reported over 4 folds. The architecture is from (McNally et al., 2019).

	Loss	$\delta = 1$ frame	2 frames	4 frames	8 frames	16 frames
A	Naïve upsampling	18.2	18.4	20.9	22.1	20.5
	Frame interpolation	— " —	19.4	23.1	19.0	15.6
	Dense classification	— " —	19.7	22.9	21.3	21.2
	Ours	**23.9**	**24.6**	**23.4**	**25.1**	**22.4**
TU	Naïve upsampling	79.0	80.5	68.7	47.7	28.1
	Frame interpolation	— " —	73.3	71.2	60.2	42.5
	Dense classification	— " —	**81.8**	78.8	75.6	63.1
	Ours	**80.1**	77.9	**79.0**	**76.5**	**69.6**
MB	Naïve upsampling	81.3	83.7	68.9	46.6	30.4
	Frame interpolation	— " —	82.3	78.6	66.3	44.2
	Dense classification	— " —	82.9	**84.4**	78.6	63.6
	Ours	**86.6**	**86.4**	84.3	**81.9**	**68.8**
T	Naïve upsampling	62.3	62.8	60.6	43.1	25.1
	Frame interpolation	— " —	64.5	64.6	62.4	45.3
	Dense classification	— " —	69.4	72.8	69.3	67.4
	Ours	**70.4**	**70.5**	**75.7**	**78.0**	**70.5**
MD	Naïve upsampling	95.7	95.3	83.9	52.0	30.3
	Frame interpolation	— " —	95.1	94.0	85.8	58.6
	Dense classification	— " —	**96.3**	94.9	89.1	**77.8**
	Ours	**96.2**	92.9	**95.0**	**90.4**	75.7
I	Naïve upsampling	94.7	94.6	80.4	57.8	14.1
	Frame interpolation	— " —	94.7	93.5	88.8	60.2
	Dense classification	— " —	**96.3**	**94.6**	91.2	79.3
	Ours	**95.3**	96.1	**94.6**	**92.1**	**80.4**
MFT	Naïve upsampling	94.4	94.6	80.6	71.5	31.6
	Frame interpolation	— " —	92.9	92.0	84.6	55.4
	Dense classification	— " —	**94.8**	92.9	87.2	75.8
	Ours	**94.9**	94.3	**93.3**	**91.3**	**77.7**
F	Naïve upsampling	15.9	17.8	15.9	15.8	10.8
	Frame interpolation	— " —	17.1	20.1	16.9	13.8
	Dense classification	— " —	15.8	18.1	18.4	14.4
	Ours	**20.0**	**20.2**	**20.4**	**20.9**	**21.5**