IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 30, 2020, accepted December 25, 2020, date of publication December 31, 2020,
date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3048622

End-to-End Sinkhorn Autoencoder With Noise Generator

KAMIL DEJA[1], JAN DUBIŃSKI[1], PIOTR NOWAK[2], SANDRO WENZEL[3], PRZEMYSŁAW SPUREK[4], AND TOMASZ TRZCIŃSKI[1,5], (Member, IEEE)

[1]Faculty of Electronics and Information Technology, Warsaw University of Technology, 00-661 Warsaw, Poland
[2]Faculty of Physics, Warsaw University of Technology, 00-661 Warsaw, Poland
[3]CERN, 1211 Geneva, Switzerland
[4]Faculty of Mathematics and Computer Science, Jagiellonian University, 31-007 Kraków, Poland
[5]Tooploox, 53-601 Wrocław, Poland

Corresponding author: Kamil Deja (k.deja@ii.pw.edu.pl)

This work was primarily supported by the Polish National Science Center Project under Grant UMO-2018/31/N/ST6/02374 in part Grant UMO-2016/21/D/ST6/01946, and in part Warsaw University of Technology Grant IDUB-POB-FWEiTE-1. Computing resources were provided by the Dean of the Faculty of Electronics and Information Technology, Warsaw University of Technology, under Project II/2017/GD/1.

ABSTRACT In this work, we propose a novel end-to-end Sinkhorn Autoencoder with a noise generator for efficient data collection simulation. Simulating processes that aim at collecting experimental data is crucial for multiple real-life applications, including nuclear medicine, astronomy, and high energy physics. Contemporary methods, such as Monte Carlo algorithms, provide high-fidelity results at a price of high computational cost. Multiple attempts are taken to reduce this burden, e.g. using generative approaches based on Generative Adversarial Networks or Variational Autoencoders. Although such methods are much faster, they are often unstable in training and do not allow sampling from an entire data distribution. To address these shortcomings, we introduce a novel method dubbed *end-to-end Sinkhorn Autoencoder*, that leverages the Sinkhorn algorithm to explicitly align distribution of encoded real data examples and generated noise. More precisely, we extend autoencoder architecture by adding a deterministic neural network trained to map noise from a known distribution onto autoencoder latent space representing data distribution. We optimise the entire model jointly. Our method outperforms co mpeting approaches on a challenging dataset of simulation data from Zero Degree Calorimeters of ALICE experiment in LHC. as well as standard benchmarks, such as MNIST and CelebA.

INDEX TERMS Computer simulation, generative modeling, machine learning.

I. INTRODUCTION

Multiple real-life applications rely heavily on detailed simulations of ongoing processes, from atomic structures in nuclear medicine (e.g. tomography) [38] or genetics [19], to astrophysics [47]. This is also true for the Large Hadron Collider (LHC) [11] – one of the biggest scientific programmes currently being carried out worldwide. In the LHC, two beams of particles are accelerated to the ultra-relativistic energies and brought to collide. In such an environment, high energy density leads to the appearance of very rare phenomena. To understand these processes, physicists compare recorded data with accurate theoretical models simulations. Currently employed simulation techniques use complex Monte Carlo processing in order to compute all possible interactions between particles and matter. Such an approach produces accurate results at the expense of high computational cost.

Therefore multiple attempts are taken to speed up this processing, including those that leverage state of the art generative models [9], [21], [30] such as Generative Adversarial Networks [16] (GANs) or Variational Autoencoders [23] (VAE). While the above methods are much faster than standard simulations, they suffer from limitations which make them unsuitable for reliable real data simulation tool. Training of Generative Adversarial Networks is often unstable and may result in limited quantitative properties [2], [3]. On the other hand, Variational Autoencoders converge in steady manner. However, because of the maximum likelihood approximation they also produce blurry results with both

The associate editor coordinating the review of this manuscript and approving it for publication was Changsheng Li.

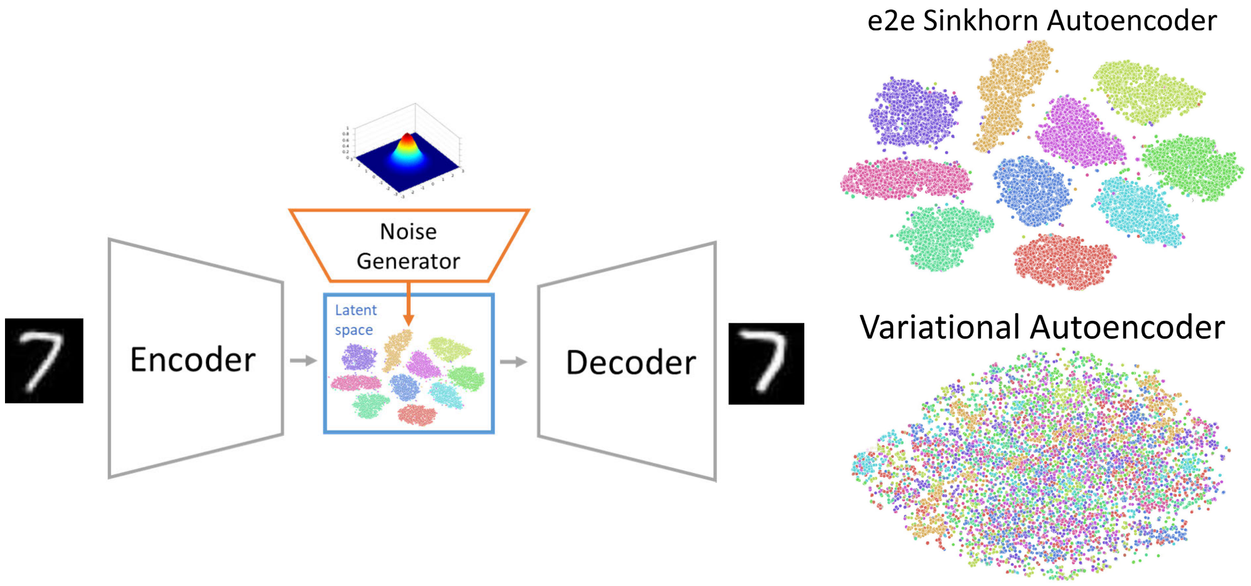


FIGURE 1. Schematic visualisation of end-to-end Sinkhorn Autoencoder processing (left). TSNE visualisation of latent space for MNIST dataset (right). Our conditional e2e Sinkhorn Autoencoder (top) and conditional VAE (bottom). Our model does not restrict latent space to the normal distribution, therefore classes may be even linearly separable.

visual and statistical problems. In this work we address those shortcomings, and propose a novel solution built on top of recent advancements in generative modelling.

To stabilise the training of GANs, in [1] authors propose to substitute KL divergence with Wasserstein distance. Since it was proven to be more reliable, a new model dubbed Wasserstein Autoencoder (WAE) [39] was proposed, where the same metric is used to regularise distribution of data in autoencoder's latent space. In WAE, authors employed two different methods to calculate Wasserstein distance – MMD and adversarial critic. While original processing of WAE provides high-quality results, new autoencoder based architectures such as Sliced Wasserstein Autoencoder [26] or Sinkhorn Autoencoder [31] introduce faster non-adversarial Wasserstein distance approximations.

Thanks to the autoencoder based processing, the above techniques provide stable training. However, they require significant regularisation on the original autoencoder's latent space, which enables sampling from parametrised distribution. This regularisation enforces encoding in a particular hyperspace that often leads to limited representation capabilities. In particular, the commonly used normal distribution hinders linear separability of different components of complex distribution [15]. It also does not allow sparse representation obtained via relu activation [41].

To address these shortcomings, we propose a new generative model build on top of the Sinkhorn Autoencoder [31]. Instead of restricting autoencoder to encode examples on the parametrised distribution, we approximate it with explicit *noise generator* implemented through the additional deterministic neural network, as presented in Fig. 1. We input noise from a known distribution (*e.g.* normal) to this network and encode it to follow the distribution of real data in the autoencoder's latent space. Although such an approach allows us to generate new data samples from a parametrised distribution, thanks to an additional neural network, we do not regularise our encoder's latent space with such constraints. To our knowledge our end to end Sinkhorn Autoencoder is the first generative autoencoder without explicit constraint on the latent space.

Problem of constrained autoencoder latent space is even more evident in conditional generative models, as it hinders separation based on a priori information.

Currently proposed conditional generative models such as conditional VAE (condVAE) [37] include additional a priori parameters to the encoder and decoder. At the same time condVAE regularises latent space to follow normal distribution. Therefore, the model learns to encode information related to classes only in encoder and decoder, while in latent space all of the examples are shuffled into a single manifold as presented in Fig. 1. This behaviour limits classes separation, since they have to be learned in decoder from one common continuous distribution.

In this work we introduce a conditional version of our solution. Contrary to prior methods, we do not input conditional parameters into the encoder and decoder. We allow autoencoder to encode different classes in different areas of the latent space, while we match them with the conditional noise generator. Such an approach, is more suitable for different (e.g. imbalanced) conditional classes. It allows to encode data into a more natural, disentangled representation with clear classes separation as depicted in Fig. 1.

We evaluate the quality of our standard and conditional end-to-end Sinkhorn Autoencoder with commonly used benchmark datasets, such as MNIST [27] and CelebA [29], and achieve state-of-the-art results. To show generalisation of our solution we then apply it to the problem of fast simulation of particle showers in High Energy Physics (HEP). We show that our method allows to generate high-quality calorimeter responses from the whole distribution of original data. The superiority of our model is even more pronounced on this dataset.

The main contributions of this work are:

- A new non-adversarial end-to-end generative model with explicit noise generator.
- A novel conditional generative model based on the autoencoder architecture which, contrary to the currently employed models, leaves the structure of autoencoder's latent space intact.

The remainder of this work is organised as follows. In Sec. II we describe related works in the field of autoencoding generative modelling and fast simulations for HEP. Sec. III introduces our end-to-end Sinkhorn Autoencoder method followed by its conditional version. We conclude this work in Sec. IV, with experiments on MNIST, CelebA and HEP datasets and a description of potential further studies.

II. RELATED WORK

We divided the related work section into three parts. First, we describe autoencoder based generative models. Then, existing solutions improving autoencoder properties by adding a neural network in the latent space. In the end, we discuss similar approaches that combine GANs and autoencoder architectures.

a: AUTOENCODER BASED GENERATIVE MODELS

Autoencoders are commonly used for dimensionality reduction [43], representation learning [42] or anomalies detection [35] (also in HEP [33]). Its direct application in generative modelling is hard because of the natural tendency to encode examples into complex latent space. Without regularisation such an encoding often results in non-overlapping distribution with discontinuities as in Fig. 1. While it enables accurate reconstructions, it makes sampling from a latent space nearly impossible.

Therefore, different generative models based on autoencoders are trained to regularise encoder so that the latent space is continuous and follows a parameterised distribution. Authors of the first in the field Variational Autoencoder [23] propose regularisation on latent space based on KL divergence $D_{KL}(P_X, P_Z)$, where P_X is the original encoded data distribution, and P_Z is the prior distribution. This distance between the true and the model prior distribution is computed in the variational scheme.

Wasserstein metric [39] introduces a significant improvement in the construction of autoencoders leading to the inception of the Wasserstein Autoencoders (WAE) [39] model. WAE changes the regularisation objective from KL divergence to Wasserstein distance. In [39], the authors introduce two possible methods for the application of Wasserstein distance on the autoencoder's latent space. The first one is based on the *maximum mean discrepancy* [17] (MMD) technique, while the second one, similarly to [1], uses a neural network as a critic. In such a framework the encoder is trained to store data examples in the latent space as close as possible to the prior distribution.

In [26], the authors present the Sliced-Wasserstein Autoencoder (SWAE), which substitutes MMD with an approximation obtained by a cumulative distribution of one-dimensional distances. The main innovation of SWAE was the introduction of the sliced-Wasserstein distance, a fast to estimate metric for comparing two distributions based on the mean Wasserstein distance of one-dimensional projections. This solution is much simpler, but as reported in [31] it results in a lower diversity of generated results.

The modification of SWAE is presented in [25], where the authors constructed the Cramer-Wold AutoEncoder (CWAE), by replacing the sliced Wasserstein distance in SWAE by using CW-distance between distributions. CWAE model can be seen as a version of the WAE-MMD method with a choice of a specific kernel (Cramer-Wold kernel).

To solve limitations of the Wasserstein distance, in [31] authors introduce the Sinkhorn Autoencoder (SAE), which approximates and minimises the p-Wasserstein distance in a latent space via backpropagation through the Sinkhorn algorithm [7]. SAE is thus able to work with different metric spaces and priors with minimal adaptations. In particular, in [31] authors experiment with the normal and hypersphere prior.

We build our model on top of Sinkhorn Autoencoder [31]. In our end-to-end Sinkhorn Autoencoder, we benefit from the processing of those standard models. However, we use faster and more stable approximations of the Wasserstein distance, and we do not regularise the autoencoder's latent space to the prior distribution. Last but not least, we propose a neural network trained to map noise from a known distribution onto autoencoder's latent space.

b: LATENT SPACE GEOMETRY

In our approach, we use an additional neural network (noise generator) to improve latent space geometry. Similar approaches were used in the literature.

In [44], [46] authors apply normalizing flows [24] in the latent space. Thanks to this, latent distribution (which is similar to the standard Gaussian) is transformed into the Gaussian prior. In [8] authors present a similar solution based on adding additional autoencoder in the latent space (TwoStageVAE model). They present theoretical results stating that the VAE model does not properly approximate prior distribution in the latent space. But, as they propose, the second VAE model is able to correct the distribution in the latent. On the other hand, in [40] authors increase a limit of the capacity of the prior by using a new prior that is expressed as a mixture of variational posteriors (VampPrior). The VampPrior consists of a mixture of Gaussians with components conditioned on learnable pseudo-inputs. Such prior is implemented as a two-level hierarchical model.

All the above approaches give good results but use very complicated structures or a two-stage training procedure. In our paper, we show a simple and elegant end-to-end generative model with an explicit noise generator.

c: AUTO-ENCODER ARCHITECTURES WITH ADVERSARIAL TRAINING

Training of GANs is unstable and may result in limited quantitative properties [2], [3]. On the other hand, Variational Autoencoders converge much better but tend to generate blurry samples when applied to natural images. Such issues can be partially solved by using autoencoder architecture as a generator in GAN architecture (with adversarial training).

In [45] authors propose Latently Invertible Autoencoder (LIA) architecture, which uses an invertible network in the latent space of VAE. The decoder of LIA is first trained as a standard GAN with the invertible network and then the partial encoder is learned from a disentangled autoencoder by detaching the invertible network from LIA.

On the other hand, in [32] authors introduce an autoencoder architecture by modifying the original GAN paradigm. The generator and discriminator are decomposed into two networks. In consequence, we obtain two additional latent spaces, where we add regularisation terms. The model is optimised by adversarial training.

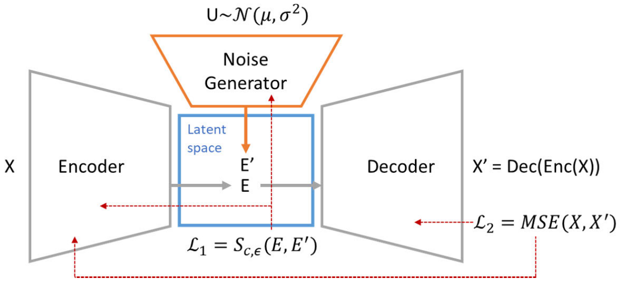
All the above approaches give sharp images, but most adversarial training limitations are not solved. In our paper, we present a non-adversarial end-to-end generative model with an explicit noise generator.

We introduce our model together with its application to the problem of calorimeters response simulations. The majority of current works in this field focus on GAN architectures, e.g. CaloGAN [30] or [21]. In those works, the authors adapt the conditional DCGAN [34] architecture. We tackle the same problem from a different perspective using the autoencoding generative model. Therefore, in Sec. IV, we compare our results to the DCGAN model – a current state-of-the-art solution to the problem of calorimeters response simulation.

III. SINKHORN AUTOENCODER WITH NOISE GENERATOR

In this section, we describe our new end-to-end Sinkhorn Autoencoder generative model. Fig. 2 shows its general architecture. Then, we examine three parts of the final model optimisation objective: a reconstruction loss, a Sinkhorn loss on the latent space, and additional regularisations. Finally, we introduce a conditional version of our model.

In the Sinkhorn Autoencoder work [31], authors introduce a generative model framework where the Sinkhorn algorithm is used to match the autoencoder's latent space with a known distribution. In our work, we leverage this analysis and present an extended version of this method with a trainable prior approximator dubbed *noise generator*. We implement it as a neural network. With such an approach, we can use a gradient obtained from the Sinkhorn loss to simultaneously train the encoder and the noise generator to converge into a similar distribution on the latent space.



$U \sim \mathcal{N}(\mu, \sigma^2)$
Noise Generator
Latent space
X
Encoder
E'
E
Decoder
X' = Dec(Enc(X))
$\mathcal{L}_2 = MSE(X, X')$
$\mathcal{L}_1 = S_{c,\epsilon}(E, E')$

FIGURE 2. **The architecture of the Sinkhorn Autoencoder with a neural network as an explicit noise generator. Red arrows indicate the gradient flow. Reconstruction Loss L_2 is backpropagated through decoder and encoder, while Sinkhorn loss L_1 is propagated in two directions to encoder and noise generator. Encoder network is optimised with a sum of L_1 and L_2 losses.**

In Fig. 2 we demonstrate the general layout of our solution. It consists of three neural networks – *encoder*, *decoder* and *noise generator*. As presented in Fig. 2, the loss of our model composes of two main terms - L_1 - Sinkhorn loss on latent space and L_2 reconstruction loss of the autoencoder. Additionally, to prevent overfitting and promote diversity, we employ regularisations on both model weights and autoencoder's latent space.

A. RECONSTRUCTION LOSS

The core of our network is based on a standard autoencoder. Hence, it follows the original autoencoder training procedure. The encoder is trained to map the original data X into the latent space E, while the decoder is optimised to reconstruct original data examples X'. In this part, we experiment with two different losses. Standard Mean Squared Error loss $MSE(X, X') = (X - X')^2$ is a simple choice, but as denoted in [6] it may lead to blurry images. Therefore, following [6], we also employ Laplacian pyramid Lap_1 loss presented below:

$$Lap_1(x, x') = \sum_j 2^{2j} |L^j(x) - L^j(x')|_1 \tag{1}$$

where $L^j(x)$ is the j-th level of the Laplacian pyramid representation of $x \in X$ [28].

Similarly to [6], as a final reconstruction objective we use a weighted mean of the standard mean squared error and the Lap_1 loss.

$$L_{recon}(X, X') = \alpha Lap_1(X, X') + MSE(X, X') \tag{2}$$

where α is a scaling parameter.

B. SINKHORN LOSS

To map the generated noise onto autoencoder's latent space, we leverage Sinkhorn algorithm [7]. However, contrary to [31], we use gradient obtained from this loss to train both our *encoder* and additional network – *noise generator*. We train both of those models so that their outputs – encoded data and noise – follow the same distribution in the latent space.

This proceeds as follows. First, we encode the batch of original images X to obtain their encoded representation E. At the same time, we process a random vector U sampled from a known distribution (e. g. $\mathcal{N}(0, 1)$) through the noise generator. It creates noise representation E' in the same latent space as for encoded images.

Then we compute the distance between real data and noise embeddings. Following WGAN or WAE architectures, we could approximate this with an additional neural network, but to simplify the solution, we opt for entropy regularisation of the Wasserstein distance implemented with Sinkhorn algorithm.

For this purpose we follow [13], [14] to define the entropy regularised Optimal Transport cost with $\epsilon \geq 0$ as:

$$\tilde{S}_{c,\epsilon}(P_X, P_Y) = \inf_{\Gamma\in\Pi(P_X,P_Y)} \mathbb{E}_{(X,Y)\sim\Gamma}[c(X, Y)] + \epsilon \cdot KL(\Gamma, P_X \otimes P_Y). \quad (3)$$

As suggested in [31] we remove the entropic bias of the above approximation with three passes of the Sinkhorn algorithm, as presented below:

$$S_{c,\epsilon}(P_X, P_Y) = \tilde{S}c, \epsilon(P_X, P_Y) - \frac{1}{2}(\tilde{S}c, \epsilon(P_X, P_X) + \tilde{S}c, \epsilon(P_Y, P_Y)) \quad (4)$$

With the above equation we calculate the loss value for two representations of encoded images E and generated noise E' in a batch-wise manner as $S_{c,\epsilon}(E, E')$. For the Wasserstein cost function c we use standard 2-Wasserstein distance with euclidean norm $c(x, y) = \frac{1}{2}||x-y||_2^2$. As indicated in [13] $S_{c,\epsilon}$ deviates from the original Wasserstein distance by approximately $O(\epsilon log(1/\epsilon)$, hence we keep our ϵ small to avoid the influence on network's convergence. In practice, we use the efficient implementation of the Sinkhorn algorithm with GPU acceleration from GeomLoss package [12].

C. END-TO-END SINKHORN AUTOENCODER OBJECTIVE

To improve the diversity of generated images, we include additional regularisation on the autoencoder's latent space. For this purpose, we adapt diversity regularisation proposed in [4]. In this work, authors compute a similarity matrix SIM_C to assess the diversity in the neural network's weights according to the cosine similarity between the outputs of consecutive layers.

We adapt this technique in our model to measure the similarity between all of the encoded real data examples from the batch. Then, following [4] we compute the regularisation as a sum of these similarities as presented in 5:

$$R_s(\mathbf{y}) = p \sum_{i=1}^{bs} \sum_{j=1,j\neq i}^{bs} m_{i,j} \left(SIM_C \left(\mathbf{y}_i, \mathbf{y}_j^T\right)\right)^2 \quad (5)$$

where p is a scaling factor bs is batch size and m is a binary mask variable which drops pairs below threshold τ.

$$m_{i,j} = \begin{cases} 1, & \left|SIM_C\left(\mathbf{y}_i, \mathbf{y}_j^T\right)\right| \geq \tau \\ 0, & otherwise \end{cases} \quad (6)$$

Additionally, we also experiment with different regularisations on autoencoder's weights. In our experiments, we observed better convergence with L_2 regularisation on the last layer of our encoder.

Below we outline the joint loss function of our autoencoder as a sum of four elements: reconstruction loss, Sinkhorn loss between generated noise and original encoded images, and additional regularisations on the network's latent space and weights values θ.

$$\begin{aligned} L_{sum}(X) = {} & \alpha L_{recon}(X, dec(enc(X))) \\ & + \beta Sc, \epsilon(enc(X), gen(X' \sim \mathcal{N}(0, 1)) \\ & + \delta R_s(enc(X)) + \gamma reg(\theta) \end{aligned} \quad (7)$$

D. CONDITIONAL SINKHORN OBJECTIVE

While the goal for most applications of generative modelling is to generate more examples from a given distribution, for certain tasks we have to include additional information about simulated data. This is also the case for HEP, where we want to simulate possible responses of a calorimeter for a given particle. For the purpose of conditional images generation, we propose a simple adjustment to the standard version of our processing presented in the previous section.

Firstly, for a given batch of samples X with corresponding conditioning variables Q, we propose to pass the original conditional values to the *noise generator* as a separate input for the neural network. Thanks to this, we change our *noise generator* to encode random noise U with respect to conditional values Q.

Secondly, we train the *noise generator* and the *encoder* to encode examples with similar conditional values near to each other in the latent space. For that purpose, we first encode all of the examples X into their representation in the latent space E. Then, for each example $e \in E$, we concatenate its encoding with corresponding conditional values $q \in Q$. We perform the same operation for noise encodings E' and the same conditional parameters Q obtained from original training data. Finally, we pass concatenated vectors through the Sinkhorn algorithm, calculating the loss value.

$$S_{c,\epsilon}(enc(X) \oplus Q, gen(U \sim \mathcal{N}(0, 1), Q) \oplus Q) \quad (8)$$

Thanks to this approach we add cost to the original Sinkhorn objective. For the $Wasserstein_2$ metric it is equal to the euclidean norm distance between different conditional values. However, depending on the nature of the a priori information Q and the potential real cost of generating samples from the distribution related to other conditions, it might be beneficial to scale it accordingly.

With this approach, contrary to the other conditional generative models such as conditional VAE or WAE, our solution

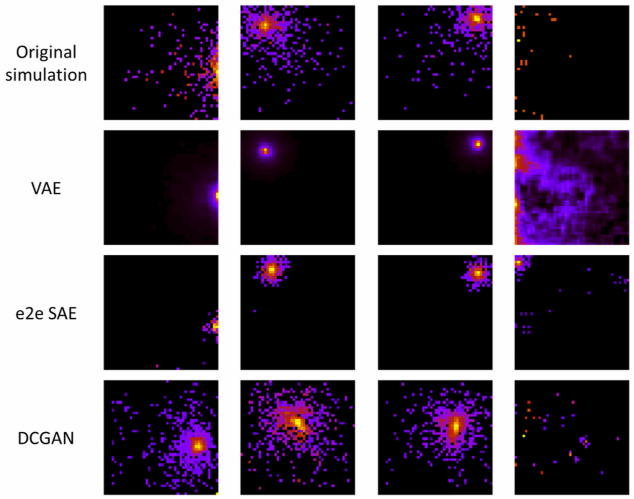


FIGURE 3. Examples of calorimeters response simulations with different methods. Although results from GAN are visually sound with collisions, model was not able to properly capture relations from conditional values. Our solution does not reproduce all of residual values, but it outperforms other methods in terms of accuracy of positioning for the most significant centre of the collision.

leaves the original autoencoder's latent space intact. We do not enforce it to encode different classes into the general consistent distribution. Thanks to the fact that conditional parameters are included in the noise generator, we can observe that autoencoder distribute different classes in separate areas of its latent space. We compare both exemplar latent spaces for the MNIST dataset in figure Fig. 1.

IV. EXPERIMENTS

In this section, we evaluate the performance of our end-to-end Sinkhorn Autoencoder model in reference to other generative solutions. Primarily, we compare the results of our experiments to other autoencoder based generative approaches. For that purpose, we use two standard benchmarks: the MNIST dataset of handwritten digits [27] and CelebA dataset of celebrity face images [29], cropped to 64×64 pixels. We also evaluate different conditional generative solutions, including the adversarial example of conditional Deep Convolutional GAN [34] on a challenging dataset of calorimeter response simulations.

The dataset for the latter, which we call HEP, consists of 117 817 Zero Degree Calorimeter responses to colliding particles, calculated with the full GEANT4 [19] simulation tool. Each particle response simulation starts with the single particle described with 9 attributes (mass, momenta, charge, energy, primary vertex). Particle is then propagated through the detector where simulation tools are employed to calculate all of its interactions with the detector's matter. The final outcome of a simulation is the result of those interactions observed as a total energy deposited in calorimeter's fibres. Since those fibres are arranged in a grid with 44×44 size, we can treat the final response as an image with 44×44 pixels. Visualisation of such simulations is presented in Fig. 3. Although, resulting images are non-deterministic, they are highly affected by initial particle attributes. In principle, particle type (mass and charge) defines the trajectory of particle, while it's energy and momenta directly influence the luminosity of the response.

In our experiments, we use network architectures proposed in Wasserstein Autoencoder [39]. Therefore for the CelebA dataset, we use a convolutional deep neural network with 4 convolutional/deconvolutional layers for both encoder and decoder with 5×5 filters. Additionally we use batch normalisation [20] after each convolutional layer. For our noise generator, we use a simple, fully connected network with 3 hidden layers and ReLU activations. We optimise our networks with Adam [22] in batches of 1000 examples.[1]

To assess the quality of generated samples we use *Fréchet inception distance* (FID) introduced in [18]. As proposed in [5], for the MNIST dataset, we change the original Inception neural network to the LeNet based convolutional classifier. While FID is criticised for approximating distributions with Gaussians [36], we also introduce a new measure to monitor the diversity of generated examples. After propagating original and generated images up to the LeNet's penultimate layer, we compare their distributions with Wasserstein distance approximation implemented with Sinkhorn algorithm. We refer to this measurement as *Sinkhorn*.

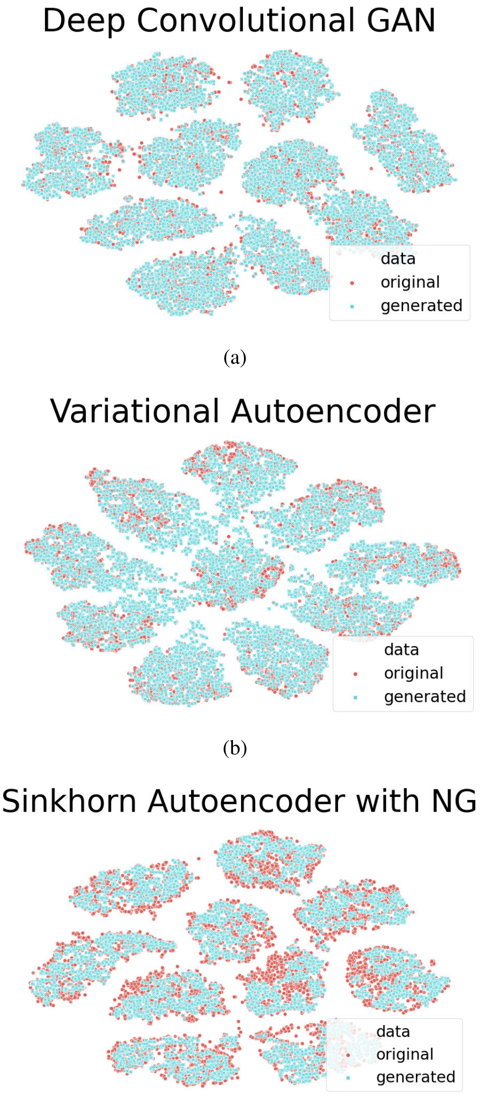
For the HEP dataset, we benefit from the fact that the original data is simulated, hence we can assess the quality of generated samples on the basis of their physical properties. Following the calorimeter's specification [10], we sum pixels of generated images into five *channels*. Calculated channels are usually employed for the calibration purposes, since they represent well the physical properties of simulated collision. To assess the quality of generations we compare them to the original full simulations by measuring mean absolute error between channels from original and generated responses with the same input. In table 1 we refer to this measurement as MAE. While original simulations are also non-deterministic and there are several realistic outputs for the same particle, their general characteristic captured by channels deviates by a small margin of 6.59 MAE. This error allows us to measure how accurate our model is in terms of data generation with respect to conditional values. To measure how well it reproduces the whole distribution of channel values, we also calculate the Wasserstein distance between original and generated channels distribution on the whole test-set.

As presented in Tab. 1, our conditional model outperforms other non-adversarial solutions on both MNIST and HEP datasets. As shown in Fig. 3, HEP dataset remains challenging for all generative models. For VAE, we can see the blurry generations as an outcome of regularisation with the normal distribution. Thanks to the adversarial training with discriminator instead of averaged reconstruction error DCGAN produces visually more attractive results. However,

[1] Code for our work is available at https://github.com/KamilDeja/e2e_sinkhorn_autoencoder

TABLE 1. Results Comparison for Conditional Generative Models on MNIST and HEP Datasets. Our Conditional e2eSAE Outperforms Other Conditional Methods on Both Standard Benchmark as Well as HEP Dataset. In Terms of the Latter our Model is Able to Simulate the Exact Localisation of Collision With Very High Accuracy.

	MNIST		HEP	
Model	FID	Sinkhorn	MAE	Sinkhorn
cond VAE	6.61	30.13	23.13 (±65.53)	14.92
cond WAE (MMD)	34.73	30.44	43.54 (±55.23)	34.46
cond e2eSAE (ours)	**4.11**	**24.92**	**13.50 (±29.82)**	**7.91**
cond DCGAN	0.93	22.23	68.27 (±180.45)	6.95
original data	0.33	0	6.59	2.89


Deep Convolutional GAN

(a)

Variational Autoencoder

(b)

Sinkhorn Autoencoder with NG

(c)

FIGURE 4. TSNE visualisation of generated examples (blue) and original mnist data (red), processed through the LeNet model. Well train DCGAN, without mode collapse, reproduces the whole data distribution well, while VAE additionally produces images from outside of real data distribution (between real classes). Our solution (bottom) generates only examples within true data distribution but has minor problems with reproducing their whole variety.

TABLE 2. Results Comparison on the CelebA Dataset. For Competitive Solutions we Include the Best of Reported Result. Our Solution Outperforms Recent Non-Adversarial Generative Models.

CelebA	
Model	FID
VAE + Flow	65.7
SWAE	64
SAE (H)	56
VAE	55
WAE (MMD)	55
e2e SAE (ours)	**54.5**

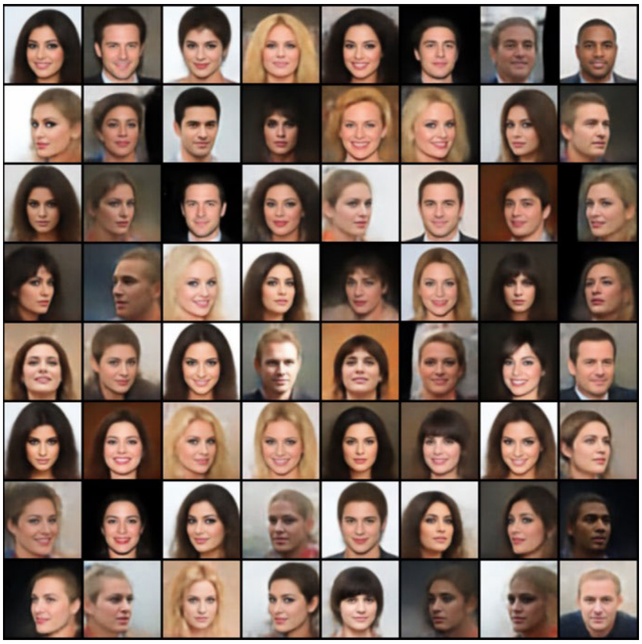


FIGURE 5. Samples of generated images from model trained on CelebA dataset. Our model is capable of generating diverse, high quality images without blurred effect.

our end-to-end Sinkhorn Autoencoder with Noise Generator better captures relations between conditional parameters such as the centre of collision, what is of the special interest in real data simulation.

For MNIST, we also visually analyse coverage of data distribution for evaluated methods. As presented in TSNE visualisation of LeNet penultimate layer activations in Fig. 4, our method has better coverage of original data than other autoencoder based approaches. As depicted in Fig. 4(c), our model does not produce examples outside of original data distribution. On the other hand, results from the well-trained adversarial model (Fig. 4(a)) better overlaps with the full data distribution of the MNIST dataset.

On the CelebA dataset, as demonstrated in table 2, our method outperforms other competitive autoencoder based solutions. As displayed in figure Fig. 5, our end-to-end Sinkhorn Autoencoder generates visually sharp images with high variance.

V. CONCLUSION

In this work, we introduced a new generative model based on autoencoder architecture. Contrary to contemporary solutions, our end-to-end Sinkhorn Autoencoder does not enforce encoding on any parametrised distribution. In order to learn the distribution of standard autoencoder, we converge to it with an additional deterministic neural network, which we train together with an autoencoder. We show that our solution outperforms other comparable approaches on benchmark datasets and the challenging practical dataset of calorimeter response simulations. We postulate that the general approach proposed in this work may also be used with other metrics between probability distribution.

REFERENCES

[1] M. Arjovsky, S. Chintala, and L. Bottou, ''Wasserstein GAN,'' 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875

[2] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, ''Generalization and equilibrium in generative adversarial nets (GANs),'' in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 224–232.

[3] S. Arora and Y. Zhang, ''Do GANs actually learn the distribution? An empirical study,'' 2017, *arXiv:1706.08224*. [Online]. Available: http://arxiv.org/abs/1706.08224

[4] B. O. Ayinde, T. Inanc, and J. M. Zurada, ''Regularizing deep neural networks by enhancing diversity in feature extraction,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2650–2661, Sep. 2019.

[5] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, ''Demystifying MMD GANs,'' 2018, *arXiv:1801.01401*. [Online]. Available: http://arxiv.org/abs/1801.01401

[6] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam, ''Optimizing the latent space of generative networks,'' 2017, *arXiv:1707.05776*. [Online]. Available: http://arxiv.org/abs/1707.05776

[7] M. Cuturi, ''Sinkhorn distances: Lightspeed computation of optimal transport,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2292–2300.

[8] B. Dai and D. Wipf, ''Diagnosing and enhancing VAE models,'' 2019, *arXiv:1903.05789*. [Online]. Available: http://arxiv.org/abs/1903.05789

[9] K. Deja, T. Trzciński, and Ł. Graczykowski, ''Generative models for fast cluster simulations in the TPC for the ALICE experiment,'' in *Proc. Conf. Inf. Technol., Syst. Res. Comput. Phys.* Cham, Switzerland: Springer, 2018, pp. 267–280.

[10] G. Dellacasa and The ALICE Collaboration, ''ALICE technical design report of the zero degree calorimeter (ZDC),'' Tech. Rep. CERN-LHCC-99-05, 1999. [Online]. Available: https://cds.cern.ch/record/381433

[11] L. Evans and P. Bryant, *LHC Machine*, document JINST, 3:S08001, 2008.

[12] J. Feydy, T. Séjourné, F.-X. Vialard, S.-I. Amari, A. Trouve, and G. Peyré, ''Interpolating between optimal transport and MMD using sinkhorn divergences,'' in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 2681–2690.

[13] A. Genevay, L. Chizat, F. Bach, M. Cuturi, and G. Peyré, ''Sample complexity of sinkhorn divergences,'' 2018, *arXiv:1810.02733*. [Online]. Available: http://arxiv.org/abs/1810.02733

[14] A. Genevay, G. Peyré, and M. Cuturi, ''Learning generative models with sinkhorn divergences,'' 2017, *arXiv:1706.00292*. [Online]. Available: http://arxiv.org/abs/1706.00292

[15] X. Glorot, A. Bordes, and Y. Bengio, ''Deep sparse rectifier neural networks,'' in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, pp. 315–323, 2011.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ''Generative adversarial nets,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[17] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, ''A kernel two-sample test,'' *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.

[18] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, ''GANs trained by a two time-scale update rule converge to a local nash equilibrium,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2017., pp. 6626–6637

[19] S. Incerti, I. Kyriakou, M. A. Bernal, M. C. Bordage, Z. Francis, S. Guatelli, V. Ivanchenko, M. Karamitros, N. Lampe, S. B. Lee, S. Meylan, C. H. Min, W. G. Shin, P. Nieminen, D. Sakata, N. Tang, C. Villagrasa, H. N. Tran, and J. M. C. Brown, ''Geant4-DNA example applications for track structure simulations in liquid water: A report from the Geant4-DNA project,'' *Med. Phys.*, vol. 45, no. 8, pp. e722–e739, Aug. 2018.

[20] S. Ioffe and C. Szegedy, ''Batch normalization: Accelerating deep network training by reducing internal covariate shift,'' 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[21] G. R. Khattak, S. Vallecorsa, and F. Carminati, ''Three dimensional energy parametrized generative adversarial networks for electromagnetic shower simulation,'' in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3913–3917.

[22] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[23] D. P. Kingma and M. Welling, ''Auto-encoding variational Bayes,'' 2013, *arXiv:1312.6114*. [Online]. Available: https://arxiv.org/abs/1312.6114

[24] D. P. Kingma and P. Dhariwal, ''Glow: Generative flow with invertible 1x1 convolutions,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10215–10224.

[25] S. Knop, P. Spurek, J. Tabor, I. Podolak, M. Mazur, and S. Jastrzębski, ''Cramer-wold auto-encoder,'' *J. Mach. Learn. Res.*, vol. 21, no. 164, pp. 1–28, 2020.

[26] S. Kolouri, P. E. Pope, C. E. Martin, and G. K. Rohde, ''Sliced-wasserstein autoencoder: An embarrassingly simple generative model,'' 2018, *arXiv:1804.01947*. [Online]. Available: http://arxiv.org/abs/1804.01947

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, ''Gradient-based learning applied to document recognition,'' *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[28] H. Ling and K. Okada, ''Diffusion distance for histogram comparison,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2006, pp. 246–253.

[29] Z. Liu, P. Luo, X. Wang, and X. Tang, ''Deep learning face attributes in the wild,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.

[30] M. Paganini, L. D. Oliveira, and B. Nachman, ''CaloGAN: Simulating 3D high energy particle showers in multi-layer electromagnetic calorimeters with generative adversarial networks,'' 2017, *arXiv:1712.10321*. [Online]. Available: https://arxiv.org/abs/1712.10321

[31] G. Patrini, R. van den Berg, P. Forré, M. Carioni, S. Bhargav, M. Welling, T. Genewein, and F. Nielsen, ''Sinkhorn AutoEncoders,'' 2018, *arXiv:1810.01118*. [Online]. Available: http://arxiv.org/abs/1810.01118

[32] S. Pidhorskyi, D. A. Adjeroh, and G. Doretto, ''Adversarial latent autoencoders,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 14104–14113.

[33] A. A. Pol, V. Azzolini, G. Cerminara, F. D. Guio, G. Franzoni, M. Pierini, F. Sirokỳ, and J.-R. Vlimant, ''Anomaly detection using deep autoencoders for the assessment of the quality of the data acquired by the cms experiment,'' in *EPJ Web Conferences*, vol. 214. Les Ulis, France: EDP Sciences, 2019, Art. no. 06008.

[34] A. Radford, L. Metz, and S. Chintala, ''Unsupervised representation learning with deep convolutional generative adversarial networks,'' 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[35] M. Sakurada and T. Yairi, ''Anomaly detection using autoencoders with nonlinear dimensionality reduction,'' in *Proc. MLSDA 2nd Workshop Mach. Learn. Sensory Data Anal. - MLSDA*, 2014, pp. 4–11.

[36] K. Shmelkov, C. Schmid, and K. Alahari, ''How good is my GAN,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 213–229.

[37] K. Sohn, H. Lee, and X. Yan, ''Learning structured output representation using deep conditional generative models,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.

[38] D. Strulab, G. Santin, D. Lazaro, V. Breton, and C. Morel, ''GATE (geant4 application for tomographic emission): A PET/SPECT general-purpose simulation platform,'' *Nucl. Phys. B Proc. Supplements*, vol. 125, pp. 75–79, Sep. 2003.

[39] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, ''Wasserstein auto-encoders,'' 2017, *arXiv:1711.01558*. [Online]. Available: http://arxiv.org/abs/1711.01558

[40] J. Tomczak and M. Welling, ''VAE with a vampprior,'' in *Proc. Int. Conf. Artif. Intell. Statist.*, Mar. 2018, pp. 1214–1223.

[41] F. Tonolini, B. S. Jensen, and R. Murray-Smith, ''Variational sparse coding,'' in *Uncertainty in Artificial Intelligence*. PMLR, Aug. 2020, pp. 690–700.

[42] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, ''Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,'' *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

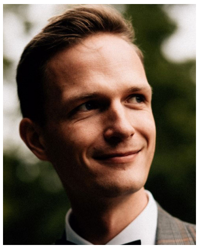
[43] Y. Wang, H. Yao, and S. Zhao, ''Auto-encoder based dimensionality reduction,'' *Neurocomputing*, vol. 184, pp. 232–242, Apr. 2016.

[44] Z. Xiao, Q. Yan, and Y. Amit, "Generative latent flow," 2019, *arXiv:1905.10485*. [Online]. Available: http://arxiv.org/abs/1905.10485

[45] J. Zhu, D. Zhao, B. Zhang, and B. Zhou, "Disentangled inference for GANs with latently invertible autoencoder," 2019, *arXiv:1906.08090*. [Online]. Available: http://arxiv.org/abs/1906.08090

[46] Z. M. Ziegler and A. M. Rush, "Latent normalizing flows for discrete sequences," 2019, *arXiv:1901.10548*. [Online]. Available: http://arxiv.org/abs/1901.10548

[47] A. Zoglauer, R. Andritschke, and F. Schopper, "MEGAlib—The medium energy gamma-ray astronomy library," *New Astron. Rev.*, vol. 50, nos. 7–8, pp. 629–632, Oct. 2006.



KAMIL DEJA was born in Katowice, Poland, in 1994. He received the M.Sc. degree in computer science from the Warsaw University of Technology, Warsaw, Poland, in 2018, where he is currently pursuing the Ph.D. degree.

He is a member of the ALICE Collaboration at LHC CERN.

JAN DUBIŃSKI was born in Warsaw, Poland, in 1995. He received the B.Sc. and M.Sc. degrees in power engineering from the Warsaw University of Technology, Warsaw, Poland, the bachelor's degree in quantitative methods from the Warsaw School of Economics, Warsaw, and the M.Sc. degree in computer science from the Warsaw University of Technology, where he is currently pursuing the Ph.D. degree in machine learning in the future.

He is a member of the ALICE Collaboration at LHC CERN.

PIOTR NOWAK was born in Nowy Sącz, Poland, in 1994. He received the B.Sc. degree in physics from the AGH University of Science and Technology, Cracow, Poland. He is currently pursuing the M.Sc. degree in physics with the Warsaw University of Technology.

He is a member of the ALICE Collaboration at LHC CERN.

SANDRO WENZEL received the M.Sc. and Ph.D. degrees in computational physics from the University of Leipzig, Germany, and the Certificate of Advanced Study in mathematics/theoretical physics from the University of Cambridge, U.K.

After research positions at the Max-Planck Institute for the Physics of Complex Systems, Dresden, Germany, and also at the Ecole Polytechnique de Lausanne (EPFL), Switzerland, where he was a High-Performance Computing Scientist with the Blue Brain project and shortly in the Computer Vision industry. In 2013, he joined CERN, where he mainly works on particle transport simulation and data processing frameworks for the ALICE collaboration.

Dr. Wenzel is a member of the ALICE and GEANT4 collaborations.

PRZEMYSŁAW SPUREK received the master's degree in mathematics from Jagiellonian University, Krakow, Poland, in 2009, and the Ph.D. degree in computer science from Jagiellonian University, in 2014. He is currently an Assistant Professor with the Institute of Computer Science, Jagiellonian University.

TOMASZ TRZCIŃSKI (Member, IEEE) received the M.Sc. degree in research on information and communication technologies from the Universitat Politècnica de Catalunya, the M.Sc. degree in electronics engineering from the Politecnico di Torino, in 2010, and the Ph.D. degree in computer vision from the École Polytechnique Fédérale de Lausanne, in 2014.

His professional appointments include Telefónica R&D in 2010, Qualcomm Corporate R&D in 2012, and Google in 2013. In 2017, he was a Visiting Scholar with Stanford University. He is currently an Assistant Professor with the Division of Computer Graphics, Institute of Computer Science, Warsaw University of Technology. He is also a Chief Scientist and a Partner with Tooploox Sp. z o.o., a software services company with more than hundred people on board, where he leads a team of machine learning researchers and engineers.

Dr. Trzciński is a member of the Computer Vision Foundation and the Scientific Board of the PLinML Conference. He is currently an Associate Editor of IEEE Access and frequently serves as a Reviewer of major computer vision conferences including CVPR, ICCV, ECCV, ACCV, BMVC, ICML, and MICCAI, and international journals such as IEEE Transactions on Pattern Analysis and Machine Intelligence, *IJCV*, *CVIU*, IEEE Transactions on Image Processing, and IEEE Transactions on Multimedia.