Navigation of Extrinsic Factors for Reducing the Uncertainty of Intrinsic Factor Estimates

İşler, Berk

Award date:
2020

Department of Mathematics and Computer Science
System Architecture and Networking Research Group

Navigation of Extrinsic Factors for Reducing the Uncertainty of Intrinsic Factor Estimates

Master Thesis

Berk Isler

Supervisors:
University Supervisor: Mike Holenderski
Industry Supervisor: Luis Armando Perez Rey
Committee Member: Dmitri Jarnikov
Committee Member: Vlado Menkovski

Eindhoven, October 2020

Preface

The past two years have been very challenging times for me, as a mechanical engineer who found himself in the world of Data Science. As I am concluding the journey of my master's degree with this thesis, I think I can say I have come a very long way. I was expecting that the preface of my thesis would be by far the easiest to write. However, it turned out one of the hardest, especially when one has so many great people to thank for their invaluable help throughout this two-year adventure.

To begin with, I would like to thank my supervisors Mike Holenderski and Luis Armando Perez Rey. I have learned so much from both of you about Data Science, but most importantly about scientific perspective and mindset. I am truly grateful for your guidance and effort to push me to achieve my best. I would like to further thank Vlado Menkovski and Dmitri Jarnikov for their valuable participation in my defense committee and once again many thanks to Dmitri Jarnikov for his valued supervision.

I would like to give my most sincere appreciation to Gökhan, Selima, Sonali, and my dear neighbor Andrea who became my family in the Netherlands. I cannot describe enough, your support and help in those never-ending library days. I owe all of you many thanks for always being there for me and also sharing those amazing moments with me when we had the most fun. I am sure we will have much more great time together, the best is yet to come.

I would like to offer my thanks to my colleagues from Digiterra for their understanding and support since the first day I started my master's degree. I would like to especially thank Bahadır Balkır who believed in me and gave me a chance to prove myself.

Finally, my greatest thanks to my parents and my girlfriend. Irmak, my girlfriend, has been my biggest support during the thesis process with continuous encouragement. Thank you for all the sacrifices you made to stay with me, I can't even imagine how could I complete this journey without having you by my side. My parents were the driving force in those tough times, that kept me going. You always unconditionally loved me and supported me with your utmost trust for all the decisions I make. I cannot describe with words, how thankful I am to have you as my parents.

Berk Isler
Eindhoven, 2020

CONTENTS

List of Figures

List of Tables

Navigation of Extrinsic Factors for Reducing the Uncertainty of Intrinsic Factor Estimates

Berk Isler

ABSTRACT

Evaluation of the uncertainty of the model's predictions is important for black-box models, such as deep neural networks. Hence, there has been an intensive effort to be able to capture the uncertainty, in other words allowing the model to be aware of the cases when the predictions have high uncertainty. However, there has been limited work to reason about or even act upon the prediction uncertainty. Active learning is one example where the prediction uncertainty is used to make manual labeling more efficient. In this thesis, we define a method to identify underlying reasons of uncertainty and obtain insights from it such that we can reduce the uncertainty of the predictions. Given an object to perform predictions on, intrinsic factors denote its characteristic properties (color, shape, texture) whereas extrinsic ones denote the external factors (lighting, orientation) that the object is independent of, yet affect how the intrinsic factors are perceived. We defined our task to estimate an intrinsic property of the object of interest in an environment where we have control over extrinsic factors. We propose to manipulate the extrinsic factors using a mobile agent to collect observations of the object. The final objective is to improve intrinsic factor classification performance. This is done by using uncertainty as guidance to efficiently sample the data space in order to improve prediction accuracy during inference time. The idea is to focus on ambiguous cases with low certainty and propose more favorable extrinsic factor configurations such that we can classify with higher certainty. The experiments show that we can obtain an increase of f1-score of about 6.8% without any additional training.

1 INTRODUCTION

1.1 Motivation

Currently, fields of Machine Learning (ML) and Artificial Intelligence (AI) are attracting a great amount of attention. Thanks to the efforts of the research community, ML models have reached the capability of performing many tasks close to or even better than the human level. Hence, ML models have been deployed in many real-life decision-making processes. Moreover, to maintain transparency in such cases, the interpretability of the model's outcome is essential which is hard to obtain with black-box models. Despite their success due to huge parametric search space and efficient learning algorithms, Deep Neural Network (DNN) is an example of the black box models [1]. This kind of opaque behavior is detrimental for the applications in social environments where human life and security can be directly affected [2].

Explainable AI (XAI) is the sub-filed of AI, that focuses on explaining the causal link about the results and inner mechanism of the models. According to [14], XAI aims to *'produce more explainable models while maintaining a high level of learning performance (e.g., prediction accuracy), and enable humans to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners.'*. As emphasized in the definition, the efforts towards the XAI should maintain transparency about the decisions of the ML models as well as an adequate learning performance for the task at hand. As stated in [29], there is an inverse relationship between the capacity/performance and the opacity of the model. Even though reducing model capacity can be used to increase the model interpretability, these solutions would also limit the success of the predictions of the AI/ML systems.

In AI/ML community, uncertainty can be used to capture the confidence of the models' predictions [1]. The prediction **uncertainty** is one of the aspects of the models that would indicate the trust and interpretability of the model's outcome. The uncertainty of the model is considered as a fundamental element of all ML models, especially in domains where safety issues are critical [19]. To emphasize the importance of considering model confidence, the following example can be given. The accident of an autonomous vehicle with a truck that results in a causality is described in [34]. The main cause of the accident is attributed to the misinterpretation of the white side of the truck as a clear sky. This accident may have been avoided if the system were aware of the uncertainty for predicting the side of the truck as the sky and notify the driver of low certainty.

Apart from AI safety issues, evaluation of uncertainty is fundamental for numerous of ML approaches. In *active learning*, model decides **actively** which data points to label in situations where annotating is costly. The model seeks for examples that would be the most informative for the task at hand, by focusing on those with the highest prediction uncertainty. *Active inference* describes the methodology to allow an agent to make decisions under uncertainty in a dynamic environment [32]. The agent has a generative model to infer the world in terms of the hidden states that describe the current condition of the environment. These states are then evaluated to decide on the next action. Furthermore, the agent behaves in the environment (characterized by its states), in a way that minimizes the unexpected observations, in other words, the *surprise* [8]. The notion of surprise captures the uncertainty about the environment. Thus, the model is trained to jointly optimize the perception and action in a Bayesian fashion to minimize the surprise.

1.2 Problem Definition

The process of vehicle evaluation is an important step in the second-hand vehicle trading business. Potential buyers want to make sure the vehicle meets certain standards. These standards can vary from more fundamental aspects such as safety and moving parts to more aesthetic ones as bodywork. The former ones have better-established ground-truth values, whereas the latter ones are less definitive in terms of optimal conditions. Hence, the inspection of aesthetic aspects requires more human interpretation and thus automated techniques are less applicable for them.

Some of the available automated solutions use multiple cameras with fixed locations. These cameras take pictures of the vehicles from various angles to capture the maximum amount of information about the car. However, to ensure a certain level of success, these methods require controlled environments where external conditions such as lighting conditions or surroundings of the car are standardized. Thus, due to the limited flexibility, these methods cannot offer scalable solutions, considering numerous inspection locations. Therefore, a more complete solution should be able to adapt various external factors in the scene to perform the prediction task with the least number of observations (efficiently).

To define it more formally, our ultimate goal is to make predictions about some visual property of the car with utmost accuracy and certainty. Hence, we need to somehow capture the maximum amount of information about that property of the car. Given that the prediction task is about visual properties, an intuitive way to collect information is by taking its pictures. The features extracted from an image of the scene can be subdivided into extrinsic (f_e) and intrinsic (f_i). Intrinsic factors describe the vehicle and its characteristic information, such as the geometry (shape), color, or texture. Extrinsic factors, on the other hand, describe the ambient factors of the scene such as the lighting conditions, shadings, and orientation of the camera w.r.t the vehicle. Essentially the task of visual vehicle evaluation depends on the intrinsic factors of the image. Hence, our aim is to estimate the intrinsic factors of the car to aid its visual evaluation.

We use the term *observation* to denote a particular image of the car. The configuration of the scene defines the values of the extrinsic factors that lead to a specific observation. Observations of a car vary depending on the scene configuration that defines it. Since the possible number of observations is excessive, we cannot try all of them during the visual evaluation of a car. One intuitive approach can propose to evaluate all observations, given a scene, before the visual evaluation. Then, we can perform the visual evaluation with the observation, described by a scene configuration, that always provides the highest information about any car. However, given the broad diversity of the cars, we cannot expect a single scene configuration with maximum informativeness for any car. Hence, we need to know dynamically which observations are the most informative for a given car. However, this requires a way to measure the amount of information held by an observation.

In our case, prediction uncertainty is used to assess the information gain that can be obtained from observation. To be more precise, we can use prediction uncertainties of observations from the scene, such that we can detect observations with minimum uncertainty and maximum informativeness. At this point, the notion of uncertainty assessment can be refined. In general, the sources of uncertainty within an ML model can be attributed to two main components as *aleatoric* and *epistemic*. Aleatoric uncertainty refers to the uncertainty inherent to the data generation process whereas epistemic uncertainty denotes the model uncertainty [20]. In other words, aleatoric uncertainty cannot be reduced by refining the model or introducing more data. On the other hand, epistemic uncertainty is essentially ML models' inefficacy to capture the generative process of the data and can be reduced with additional information such as more data points. Hence, our focus is on epistemic uncertainty such that we are trying to detect the most informative observations which reduce the uncertainty of our predictions due to epistemic sources.

Due to rich and complex interactions in the scene, extrinsic factors may affect the perception of the intrinsic factors. In particular, this behavior can lead to changes in terms of resulting uncertainty when extrinsic factors are altered. Hence, being able to understand the relationship between the uncertainty of an intrinsic factor estimate and extrinsic factors would allow us to control and improve the certainty. More formally, we can define the research questions as follows:

- How can we capture the information about individual intrinsic and extrinsic generative factors independently?
- How can we quantify the uncertainty on the prediction of intrinsic factors?
- How can we assess to which degree an extrinsic factor contributes to the intrinsic factor prediction uncertainty?
- How can we improve the performance of intrinsic factor prediction by manipulating the extrinsic factors to reduce the uncertainty of the prediction.

1.3 Proposed Solution

Having stated the research questions, we defined three sub-steps to be able to address them.

In the context of this study, we assume we are collecting the observations with a mobile agent. This allows us to manipulate individual extrinsic factors separately. In particular, we obtain the representation of an observation where the information about each extrinsic factor is separated. Hence we can evaluate the extent they contribute to the prediction uncertainty of the intrinsic factor. Consequently, the agent can specifically act upon the most responsible extrinsic factors to reduce uncertainty. In the first step of our method, we focus on this aspect using the notion of disentanglement.

In the second step, we propose to perform intrinsic factor prediction using the latent representations of the observations. We defined the intrinsic factor prediction task as multi-class classification of the cars into different *types* based on their shape. This particular choice was based on the data set we used and we will further reason about it in section 3.2. Subsequently, we will use the classification results as the predictive distribution and estimate corresponding uncertainty using the *entropy* measure. It should be noted that to use soft-max outputs as prediction probabilities, additional measures should be performed, which are explained in detail in section 3.2.

In the third step, assuming disentanglement, we use the error signal of intrinsic factor prediction to update the representation in the latent space for each extrinsic factor. In this thesis, the term *transformation* denotes the change in the value of an extrinsic factor. Hence, the updates of the representation can be seen as transforming it in terms of extrinsic factors in the latent space. Consequently, we evaluate the extrinsic factors in terms of their capability to reduce uncertainty, using their respective transformed representation. Finally, we map the transformations in the latent space to the new value of the corresponding extrinsic factor in the scene. These updated extrinsic factors describe the new scene configuration for the next observation with lower uncertainty that is obtained using the agent.

1.4 Contributions

The aim of this thesis to develop a model that guides an agent in an environment to perform a prediction task on a vehicle. Particularly the model can reason about the root causes of prediction uncertainty in terms of extrinsic generative factors of the scene using the disentangled representation of the environment. Furthermore, the model can suggest new values on the extrinsic factors such that resulting observations would lead to lower uncertainty. Hence, the model could use the ability to navigate an agent in an uncertainty-aware manner to perform the prediction with the least number of observations while achieving high prediction accuracy and low prediction uncertainty.

- We associate the uncertainty of the predictions to the extrinsic generative factors such that main contributing factors can be identified.
- We defined a novel pipeline, that identifies scene configurations for the next observation that would lead to low prediction uncertainty.
- We generated a data set with synthetic images, where the images have known generative factors with correct ground truth values.

2 Literature Review

2.1 Disentanglement

Obtaining disentangled representations, that capture distinct sources of variation independently, is an important step towards human-level AI/ML systems [21]. Despite the lack of agreement on the definition, one description states that a disentangled representation should separate the distinct, informative factors of variations in the data [3]. There has been a considerable effort on that matter. These efforts can be investigated as two main approaches namely, *supervised* and *unsupervised*.

In the scope of disentangled representation learning, unsupervised methods try to achieve disentanglement without any guidance about the actual factors of variation [30]. It is further motivated by human's ability to learn these factors in an unsupervised manner and the inconsistency and difficulties of providing labels [21]. Within Deep Generative Models (DGM), this is done by using neural networks to approximate a conditional distribution on the data. Particularly VAEs are heavily favored due to their ability to model a joint distribution while maintaining scalability and training stability [18]. Therefore most of the methods are based on augmentations on original VAE framework [18] [6] [21] [28].

The authors of [18] are one of the first to propose to change the standard *vanilla-VAE* objective by introducing additional regularization term. Particularly, they put additional weight term to KL divergence in the VAE loss. Hence, they force the approximate posterior to be factorized, which leads to better disentanglement. As [18] shows a promising approach to the disentangled representation learning problem, various other papers follow the same principle of augmenting VAE objective. Both [6] and [21] argued that the penalty term on the KL divergence term in equation 2 is causing the low reconstruction capability and proposed to further decompose the KL divergence term and identify underlying sources of disentanglement. In [6], KL divergence term of original VAE objective is decomposed into mutual information, total correlation and dimension-wise KL divergence. Consecutively, they declared a total correlation term as the most crucial element of disentanglement and hence proposed to penalize this term with additional weight term. Similarly, in [21], they define composition in a more implicit way, where total correlation and dimension-wise KL divergence expressed as one term and proposed to use additional weight for this term.

Despite improved disentanglement offered by unsupervised methods, in [28], the authors showed that obtaining disentangled representation in an unsupervised way is theoretically not possible. Moreover, they stated supervision is essential for disentanglement. The methods described in [9], [17], [26] and [37] use supervision imposed by either the training procedures or through the labels. The method described in [17], disentangle the factors corresponding to ambient values (extrinsic) and the factors denoting identity information of the object (intrinsic) in the image that is invariant ambient values. It is achieved by inferring two sets of latent variables by a VAE and using additional loss terms to define constraints on a subset of latent factors. In [9] again similar sets of factors are disentangled, namely class and content that corresponds to class-specific identity information and the aspects that can vary within the same class respectively. The proposed training procedure defines two separate latent spaces for each factor. These latent spaces are jointly trained using reconstruction error between sequence of content varying images and reconstructions of encodings sampled from these latent spaces. Both of the works defined in [37] and [26] exploit the data that describes the various transformations. They regularize the latent space such that a particular subset of latent factors shows equivariance to those transformations. In [26], this is achieved by defining linear transformations for various transformations and using them to manipulate the latent space such that its reconstruction would approximate the image with true transformation. On the other hand, in [37] two sets of latent factors are defined as identity and pose. For a fixed identity unit they apply linear transformations to the pose unit in a recurrent fashion, which are then combined with the identity unit to reconstruct the sequence of transformed images.

2.2 Uncertainty Estimation

Estimating uncertainty of the prediction of the ML models is attracting interest increasingly. In general, the success of ML models is assessed mainly based on the correctness of the predictions, such as *accuracy* or *f1* score for classification models. Yet, these metrics fail to reveal the uncertainty of predictions.

Bayesian Neural Networks (BNN) provide an inherent solution to assess prediction uncertainty due to their probabilistic aspect. BNNs are differentiated from regular NNs by how the weights of the network are defined and learned. Instead of learning weights deterministically with pointwise estimates, BNN assigns a prior distribution over the possible values that the weights can take and try to refine these distributions using Bayesian inference. Precisely, for a data set D with inputs $X = \{x_1, x_2, .., x_n\}$ and labels $Y = \{y_1, y_2, .., y_n\}$, the posterior distribution over the set of possible weights W is described by Bayes theorem as $p(W|D) = p(D|W)p(W)/p(D)$. The posterior distribution denotes the family of possible model weights, given the data set [20]. The prediction of a probabilistic classifier can be expressed as the conditional probability that depends on the chosen weights and the input image ($p(y|x,w)$). On the other hand, for a test input, x^* with true label y^*, the final prediction of a BNN is computed by marginalizing the weight term from the likelihood. In other words, the expectation of the likelihood term over the posterior distribution gives the final prediction, $p(y^*|x^*,D) = \mathbf{E}_{p(W|D)}[p(y^*|x^*,w)]$. The uncertainty can be computed by a discrepancy measure such as *variance* or *entropy* over the predictive distribution, $p(y|x)$. However, it should be noted that the posterior evaluation is intractable. Hence, it is approximated with various techniques.

A group of techniques proposes to use variational inference to approximate the intractable posterior. The general idea is to define an approximate posterior $q'(w)$ from a family of distributions over the possible W and then minimize the KL divergence between approximate and true posterior. Thus, the problem can be stated as an optimization task, where the parameters of the approximate distribution are optimized w.r.t. KL divergence. Some examples of variational inference applied to BNN are found in [13], [4], [11] and [35].

Non-Bayesian approaches also gained popularity for the estimation of prediction uncertainty. This group of methods focuses on Monte-Carlo (MC) sampling. Specifically, the prediction is performed with an arbitrary number of models with weights sampled from all possible weights described by its distribution. The final prediction is obtained by averaging the results and the uncertainty of prediction estimated by a discrepancy measure of this series of predictions. The ensemble learning is the technique where the results of multiple models are combined and used as the final decision. The MC sampling technique can be interpreted as an ensemble method since the averaging of predictions is performed using an ensemble of neural networks [25]. The work presented in [12], uses dropout as a method to approximate Bayesian inference. The authors apply dropout also during test time and disable a subset of the network according to Bernoulli distributions defined for each weight. For a test input, they perform multiple stochastic forward passes and average the results. Since at each pass, the model is randomly altered, this can be seen as an ensemble method [27]. As another ensemble method [25], defines multiple instances of a model with random initialization. The networks are trained independently on the full data set and then the final prediction is obtained by averaging the resulting predicted probability of each model.

To summarize, the methods we discussed for both disentanglement and uncertainty estimations have some shortcomings for our requirements. In [13], [4] disentanglement is achieved only between intrinsic and extrinsic set of factors without further separating each set. On the other hand, the methods described in [26] and [37] do not have axis alignment between generative and latent factors, which means the latent factors are not designated to encode a particular generative factor. However, we need to disentangle individual generative factors for both intrinsic and extrinsic sets, where each extrinsic factor has a corresponding latent factor.

For the uncertainty estimation BNN techniques [13], [4], [11] and [35] require substantial modifications to the training procedure which leads to computationally expensive solutions with slower training [19]. Moreover, during test time for a single input, multiple predictions are needed to be performed to be able to estimate the uncertainty. Although ensemble methods [25] and [12] overcome the computational challenges, they also require multiple predictions during test time inference of the model. This scales the computation time with the number of predictions which could be problematic for real-time applications.

3 Methodology

We start this section, explaining how we achieve the disentanglement of individual intrinsic and extrinsic factors within the representation of observations. Then, we describe the methodology to quantify the uncertainty of intrinsic factor prediction. We justify our choice to quantify the uncertainty. Finally, we specify the details of the inference pipeline. In particular, we explain how we evaluate the extrinsic factors based on the extent they contribute to the prediction uncertainty. Moreover, we describe how we propose the new extrinsic factor values, that underlies the new scene configuration for the next observation. Note that we used the term *inference* to refer to the model evaluation during test time as in the deep learning literature.

3.1 Disentanglement (DC-IGN)

Having a disentangled latent representation of the scene, where the agent act upon, is a crucial requirement for the proposed solution. Essentially, we are trying to capture the disentangled generative factors of the scene. In the context of this thesis, we define the generative factors as the combination of two subsets, namely *intrinsic* and *extrinsic* factors. Besides their semantic difference, one can also highlight their difference with the notions of **equivariance** and **invariance**. A transformation of the 3d scene space wrt to the camera or ambient factors would also transform 2d image space. Consequently, the latent space would also be transformed and some aspects of it would be altered. The set of factors of the scene that show immutability and kept unchanged for a given transformation are said to be invariant to that transformation. Equivariant ones, on the other hand, denote the factors that are modified in the latent space linearly to the transformation exerted on scene space. The transformations we can perform in the scene consist of the extrinsic factors, such that we can alter the position of the camera and the ambient factors of the environment. Accordingly, the intrinsic factors should have invariance for the transformations of extrinsic factors. This behavior can also be described with the following equations where I denotes the observation, $h_{int}(I)$ and $h_{ext}(I)$ denote the intrinsic and extrinsic factors of observation I respectively, and $g()$ denotes the extrinsic transformation.

$$\begin{aligned} h_{int}(g(I)) &= h_{int}(I), \\ h_{ext}(g(I)) &= g(h_{ext}(I)). \end{aligned} \tag{1}$$

To obtain the disentangled latent representations, we followed the method introduced in [24]. They also use the idea of equivariance/invariance in their method to achieve disentanglement. Moreover, they managed to disentangle not only extrinsic and intrinsic factors but further disentangle individual generative factors. In other words, the training procedure they defined allows us to decide which particular factors we encode in the latent space in a factorized way. Essentially, they used a convolutional variational auto-encoder (CVAE) network for which the training is performed via the VAE objective, namely by maximizing *ELBO*. ELBO is defined in equation 2, where $q_\phi(z|x)$ denotes approximate posterior distribution and $p_\theta(x|z)$ denotes generative distribution parameterized by encoder and decoder networks of VAE. More precisely, the first term in equation 2 is the expected value of log-likelihood of input data (x) given the latent variables (z) and the second term is the KL divergence of approximate posterior and the true prior distribution of latent variables.

$$\mathbf{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)) \tag{2}$$

There are two main traits of the training procedure used in dc-ign, that leads to disentangled representations. Firstly, the network is trained with mini-batches of data, where each mini-batch consists of a particular set of images. The data in each mini-batch are the sequence of images, that vary in terms of only a single generative factor, while the remaining factors are kept constant. With a mini-batch, the model receives a sequence of images that describe the differences in the scene when the value of a generative factor is consecutively altered.

In general, the goal is to represent generative factors by disjoint sets of latent factors such that each set encodes information about a particular generative factor throughout the training. During training, given a mini-batch of a generative factor, its respective latent factors are called *active*, while the latent factors belong to other generative factors are called *inactive*. To achieve that, as the second trait, the flow of information is modified in the latent space during both forward and backward pass. For a mini-batch with a particular generative factor, during the forward pass, all inactive latent factors are altered. Particularly, each of the inactive latent factors is replaced with their respective mean computed over the mini-batch, whereas the active latent factors keep the values inferred from the encoder. Since the inactive factors of the latent representations in a mini-batch are equalized, they don't provide any information about the variations within the batch. Thus, the decoder network is forced to reconstruct the variations within a mini-batch, using only the information from active latent factors. Similarly, in the backward pass, the gradient signal received by the latent factors is modified. Again, only the active factors are allowed to be updated with the true error signal. For the inactive factors, each error signal is replaced with the difference of its inferred latent value (z_i) and the respective mean values computed during the forward pass. Hence, the encoder network would learn to represent the variations within the batch using only the active latent factors.

Consequently, with the described training procedure, we try to achieve equivariance between the transformation of a generative factor described within a mini-batch and the latent factors specified for that generative factor. On the other hand, the latent factors, that do not belong to the generative factor of the mini-batch, forced to be invariant with respect to the transformation. Figure 1, describes an example training procedure for a mini-batch whose generative factor is attributed to the first factor of the latent representation.

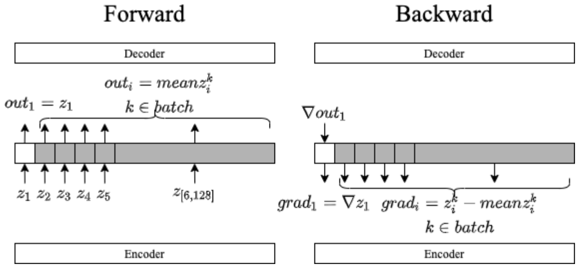

Figure 1: The training procedure of dc-ign [24]. The unshaded and shaded latent factors denote the active and inactive factors respcetively. This figure demonstrates training with a mini-batch of k images where only the first latent factor is set as active.

For our experiments, we defined six generative factors, *light intensity*, *light location*, *elevation* and *azimuth* angles of the camera w.r.t the car as extrinsic ones whereas *color* and *model* of the car as intrinsic ones. We used latent factors z_1, z_2, z_3, z_4, z_5 to encode generative factors of *light intensity*, *light location*, *elevation angle*, *azimuth angle* and *color* respectively, whereas the remaining latent factors encode the car model. Although, we would not perform transformations for intrinsic factors, there are latent factors in the learned representations, that shows equivariance with intrinsic transformations. However, having latent factors designated for the car model is essential since the defined task of estimating *car type* is directly related to the geometry, hence to the model of the car. On the other hand, the color of the car is chosen to demonstrate the difference between extrinsic and intrinsic factors in terms of disentanglement performance.

3.2 Uncertainty Estimation of Intrinsic Factor Prediction

The main prediction task, for which we try to estimate and reason about its uncertainty, is defined as *car type* prediction. To be more precise, we try to classify the cars into different types that vary based on the geometry and dimension of the car. Although this is different from the actual task of visual inspection, we believe car type is a good option to demonstrate to the usefulness of our method for visual intrinsic factors prediction. This particular choice is due to the lack of labeling of the car model intrinsic factor for the data set we generated. Consequently, car type labeling is performed manually by considering the general visual features of the cars such as the shape and dimension of the cars.

As we defined the intrinsic prediction task as a multi-class classification in section 1.3, we used a regular neural network as the classifier model. As motivated in section 2, described methods introduce additional time requirement during test time. In our setting, this is detrimental even further, since, in our inference pipeline, we need to perform predictions one for original input and one for each extrinsic factor. Considering, we are required to perform the predictions in real-time, the BNN and ensemble methods would overburden the computation time.

Consequently, for the intrinsic factor prediction task, we propose to use the softmax output of the model as the predictive distribution. This choice should be justified since the softmax output of a NN might produce over-confident predictions [15]. For classification problems, the confidence of a prediction is the highest probability in the softmax output. Having calibrated confidences refers that the confidence of the prediction is indicating the actual likelihood that the prediction is correct [5]. In other words, well-calibrated prediction confidence should reflect the empirical likelihood of the predicted class [10].

It is argued that the depth of the network has a direct effect on miscalibration [15]. Hence, we defined our intrinsic factor classifier as a simple and shallow model. We used the latent representations as input to the classifier rather than actual images. This allowed us to exploit the information in the disentangled intrinsic latent factor *car model* and achieve decent classification performance with simpler and confidence calibrated models. While deciding model parameters such as the number of layers and number of neurons, we used *expected calibration error (ECE)* [31]. This metric measures the expected error between the prediction confidences and the true class probabilities.

To compute ECE, the confidences of the predictions over a set of data, with N observations (images), are divided into M bins, where the confidences within bin m fall into interval of $(\frac{m-1}{m}, \frac{m}{M}]$. The observations in a particular bin m are denoted by B_m. The accuracy and confidence of B_m are computed as following where y_i, $\hat{y}_i$, and p are the true label, predicted label, and confidence of the prediction respectively.

$$\begin{aligned} acc(B_m) &= \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i) \\ conf(B_m) &= \frac{1}{|B_m|} \sum_{i \in B_m} (p_i) \end{aligned} \tag{3}$$

Having defined the accuracy and the confidence of a bin, ECE is approximated as the following

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |acc(B_m) - conf(B_m)| \tag{4}$$

Assuming we have a classifier with well-calibrated prediction confidences, we estimate the prediction uncertainty by computing *entropy* value, given in equation 5, over the soft-max output (S).

$$H(S) = -\sum_{p_i \in S} p_i \log p_i \tag{5}$$

For a soft-max output where all classes have the same probability would maximize the entropy. On the other hand, for confident predictions where the probability of the predicted class is higher than the remaining classes, we would obtain lower entropy.

3.3 Inference Pipeline

The inference pipeline describes the method that evaluates the prediction uncertainty of the intrinsic factor for observation and describes the new scene configuration for the next observation. It is expected that the next observation leads to more confident predictions. Besides exploiting uncertainty for this decision making, we also want to improve the prediction accuracy. In that sense, we are assuming the classifier is trained enough, such that confident (low entropy) estimates would only be obtained by correct classifications. Hence, we should expect an increase in the prediction performance as we try to increase the confidence of the predictions. In other words, the main idea of the inference pipeline is as following; given the data distribution, we used to train the intrinsic factor classifier, it essentially describes space of scene configurations where we are trying to navigate the agent. We are trying to tailor our movements such that we detect the places in that space, where predictions are confident and hence correct with the least amount of steps. It should be noted that the structure of this space might vary depending on the car type. That means for different car types, the scene configuration of high confidence can be different. Thus the proposed method should adapt these variations.

The working mechanism of the pipeline is depicted in figure 2. Particularly, for an initial observation, we start by obtaining the disentangled latent representation using the encoder of the dc-ign model. This representation is fed to the calibrated intrinsic factor classifier to obtain its softmax output. We assess the prediction uncertainty of this initial observation by computing the entropy over its softmax distribution. Based on the entropy value, the gradient values are computed and back-propagated to latent representation. Assuming the extrinsic factors are disentangled, the gradient values for each of them would indicate the share they have for the uncertainty. At this step, we scale the error signal of the gradient with the *learning rate*. This value denotes the step size we apply to the gradient values. We used the term learning rate since that is one common name for that parameter in ML. Given the gradient vector, we separately update each latent factor that corresponds to true extrinsic factors by their respective gradient value. At this point, we obtain transformed latent representations that correspond to the observations with new scene configuration.

These transformed representations are used one by one to perform intrinsic factor prediction again with the calibrated classifier, followed by entropy calculation on their softmax outputs. The extrinsic

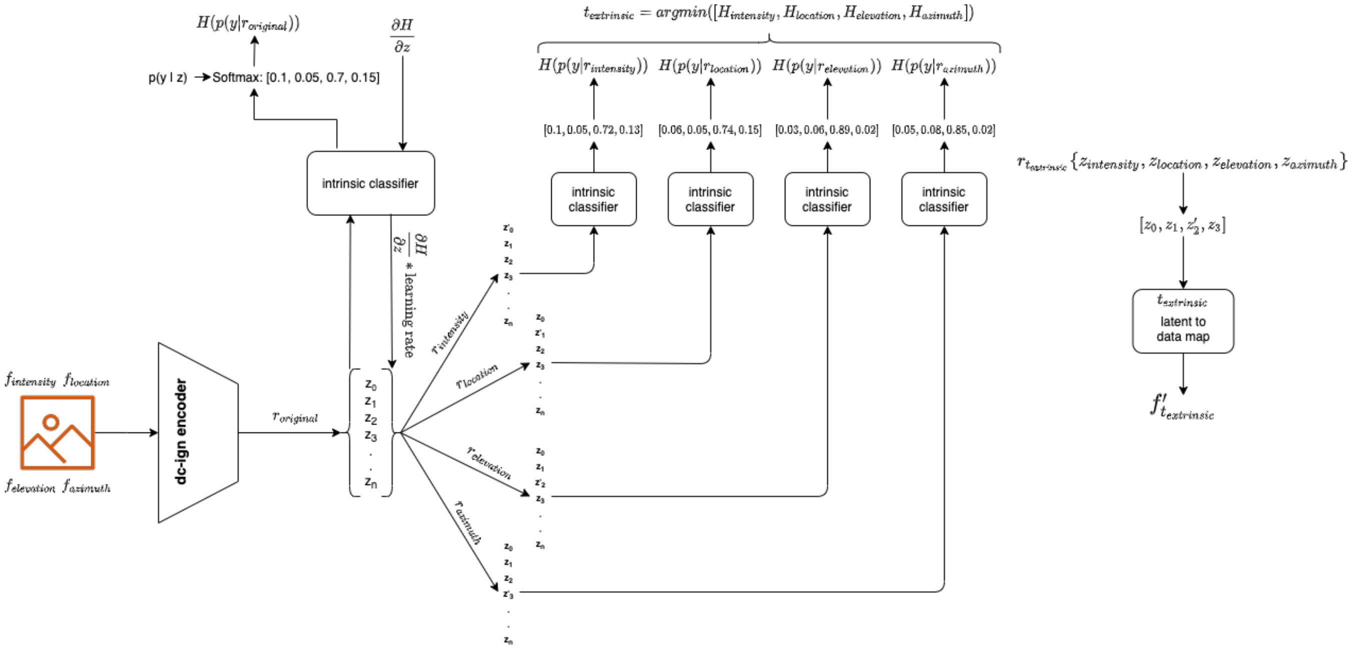

Figure 2: Inference Pipeline: The diagram depicts an example where elevation is found as extrinsic factors to be altered. Particularly, $f_{intensity}$, $f_{location}$, $f_{elevation}$, $f_{azimuth}$ denotes the true generative factors of the test image for which the latent representation, $r_{original}$, is obtained. $r_{intensity}$, $r_{location}$, $r_{elevation}$ and $r_{azimuth}$ are representing the transformed representations. Furthermore, $t_{extrinsic}$ denotes the chosen extrinsic factor for which the new proposed value ($f'_{t_{extrinsic}}$) is found using the disentangled extrinsic latent factors $r_{t_{extrinsic}}\{z_{intensity}, z_{elevation}, z_{elevation}, z_{azimuth}\}$

factor whose softmax gives the smallest entropy is chosen as the factor that would reduce the uncertainty the most. Notice that, since we used the gradient values to update the original representations, it is certain that each of the transformed representations would result in lower entropy than of the original observation.

So far, we have decided which extrinsic factor to manipulate in the scene. However, we still need to tell the new value of the chosen extrinsic factor, corresponding to the update performed on it. To perform that we have defined an additional classifier, that maps the latent value of a factor to its value in the data space. Hence, we called them as *latent to data maps*. We defined a separate mapping classifier for each extrinsic factors and trained them with the latent factors, that correspond to extrinsic factors, as input. In other words, we feed the factors of the latent representation, that are designated for the extrinsic factors to the mapping classifier. Given the extrinsic latent factors, the mapping classifier estimates the corresponding true factor value. Thus, also the new scene configuration for which the agent obtains the next observation to perform inference.

Having described the main working principle of the inference pipeline, we have further defined other variations of the pipeline. Essentially, we have three main decision points, that are *transformation technique*, *number of inference steps* and *stopping criteria*.

Transformation technique characterize decisions about how the extrinsic factors are transformed and consequently how the new scene configuration is determined. *Regular inference* follows the above-described procedure where for a single observation, only one extrinsic factor is chosen and used to define the change in the configuration of the scene. The new value of the extrinsic factor is decided based on its respective latent to the data map. *Multiple transformation inference* adopt a very similar approach where, instead of focusing on a single extrinsic factor, the transformation in terms of all extrinsic factors are considered at the same time. That means the inference pipeline can propose a new scene configuration where multiple extrinsic factors are altered. Again, the new values for all extrinsic factors are decided based on their corresponding latent to data map. Besides these transformation strategy alternatives, we defined two intuitive alternatives. With *best of each inference*, we allow again only one extrinsic factor to be transformed at a time. Similar to other alternatives, the extrinsic factor is chosen based on the estimated entropy reduction. However, instead of using latent to data map, for each extrinsic factor, the value that has the lowest mean entropy in the training set is assigned as the new value of the chosen factor. We identified the values with the lowest mean entropy for each extrinsic factor, based on the prediction uncertainties of the training set. Lastly, for *best of all inference*, we identified the scene configuration that has the lowest entropy among the entire training set. Regardless of the evaluation of the extrinsic factors, this strategy always proposes the scene configuration as the one with the minimum.

Number of inference steps describes whether consecutive inference steps are allowed. In *single step inference*, there can be at most one pass of inference pipeline, whereas with *multi step inference*, we can keep applying the inference pipeline consecutively such that observation from the proposed scene configuration of the previous step, used as input of the next step. Intuitively, it only makes sense to use the proposed configurations as long as they lead to an increase in confidence.

Stopping criteria defines the conditions whether the result of the inference pipeline should be performed in the scene via the agent. The first intuitive condition is to check if the inference pipeline suggests a new scene configuration. If the latent to data maps do not output a different value for the chosen extrinsic factors than the ones of the input observation, the agent cannot alter the scene configuration. Hence, we stop the inference. Secondly, we assess the usefulness of a proposed transformation in the sense that whether the observation with the proposed scene configuration has lower uncertainty.

In *latent space strategy*, we compare the prediction uncertainty of input observation with the uncertainty of the prediction obtained by transformed *latent* representation. If the intrinsic factor prediction using the transformed representation has lower uncertainty, then the inference pipeline returns corresponding uncertainty and predicted class as the current uncertainty and prediction respectively. Notice that with this strategy, since we evaluate the uncertainty reduction in the latent space, we can assess the usefulness of new scene configuration without actually performing the transformation. However, the latent to data map may estimate the new value of the transformed extrinsic factor falsely. Hence, there is the risk that the prediction uncertainty might be different for the transformed representation and for the actual observation defined by new scene configuration which is determined using the transformed representation. The other approach, *data space strategy*, use the same procedure from the inference pipeline to decide proposed scene configuration. Yet, instead of evaluating the usefulness based on latent space values, the agent physically performs the transformation in *data* space and obtains the observation with the new configuration. The new observation is then compared with the previous one to check if it indeed has lower uncertainty. If the uncertainty is lower we use the prediction based on the newly observed image and its confidence as the current prediction and confidence respectively. Here, with the cost of performing an additional transformation, we eliminate the risk of falsely associating a proposed observation and its prediction uncertainty due to estimating uncertainty in the latent space.

4 Experimental Setting

In this section, we will describe how we generated our data set and specific details about the architectures and the training of the models we used.

4.1 Dataset Generation

The model we used to obtain disentangled representations requires a data set from which we can define mini-batches where only one generative factor is altered. Since we defined our use case as the visual evaluation of the cars, to collect images of real cars in an environment where we have control over extrinsic factors is challenging. Hence, we propose to use synthetic images. We use the CAD models of cars from *modelnet40* data set [36] and we use Blender [7] to generate rendered images of the cad models. With Blender, we simulated an environment where we make observations about the cars by taking pictures. We defined two factors to control the ambient properties of the environment that are *light intensity* and *light location* and two factors to control the orientation of the camera w.r.t the car, namely *azimuth angle* and *elevation angle*. Note that, the position of the car is fixed at the origin of the environment and the camera is always pointing at the car. These factors together define extrinsic factors. In addition to the extrinsic ones, we have control also over the color of the car, which is an intrinsic factor. However, we also alter the color of the car in the data set to investigate the disentanglement performance of an intrinsic factor and compare it with the extrinsic ones.

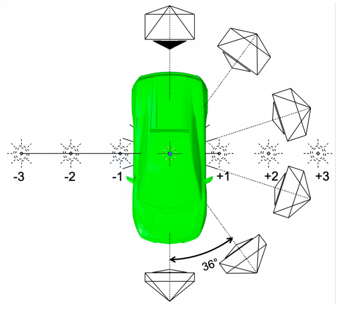

Figure 3: Possible light location and azimuth angles in the scene

Each mini-batch contains 6 images corresponding to different values of the modified factor. The factors can assume the following values:

- Azimuth angle: $\{0°, 36°, 72°, 108°, 144°, 180°\}$
- Elevation angle: $\{18.4°, 22.6°, 26.5°, 30.25°, 33.7°, 36.9°\}$
- Light location is fixed along the y-axis in the environment, such that we alter the location of the light only along the x-axis: $\{-3, -2, -1, +1, +2, +3\}$
- Light intensity: $\{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\} Watt/m^2$
- Color: green, cyan, blue, magenta, red, yellow

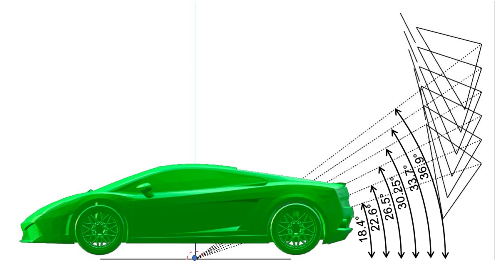

Figure 4: Possible elevation angles in the scene

For a single car model, we obtained renderings as a combination of these 5 factors, which sums up to 7776 images. We generated training, validation, and test sets with 40, 8, and 8 car models respectively. As we mentioned in section 3.2, we don't have the model label for any of the cars. Rather, we manually labeled each car instance based on the type of car it is. Specifically, we labeled the cars into 4 classes as *suv*, *sport*, *sedan* and *hatchback*. Even though there might be car models, where the distinction between these classes is vague, we believe these classes cover the majority of the available car models.

4.2 Model Architecture

In this thesis, we have used three different models as *dc-ign*, *intrinsic factor classifier* and *latent to data maps*.

The dc-ign model is based on the configuration in the paper [24]. We followed the same configuration since our settings are similar where we also used synthetic images with the same size of $150x150$. The encoder network has three convolutional blocks in which a convolutional layer with a kernel size of 5 and $2x2$ max pooling with a stride of 2 is used. Differently from the original implementation, we set the size of the latent space as 128. The decoder network is defined as again three convolutional blocks and each block has $2x2$ upsampling layer followed by a convolutional layer with kernel size 7. The training is performed for a maximum of 100 epochs. For the intrinsic factor classifier and the latent to data maps, we have explored many different model architectures. The ones described below summarize the best-performing ones. The intrinsic factor classifier is defined as a single layer network with 250 neurons and can be trained for a maximum of 50 epochs. For the latent to data maps, we used four identical multi-class classifiers where a model has two dense layers with neuron numbers of 150 and 50. We again defined the maximum number of epochs as 50. We trained all the above-mentioned models with Adam optimizer [22] together with early stopping that has a minimum loss decrease of 0.001 and a patience value of 3.

5 Results

As the proposed inference pipeline relies on proper disentanglement of extrinsic factors as well as the well-calibrated intrinsic factor classifier, in this section, we present our evaluations on these matters. Moreover, we show both the quantitative and qualitative results of the inference pipeline.

5.1 Disentanglement

We assess the outcome of the dc-ign in terms of disentanglement using one quantitative method and two qualitative methods for which the results give the same conclusions. We include the results only about extrinsic factors in this section. For the disentanglement results of the *color* as an intrinsic factor, we refer to the appendix 8.1.

5.1.1 Mutual Information Score

Measuring disentanglement quantitatively is an active research field. Although there is not a single particular metric that excels among others, each metric offers different advantages. For our case, the Mutual Information Gap (MIG) [6] score is favorable since it provides a metric of disentanglement performances for individual generative factors using mutual information. **Mutual information (MI)** measures the relationship between two random variables in terms of the amount of information they convey about each other. In other words, MI between two random variables, tells how much information can be obtained about one variable by observing the other. It can be computed with the following equation

$$\begin{aligned} MI(z_j, f_k) &= D_{KL}(q(z_j, f_k) || q(z_j) \otimes p(f_k)) \\ &= \mathbf{E}_{q(z_j, f_k)} \left[log \frac{q(z_j, f_k)}{q(z_j) p(f_k)} \right] \end{aligned} \tag{6}$$

$$\mathbf{E}\left[\mathbb{P}(Y = y | \hat{p} = p) - p\right] \tag{7}$$

where z_j denotes the j^{th} latent variable and f_k denotes the k^{th} generative factor with distributions of $q(z_j)$ and $p(f_k)$ respectively.

In the case of total independence between latent and true generative variables $MI(z_j, f_k)$ becomes 0. If the dependence between them is maximum, such that the relationship is deterministic, the mutual information is equal to the entropy of the true generative factor $MI(z_j, f_k) = H(f_k)$. Semantically *MI* evaluates the axis alignment property of the disentanglement. Ideally, the information about each generative factor should be encoded within a single latent variable [6]. In our case, we should expect high *MI* (dependency) for a generative factor and its corresponding latent factor and small values for other latent factors.

We present the results in table 1, where each column refers to a generative factor f_k and each row shows $MI(z_j, f_k)$ for a latent factor z_j. The last row highlights the entropy for each generative factor, which indicates the maximum possible *MI* between that factor and any latent factor. For the ease of interpretation, we normalized *MI* scores of each generative factor with their respective entropy value. Hence, a *MI* score of 1 indicates total dependence, whereas 0 indicates no shared information between the generative and the latent factor. We further highlighted the highest normalised *MI* for each factor in table 1. At a first glance, we can see that each generative factor has the highest MI for its corresponding latent variable. However, for each generative factor the degree of deviation, from maximum possible value 1, varies. For light intensity and elevation, the difference is smaller compared to azimuth and light location. Moreover, we can compare the normalized *MI* score of a generative factor with its designated latent factor and remaining latent factors. For a given generative factor, the higher the *MI* difference between its designated latent factor and the remaining ones, the better the disentanglement. We see that intensity and elevation have higher differences than azimuth and location. Consequently, *light location* and *azimuth* are the factors that have relatively more dependency with other latent variables which indicates entanglement of these factors.

		Generative Factors f_k			
		light intensity	light location	elevation	azimuth
Latent Factors z_j	light intensity	0.686	0.057	0.005	0.010
	light location	0.042	0.426	0.006	0.040
	elevation	0.039	0.012	0.872	0.013
	azimuth	0.054	0.037	0.006	0.523
	$H(f_k)$	1.7916	1.7917	1.7917	1.7917

Table 1: Normalised Mutual Information Scores

In the previous sections, the requirement for independence among extrinsic factors is mentioned. Nevertheless, some of these factors, by their nature, can be inherently entangled. In our setting, although *azimuth* and *light location* can be manipulated separately, they affect the shading in the scene together. Following a similar logic, we further identified another inherent dependency between *light location* and *light intensity* due to their common effect on the **brightness** in the scene. Brightness in the scene can be identified by the luminosity of the observations. Besides the straightforward relation of light intensity and brightness, light location also affects brightness since, as light moves away from the car, brightness also decreases as a result of the lower angle of incidence of the light. These insights also align with the MI scores. The latent variable with the second-highest MI for light location and azimuth generative factors are light intensity and light location respectively.

5.1.2 Traversing Latent Space

To further validate our findings of disentanglement, we used various qualitative methods. The **latent space traversal** method is used for the visual assessment of disentanglement. Given the latent representation of an observation, the value of a single latent dimension with a known generative factor encoded in it is altered linearly within a range. As that latent dimension is traversed, at each step the altered representation is reconstructed using the decoder network. Assuming disentanglement, these sequences of reconstructed images should show only visual changes corresponding to the traversed generative factor while other visual aspects of the scene should remain the same. We represented the results for all generative factors in the appendix in figure 18, where we can see all factors have smooth traversals.

Particularly, we investigated in detail the entangled cases for the azimuth and light location to visualize our above-mentioned insights. Figure 5, shows a partially entangled example for each of these two extrinsic factors. The latent azimuth is traversed from 0° to 180°. Between the first and last two consecutive images, we can observe the change in terms of only shading, which is due to light location. This behavior visually validates that indeed azimuth and light location remain entangled to each other. Similarly in the light location traversal, the reconstructed images illustrate the change in light location from -3 on the left-hand side of the car to 3 on the right-hand side. During the first half of the traversal, it can be noticed that the brightness in the scene is increasing despite altering only the location. This investigation indicates that light location has a dependency on the intensity. Additionally, in the last picture, although the figure of the car is distorted the shading depicts that the car is rotated counterclockwise. These conclusions agree with the findings of MI scores as we detected dependency of the light location to light intensity and azimuth angle.

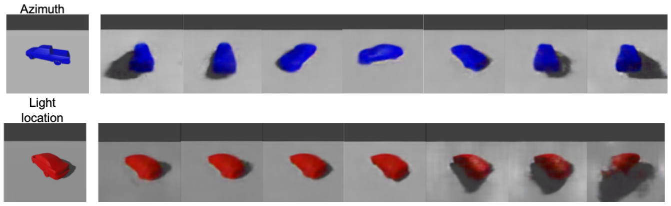

Figure 5: Entangled latent space traversal of azimuth and light location

5.1.3 Visualising Latent Space

Lastly, we visualized the alignment of disentangled factors in the latent space for the test set. Latent space visualizations are obtained by choosing a pair of disentangled factors and plotting the values of corresponding latent dimensions for each image in the test set. Each plot is colored based on the values of the true generative factor displayed on the y-axis. Here, we presented and analyzed the results for azimuth and light location as challenging cases. Whereas, we demonstrate the behavior of properly disentangled factor with elevation. The visualization of other disentangled factors are presented in the appendix 8.1 in figures 19 and 20.

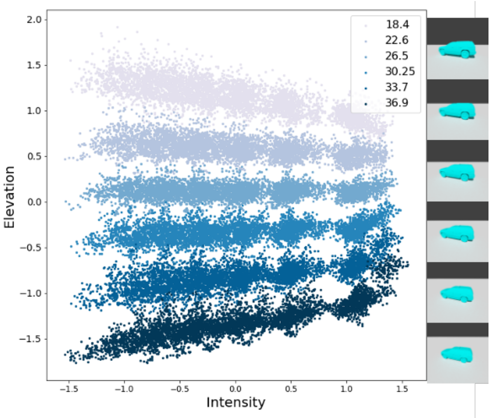

Figure 6: Latent space visualization of Elevation-Intensity *latent* factors. The coloring is based on true elevation values of the images represented by dots. The sequence of the images on the right side are the example images of the true elevation values. Each image denotes the cluster with a particular color.

Disentangled factors show a clear separation of the true factor values in the latent space. *Elevation* is such a factor, with the highest MI score among extrinsic ones, as we can also see the obvious distinction of different elevation values in figure 6. The latent space alignment of both *location* and *azimuth* factors show the absence of a clear structure in figure 9 and 11 respectively. The main reason for this behavior is due to the varying shade in the scene caused by the position of the light. Depending on whether the light is on the left or the right side of the car, the shade is formed on the opposite sides.

Particularly for the location latent plot in figure 9, the overlapped clusters on the side of the plot are highlighted with the red bounding boxes. These overlapped clusters denote the similar views of azimuth 0° and 180°. To be able depict this more clearly, we divided figure 9 into three separate plots as two boundary azimuth angles 0°, 180° and middle angles (36°, 72°, 108°, 144°). In figure 7 the embeddings for the boundary angles of figure 9 were separated into those corresponding to images with azimuth 0° (left) and those corresponding to 180° (right). For both boundary angles, we can see the alignment is more structured compared to the light location in figure 9. We observe that even though the light locations can be distinguished from the embeddings, the front (0°) and rear (180°) of the car are indistinguishable due to the overlap.

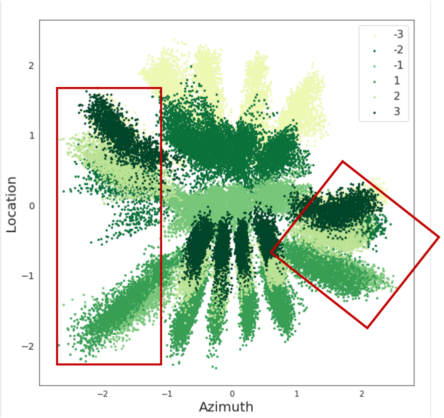

Figure 9: Latent space visualization of Location-Azimuth *latent* factors. The coloring is based on true light location values of the images represented by dots. The clusters on the side of the plot, specified by bounding boxes denote the boundary azimuth angles, whereas the ones in the middle denote the middle azimuth angles

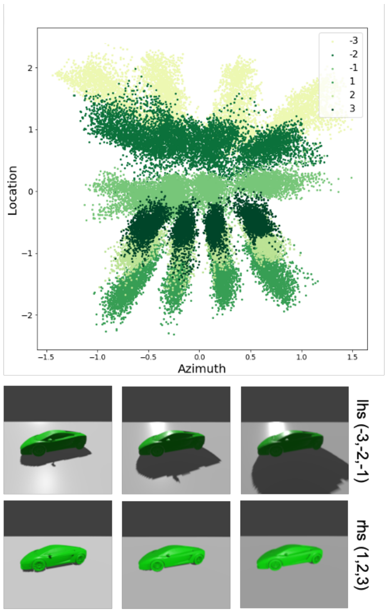

Figure 10: Separated latent space visualization of Location-Azimuth factors. The plot denotes the examples with true azimuth angles of 36°, 72°, 108°, 144°. The images below demonstrate the visual difference when the light is on the left and right-hand side of the car for the middle azimuth angles.

While the figure 7 underlies the reason for overlapping and also shows the layout in the latent space for boundary (0° and 180°) azimuth angles, with figure 10 we reason about middle angles. According to figure 10, it can be said that the middle azimuth angles illustrate a more organized layout than boundary angles without further subdividing. Especially, the left-hand side light locations are more scattered than right-hand side locations. This behavior can

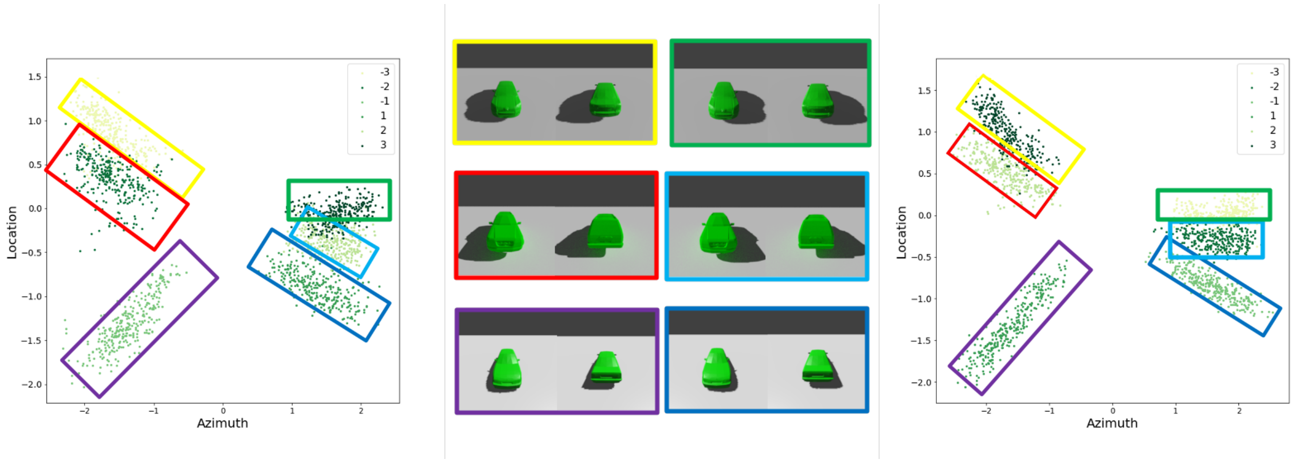

Figure 7: Separated latent space visualization of Location-Azimuth factors. The left and the right plots denote the examples with true azimuth angles 0° and 180° respectively. The pairs of images in the middle of the figure denotes similar views. Notice the shadings that are on the same side of the scene and with similar geometries for each pair, which is why their latent representations are overlapped in figure 9. We further highlighted pairs of images with bounding boxes that also match their corresponding clusters in the plots.

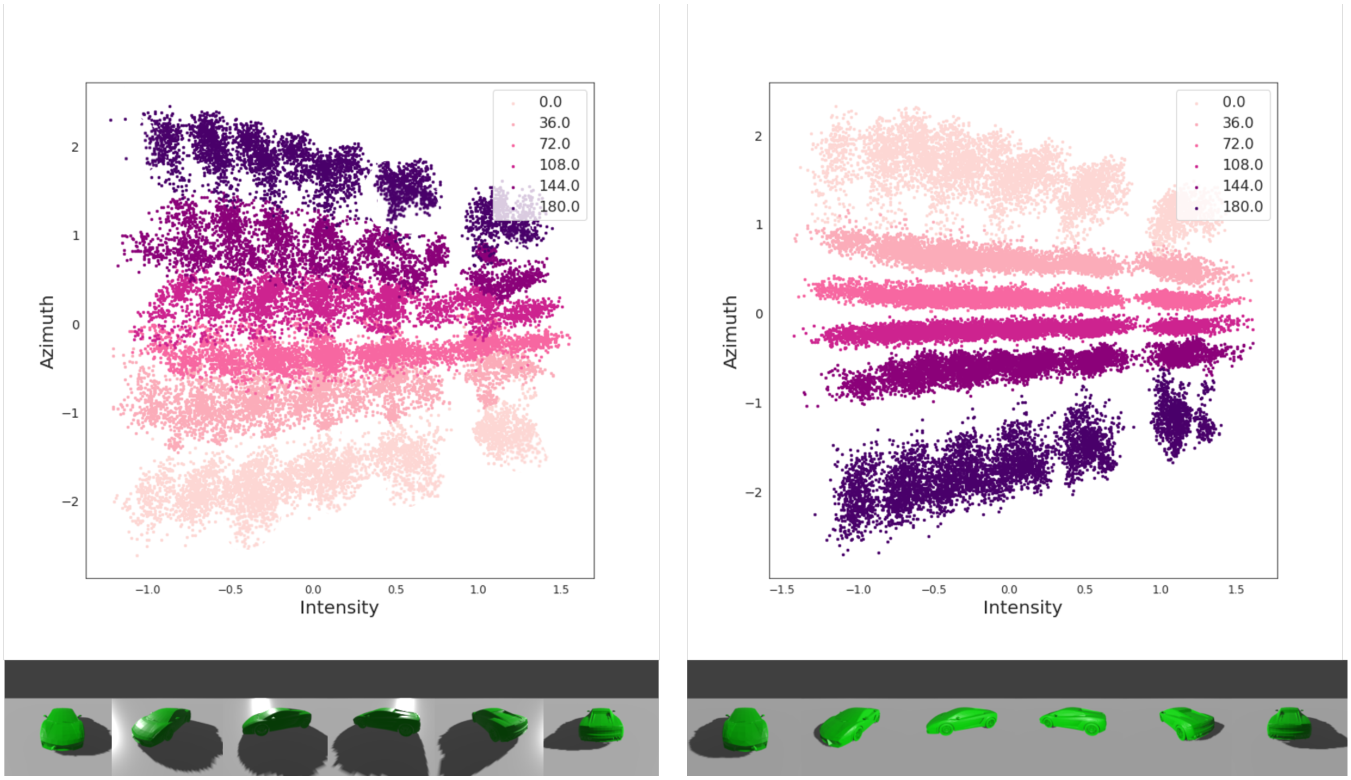

Figure 8: Separated latent space visualization of Azimuth-Intensity factors. The left and the right plot exhibit structure when light is on the left and the right-hand side of the car respectively.

be attributed again to the shade of the car. As it is also depicted with images in figure 10, when light is on the right-hand side, the shade is occluded by the car itself, whereas for left-hand side light locations the shadings are visible. When the light is on the left-hand side, the properties of the shade cause additional visual variations in the scene which are affected by multiple factors such as the light intensity or elevation. When light is on the right-hand side, there is no shadow and thus the visual appearance of the images does not vary as much with light intensity or elevation.

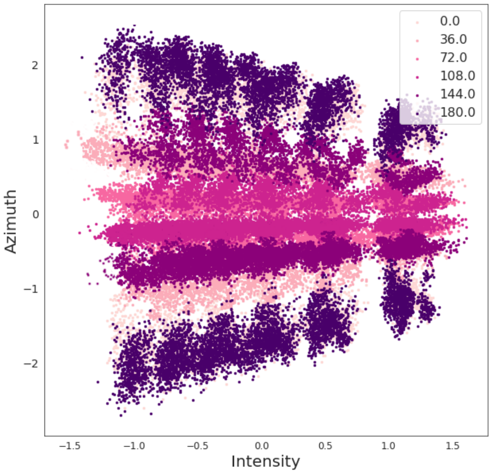

Figure 11: Latent space visualization of Azimuth-Intensity *latent* factors. The coloring is based on true azimuth values of the images represented by dots.

As another not fully disentangled factor, latent space visualization of *azimuth* vs light intensity in figure 11 exhibits an unstructured alignment with no interpretability. Nevertheless, dividing again into sub-cases reveals the underlying structure. Specifically, we investigated figure 11, in more detail by dividing the images based on the true location values as left-hand side $(-3,-2,-1)$ and right-hand side $(1,2,3)$. As displayed in figure 8, each case pictures an obvious distinction among different azimuth values but in reverse order. Hence, it would indicate the azimuth latent variable indeed captured information about the azimuth generative factor. The sequences of images on the left and right side of the figure 8 varies in terms of only the shading in the scene. Although we cannot deduce physical interpretation, it seems the existence of the shading in the scene reverses the alignment. Moreover, as depicted in the right plot of figure 8, again due to the occluded shading for middle azimuth angles when light is on the right-hand side, the azimuth latent values for these images has less disparity. The effect of this can be observed even between the boundary and middle azimuth angles for the right-hand side light location as displayed in the right plot of figure 8.

5.2 Uncertainty Estimation

5.2.1 Calibration Error

In the context of this thesis, the intrinsic factor prediction task is defined as the classification of the **type** of the car. The type classifier achieved 89% accuracy on both the training and the test set. This indicates that the classifier does not over-fit to the train set and thus it can generalize its performance. Furthermore, we presented the f1-score, precision and recall on the test set for each *type* class in table 2.

	suv	**sport**	**sedan**	**hatchback**
f1-score	0.81	0.98	0.94	0.84
precision	0.97	0.98	0.94	0.75
recall	0.70	0.97	0.94	0.96
support	15552.00	15552.00	15552.00	15552.00

Table 2: Car type prediction results of test set

Besides the classification results of the intrinsic model, the confidence calibration of the predictions should be evaluated. We estimated the uncertainty of the predictions using the *soft-max* output of the intrinsic classifier. To validate this approach, we need to make sure that on average the intrinsic factor classifier is neither over nor under-confident with its predictions, this corresponds to a *confidence calibrated* classifier. Prediction confidences of a classifier are calibrated if its prediction confidence is equal to the prediction accuracy values. To this end, we have tested the confidence calibration [15] of our intrinsic classifier. In figure 12, the reliability diagram can be found which depicts the confidence and accuracy levels for the bins computed for ECE, as described in section 3.2. Figure 12 shows that, except for the first bin, almost all confidence intervals have accuracy and confidence similar. Despite the relatively higher difference of the first bin, this has an insignificant effect on the final calibration of the model, since there are only 5 examples that fall to this bin. Moreover, the difference between accuracy and confidence has a decreasing trend as the confidence of the predictions increases.

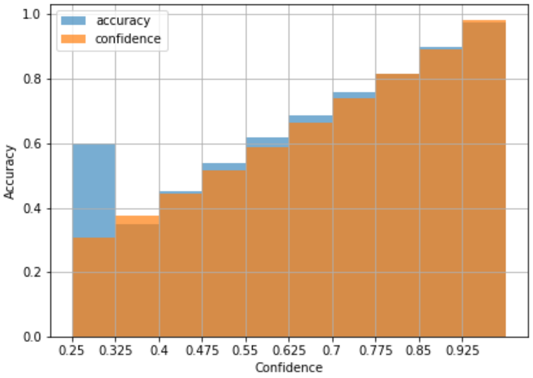

Figure 12: Reliability diagram of car type classifier

The intrinsic classifier also achieved 0.0105 ECE. This metric does not offer a certain threshold such that below that value, it can be said the model is calibrated enough. However, to evaluate our level of calibration we compared with a model in the literature that performs a similar task on a similar data set and for which the calibration results are known. In particular, we chose *ResNet 50* model with *Stanford Cars* [23] data set. This choice can be justified since the prediction task is quite similar to ours, where it is multi-class classification on the images of cars in terms of their make, model, or year. As presented in [16], ResNet 50 on Cars data set has 0.0430 ECE without any additional calibration method. Even with additional calibration methods presented in [16], the lowest ECE of Resnet 50 on Cars data set is 0.0174. Consequently, our intrinsic classifier has ECE about less than two-third of the best ECE of ResNet 50 on cars data set. This concludes that our intrinsic classifier has calibrated confidences.

5.2.2 Statistical Analysis of Uncertainty

Given that the intrinsic classifier is calibrated, *uncertainty* is estimated by the entropy of the soft-max output. Additionally, we should check that the entropy does depend on the different values of the extrinsic generative factors. Thus, when we vary the values of a factor we expect entropies to be different. If the extrinsic generative factors do not affect the prediction uncertainty then across different values of a generative factor the distribution of entropies should be the same. This assumed difference can be confirmed by assessing the distributions of prediction entropy for different values of each extrinsic factor. The estimation of entropy values indicates exponential distribution. For ease of presentation, we used log-entropy values to picture the distributions in figures 21, 22, 23 and 24.

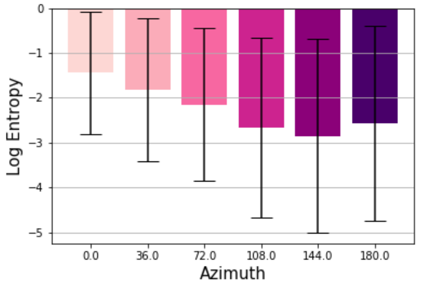

Figure 13: Error-bar plot of log entropy of prediction for azimuth extrinsic factor

To obtain a more clear picture of the distribution differences, we used error-bar and violin plots. We presented results for the azimuth extrinsic factor in this section. whereas the visualisation of the remaining extrinsic factors is given in appendix 8.2 with figures 25, 26 and 27. In figure 13, the bars, and the error lines denote the mean and the variation of entropy values for each azimuth angle subset. Among different azimuth angles, 144° has the lowest mean entropy, which indicates that on average, images with azimuth angle 144° lead to more confident predictions w.r.t. other angles. Although we can see that these subsets have different mean entropy values, it still does not inform if these subsets have different distributions. Hence, by also visualizing the distribution using violin plots as shown in figure 14, we can compare these distributions in terms of their shape. Parallel to the figure 13 where angle 36° and 72° have closer means, in figure 14 it can be seen the shapes of these distributions are also alike. Moreover, in figure 14, the angle with the smallest mean entropy (144°), has also the narrowest shape, which implies, this azimuth angle provides the highest confidence among others.

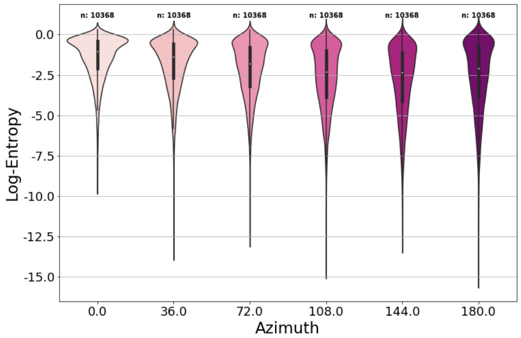

Figure 14: Violin plot of log entropy of prediction for azimuth extrinsic factor

We also performed statistical tests to ensure the distributions of the entropy depend on the extrinsic factor values. Since the distributions do not appear to be normal as seen in figures 21, 22, 23 and 24, we used *Mann Whitney U* (MWU) test, which does not require normality, to identify whether the distributions of the entropy for two values of a factor are statistically significant. The null hypothesis of the test is that the calculated entropies for two factors come from the same distribution (i.e. they are equally distributed). Since MWU is a pairwise test, we performed the test between the entropy estimated for pairs of extrinsic factor values.

Azimuth	0°	36°	72°	108°	144°	180°
0°	-	0.000	0.000	0.000	0.000	0.000
36°	0.000	-	0.190	0.000	0.000	0.001
72°	0.000	0.190	-	0.000	0.000	0.003
108°	0.000	0.000	0.000	-	0.000	0.000
144°	0.000	0.000	0.000	0.000	-	0.000
180°	0.000	0.001	0.003	0.000	0.000	-

Table 3: p-values of MWU test of azimuth latent factor

The test results of azimuth latent factor are presented in table 3, and of the remaining extrinsic factors again in appendix 8.2 in tables 7, 8 and 9. The test results are symmetric thus, the upper and lower half of the table 3 are the same. As highlighted in the table only for the tests between angle 36° and 72° the null hypothesis cannot be rejected with a p-value of 0.19. This means that except for the angles 36° and 72°, the distribution of entropy values are statistically different from each other. Therefore the entropy does depend on the values of the extrinsic factors. This finding also matches with the conclusions we made form figures 13 and 14.

Among the remaining extrinsic latent factors, we also found that for both *elevation* and *light location* except one pair of values, the distributions of the intrinsic factor prediction entropies are statistically different from each other. On the other hand, for the *light intensity*, all pairs of values indicates statistically significant differences for distribution of the entropy. In detail results of the significance test of other factors can be found in appendix 8.2 in figures 7, 8 and 9.

5.3 Inference Pipeline

We performed extensive experiments with combinations of different inference variations and different learning rates. All possible combinations of different variations give 16 settings of the inference pipeline. However, we experimented with 12 inference settings. This is because multi transform and best of all inference, where multiple extrinsic factors can be altered at once, cannot be applied with latent space strategy. In the latent space strategy, the final intrinsic factor prediction is decided based on one of the transformed latent representations. Hence, in case the transformed representations output different car type predictions, it would cause ambiguities for the final intrinsic factor prediction.

We have evaluated the usefulness of the inference pipeline, both quantitatively and qualitatively. We have tested the inference pipeline with the 12 different settings explained in section 3. As discussed earlier, we introduced the *learning rate* parameter within the inference pipeline to scale the error signal. We expect that the learning rate affects the behavior of the inference pipeline greatly. Thus, to detect the optimum learning rate, we have performed experiments with 12 different learning rates, with each inference setting on the validation set. Particularly, we evaluated the results on the validation set, based on the change of *f1-score* caused by the inference pipeline. To be precise, we apply the inference pipeline on the validation set and we computed the f1-score on the final predictions of the inference pipeline. The results can be seen in figure 28 where each plot belongs to one of the inference variations. We further denoted the best performing learning rate with a marker for all variations. When deciding optimum learning rates, we sought a value where the performance either peaks or reaches a plateau. According to the table 28, the only settings, that did not lead to an improvement, are single step regular and best of each inference with latent strategy.

Notice that in figure 28, the plots for *best of all inference* do not show a change in the performance for different learning rates since this inference baseline, does not use the gradient information, however it is just plotted for comparison. Moreover, multi-step and single-step inferences do not differ, since we have, *no change in any extrinsic factor*, as a stopping criterion and best of all inference proposes the same scene configuration even with multi-step.

Having the optimum learning rate for all inference settings, we have assessed the performance of the inference pipeline on the test set. The results are exhibited in table 10. The table contains for each setting the optimum learning rates that are found based on the validation set performance. It also shows the number of correctly classified and misclassified images by inference pipeline, among the images classified correctly by the original intrinsic classifier. Similarly, we also present in table 10 the number of images that were initially classified wrong and after the inference pipeline either classified correctly or remain misclassified. In other words, *initial correct-inference wrong* indicates the detriment due to the inference pipeline whereas *initial wrong-inference correct* marks the improvement caused by the inference pipeline. To be able to demonstrate difference in the prediction performance, we present the the change in terms of *precision*, *recall* and *f1-score* w.r.t. the original classifier for each inference setting. We further present for each inference setting the highest decrease in entropy among the images from the test set as well as how entropy changes on average. Lastly, for the multi-step settings, we presented the maximum number of inference steps applied to a single image and the average number of inference steps.

Additionally, table 10 shows the performance of the original classifier, as it predicted 55444 images correctly and 6764 of them incorrectly while achieving 91%, 89%, and 89% precision, recall, and f1 score respectively. The results presented in table 10 point that 9 out of 12 settings improve the classification. Particularly, both single and multi-step regular inference and single-step best of each inference with latent strategy degrade the classification results. Moreover, as highlighted on table 10, *multi-step multi-transformation inference with data strategy* achieved the highest improvement where precision, recall and f1 score has increased by 5.17%, 6.81% and 6.88% respectively. Consequently, with the best performing setting, we achieved 96.1% precision, 95.8% recall, and 95.7% f1 score. Moreover, with this setting, the inference pipeline managed to reduce the uncertainty in the best case by 1.37 and on average by 0.13 while spending at most 6 and on average 1.45 steps on an image.

number of inf. steps	percentage of inf. steps caused config. change	percentage of images with a new config.	total number of transformations
90255	57.9%	38.9%	79049

Table 4: Additional statistics for the best performing inference setting

To give more insight into the behavior of the multi-step multi-transformation inference with data strategy, we have computed the statistics in table 4. Given that there are 62208 images in the test set, the total number of inference steps applied is 90255. However, only 57.9% of inference steps caused a transformation in at least one of the extrinsic factors. The main reason for this difference is the additional step requirement of *data strategy* to be able to evaluate the proposed transformation. In other words, out of 90255 steps, for 42.1% of them inference pipeline decided the proposed transformation does not lead to lower entropy. With this inference setting, since it is allowed to alter multiple extrinsic factors within a single inference step, 57.9% of the total number of inference steps caused transformation of an extrinsic factor 79049 times. Moreover, due to the multi-step property of this setting, 57.9% of the inference steps that caused a change in the scene configuration result in a new scene configuration for 38.9% of the total images in the test set.

In addition to the statistic from table 4, we also presented the distribution of the extrinsic factors among in total of 79049 transformations in table 5. It can be seen that number of times the inference pipeline utilized transformations of intensity, location, and elevation are close, where azimuth is preferred slightly fewer times.

intensity	location	elevation	azimuth
20175	20164	20027	18683

Table 5: Number of transformation per extrinsic factor

In addition to the evaluation of the effect of the inference pipeline with classification metrics, we tried to convey the advantages of the proposed method visually. In figures 15 and 16 we visualised the transformations proposed by the inference pipeline. For both figures, the images in the first column are the original images that are fed to the inference pipeline. The captions on top of each image in the first column denotes the chosen extrinsic factor to alter and the initial values for each extrinsic factor. The images in the remaining columns denote the transformations of the extrinsic factor annotated at the top of each column. Furthermore, the new values of the transformed image are given on top of each image. For each inference example, the chosen transformation is highlighted with the red frame.

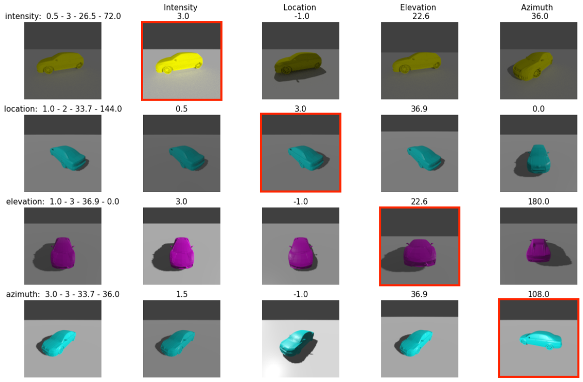
Intensity
3.0
Location
-1.0
Elevation
22.6
Azimuth
36.0
intensity: 0.5 - 3 - 26.5 - 72.0
location: 1.0 - 2 - 33.7 - 144.0
0.5
3.0
36.9
0.0
elevation: 1.0 - 3 - 36.9 - 0.0
3.0
-1.0
22.6
180.0
azimuth: 3.0 - 3 - 33.7 - 36.0
1.5
-1.0
36.9
108.0

Figure 15: Visual evaluation of inference pipeline with intuitive results

The results presented in figure 15 and 16 are the results of single-pass regular inference with data strategy. We used regular inference since the multi-transformation inference might pick multiple extrinsic factors. Hence that would prevent us to analyse the results in terms of each extrinsic factor separately. On the other hand, the choice of single-pass is to also demonstrate the usefulness of the inference pipeline when only one step is allowed. In particular, the images in figure 15 depict the cases where inference results are also visually intuitive to humans. As can be seen, for each example in figure 15, the chosen transformation indeed provides a better scene configuration such that we can infer the type of the car easier compared to other ones. Whereas, in figure 16, we presented examples where the choices of inference pipelines are not intuitively better for the car type prediction task. However, it should be noted that the inference pipeline pays attention to the preferences of the intrinsic classifier. Hence, the intrinsic classifier might have different preferences of the best scene configuration for car type classification than common human sense.

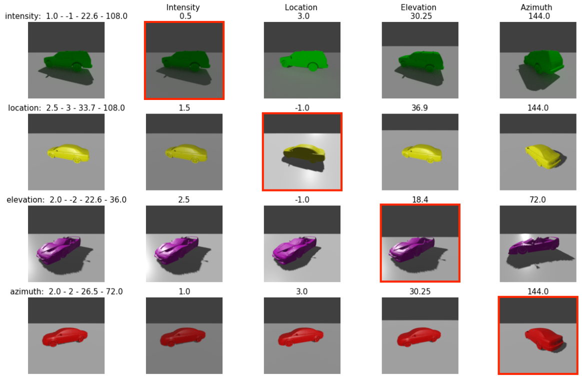
intensity: 1.0 - -1 - 22.6 - 108.0
Intensity
0.5
Location
3.0
Elevation
30.25
Azimuth
144.0
location: 2.5 - 3 - 33.7 - 108.0
1.5
-1.0
36.9
144.0
elevation: 2.0 - -2 - 22.6 - 36.0
2.5
-1.0
18.4
72.0
azimuth: 2.0 - 2 - 26.5 - 72.0
1.0
3.0
30.25
144.0

Figure 16: Visual evaluation of inference pipeline with counter-intuitive results

6 Discussion

It should be noted that the success of the inference pipeline depends on both the disentanglement and reliability of the uncertainty estimation. This is because we exploit the disentangled characteristic to assess the potential of each extrinsic factor to change the uncertainty of intrinsic factor prediction.

The various evaluations of disentanglement indicate that although we manage to capture the majority of the detail about each extrinsic factor with their respective latent factor, *light location* and *azimuth* preserve dependency to some extent. During the inference pipeline, we assess the contribution of each extrinsic factor with the assumption that they are disentangled.

Thus, if a latent factor encodes information about multiple extrinsic factors, by assessing the contribution of this latent factor to the prediction uncertainty, extrinsic factors other than its designated one would interfere. In other words, we might alter both extrinsic factors whenever we intend to manipulate only one of them. However, it should be noted this behavior is natural. Because these extrinsic factors indeed depend on each other in the data space. This problem might be tackled by defining extrinsic factors, such that they are inherently independent.

We reasoned about the validity of our estimation of predictive uncertainty using expected calibration error (ECE) and reliability diagrams. Even though ECE gives a single metric to assess the calibration of the model, it does not tell details about the calibration. On the other hand, the reliability diagram in figure 12 indicates that even for the intervals with a relatively high accuracy-confidence difference the confidence values are lower, which indicates under-confidence. Although this is not the preferred behavior, having under-confident predictions would be preferred compared to over-confidence. Since we are trying to detect the ambiguous cases, that are predictions with low confidence, by being under-confident we avoid the risk of falsely evaluating examples as confident. This can be also seen as avoiding false positives where the positive case denotes the observations with confident predictions. It should be further noted that under-confidence mostly occurs to predictions with low confidence, which refers to ambiguous cases.

It would be possible to have a more accurate classifier with a network that has more hidden layers and more neurons for each layer. Nonetheless, that would harm the confidence calibration, as it is known that the depth of the network and calibration error is proportional [15]. Moreover, having a more powerful classifier with higher accuracy would affect the behavior of the inference pipeline. Following the assumption that low uncertainty predictions imply correctness of the predictions, as the accuracy increases the entropy of predictive distribution decreases. Hence, in the inference pipeline, we would receive lower gradient values which eventually means smaller updates on the latent representation of the input observation. Consequently, we would obtain a relatively lower decrease in entropy. In other words, our inference pipeline would be more beneficial for classifiers with lower performance.

The results in section 5.3 indicate that there is an obvious performance difference between data and latent strategy. These two approaches differ from each other based on how it is decided whether the proposed scene configuration leads to lower uncertainty. Following the latent-space heuristic during the inference might lead to an incorrect mapping between the entropy of the transformed latent representation and the proposed scene configuration. In other words, latent space strategy might underestimate the entropy of the proposed scene configuration which would lead the agent in a position where entropy is higher than the previous configuration. The main reason for that behavior is the limitation of latent-data map classifiers to capture the change in data-space extrinsic values based on the change we apply in the latent space. The most intuitive solution for that problem would be to map the latent values to data space factors values in a continuous manner, rather than the grid approach. This way, we could obtain the true change in the data space rather than forcing the update in the latent space to be mapped to one of six discrete values. More precisely, given the transformed representation in the latent space we could obtain the values of the corresponding factors according to the following equation:

$$ext_{new} = ext_{old} - \frac{\partial H}{\partial z_{ext}} \frac{\partial z_{ext}}{\partial f_{ext}} \alpha \tag{8}$$

The first derivative term in the right-hand side of the equation is, the corresponding extrinsic component of the gradient vector we used during the inference. The second derivative term is essentially the coefficient of the linear function ($z_{ext} = af_{ext} + b$) that maps the values of the true factors to latent space values. By having this linear function for each extrinsic factors, we could obtain the results of the inference pipeline in terms of continuous data space factor values. However, as also depicted in figure 11 and 9, the relationship between the factors of *light location*, *azimuth* and their latent values has a non-linear nature which prohibits us to fit a linear function. Thus, due to the limited disentanglement for these factors, the alignment for them in the latent space lost the semantics that exists in the data space.

Following the above-mentioned reasoning, another indication to favor the data space strategy would be the superior performance of the data strategy when multiple inference steps are allowed. When compared with the original classifier, despite the slightly lower performance of latent strategy with single-step inference, it exhibits a considerable degraded performance with multi-step inference. This behavior can be explained by the fact that latent strategy might underestimate the entropy of the new configuration. Furthermore, by allowing this for multiple steps, the inference pipeline propagates this error and might end up in a configuration with higher entropy and possibly misclassified. In figure 17, we see an example of this behavior with multi-step regular inference. As seen in figure 17, although both strategies converge until the fourth step, the latent strategy obtains a transformed latent representation with lower entropy and falsely associates it with the new scene configuration found by the latent to data map. Hence, the inference ended up with increased entropy and misclassification. On the other hand, the data strategy obtains the actual image with the proposed configuration and stops the inference due to an increase in entropy. This finding also aligns the results of the multiple-step regular inference with the latent strategy represented in table 10. With this inference setting the entropy increased on average and the inference pipeline made correct predictions as incorrect more than it did vice versa.

As shown in table 10, the *best of all* inference, as an intuitive inference baseline, performs well with both single and multi-step settings. More precisely, the performance of *best of all* inference is

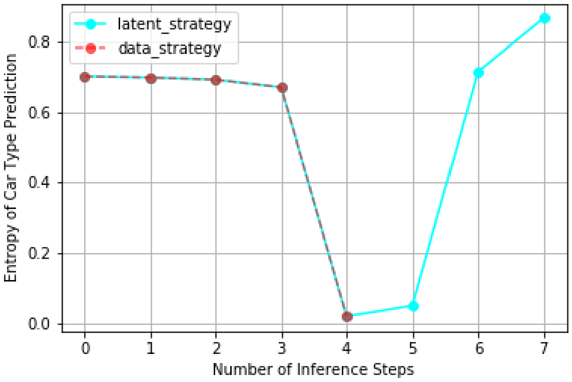

Figure 17: Performance difference between latent and data strategy

invariant through different learning rates and inference step settings since it proposes only one configuration regardless of the inference heuristics. It should be noted that the usefulness of this baseline is dependent on the number of car models that exist in the dataset. In the current setting, there are four extrinsic factors and six different values for each of them, which makes 6^4 different scene configurations. Instinctively, the chance of having one configuration that gives correct predictions with low entropy for all different car models would decrease as we increase the number of car models.

The *best of each* inference, as another baseline, behaves similarly to other inference methods as its performance is better with **data** strategy. Apart from being a baseline, the performance of this method demonstrates the benefits of the heuristic we developed for the inference pipeline. Despite the fixed transformations for each extrinsic, the best of each inference uses the proposed method for attributing the uncertainty of the prediction to extrinsic factors and identifying which extrinsic factor to modify. By looking at the results of this method especially with data strategy in table 10, we see approximately 2% increase in the f1 score even with single-step inference. Hence, we can deduce that the idea of performing transformations on the latent representations and using the uncertainty of their predictions to decide the most rewarding extrinsic is indeed beneficial.

One major limitation of this study is the requirement of additional work to be able to use it with real-world images. It should be noted that the described methodology needs a data-set with certain requirements which would be hard to obtain with real images. Using rendered images would allow us to replicate the environment with particular extrinsic factors on which the agent has control. We could decide on which extrinsic factors to control, to represent the real-world environment in the best way. Hence, the agent would optimize its performance for the given real-world environment. By increasing the number of extrinsic factors and the number of possible values they can be assigned to in the rendering environment, we can generalize to more real-world environments. At this point, the lack of generalization from rendered images to real ones is called **domain gap**. As described in [33], **domain randomization** tries to solve domain gap by augmenting the synthetic images with realistic lighting conditions (brightness, contrast) and real background images. To maximize the efficacy of this technique, the rendering should be performed on textured 3d objects. Using textured objects and rendering them with the highest settings, photo-realistic and high-quality images can be obtained. Consequently, the success of the domain randomization would be maximized. However, being able to render high-quality images is a rather computation-intensive process. Thus, given the high number of configurations we need to render for each car type, it limited the applicability of the method.

7 Conclusion

In this thesis, we pursued to develop a method to improve the performance of intrinsic factor predictions by using the uncertainty to identify conditions of observations with higher confidence. The proposed *inference pipeline* relies on disentanglement to separate the extrinsic factors and evaluate each factor for their potential to reduce the uncertainty of the prediction. We used an existing disentanglement method and showed that we could capture the latent representations in a factorized way. Nevertheless, we concluded that some of the extrinsic generative factors naturally depend on each other and cannot be entirely disentangled using the method we deployed. We qualitatively and quantitatively assessed the confidence calibration of our model, with which we estimate the uncertainty of intrinsic factor prediction. The usefulness of the method we developed is assessed both qualitatively and quantitatively. We showed that the strategy we defined for the inference pipeline leads to an improvement in the majority of the settings we experimented with. With the best performing inference setting, compared to the inference performance of the original classifier, we achieved an increase of 5.1%, 6.8%, and 6.8% for precision, recall, and f1-score respectively.

References

[1] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, 2019.

[2] V. Belle. Logic meets probability: Towards explainable ai systems for uncertain worlds. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 5116–5120, 2017. doi: 10.24963/ijcai.2017/733

[3] Y. Bengio, A. C. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 2012.

[4] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks, 2015.

[5] G. Carneiro, L. Zorron Cheng Tao Pu, R. Singh, and A. Burt. Deep learning uncertainty and confidence calibration for the five-class polyp classification from colonoscopy. *Medical Image Analysis*, 62:101653, 2020. doi: 10.1016/j.media.2020.101653

[6] T. Q. Chen, X. Li, R. B. Grosse, and D. Duvenaud. Isolating sources of disentanglement in variational autoencoders. *CoRR*, abs/1802.04942, 2018.

[7] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[8] K. J. Friston, J. Mattout, and J. Kilner. Action understanding and active inference. *Biological Cybernetics*, 104:137–160, 2011.

[9] A. Gabbay and Y. Hoshen. Demystifying inter-class disentanglement, 2020.

[10] Y. Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[11] Y. Gal and Z. Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference, 2016.

[12] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2016.

[13] A. Graves. Practical variational inference for neural networks. pp. 2348–2356, 2011.

[14] D. Gunning and D. Aha. Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58, Jun. 2019. doi: 10.1609/aimag.v40i2.2850

[15] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks, 2017.

[16] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. *CoRR*, abs/1706.04599, 2017.

[17] R. Hamaguchi, K. Sakurada, and R. Nakamura. Rare event detection using disentangled representation learning, 2018.

[18] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[19] E. Hüllermeier and W. Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods, 2019.

[20] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *CoRR*, abs/1703.04977, 2017.

[21] H. Kim and A. Mnih. Disentangling by factorising, 2018.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[23] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia, 2013.

[24] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum. Deep convolutional inverse graphics network. *CoRR*, abs/1503.03167, 2015.

[25] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[26] O. Litany, A. Morcos, S. Sridhar, L. Guibas, and J. Hoffman. Representation learning through latent canonicalizations, 2020.

[27] A. Loquercio, M. Segu, and D. Scaramuzza. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, Apr 2020. doi: 10.1109/lra.2020.2974682

[28] E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh. Disentangling disentanglement in variational autoencoders, 2018.

[29] C. Molnar. *Interpretable Machine Learning*. 2019. `https://christophm.github.io/interpretable-ml-book/`.

[30] S. N, B. Paige, J.-W. van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, and P. Torr. Learning disentangled representations with semi-supervised deep generative models. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., *Advances in Neural Information Processing Systems 30*, pp. 5925–5935. Curran Associates, Inc., 2017.

[31] M. Pakdaman Naeini, G. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907, 04 2015.

[32] N. Sajid, P. J. Ball, and K. J. Friston. Active inference: demystified and compared, 2019.

[33] M. Sundermeyer, Z. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3d orientation learning for 6d object detection from RGB images. *CoRR*, abs/1902.01275, 2019.

[34] K. R. Varshney and H. Alemzadeh. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. *CoRR*, abs/1610.01256, 2016.

[35] H. Wang, X. Shi, and D. Yeung. Natural-parameter networks: A class of probabilistic neural networks. *CoRR*, abs/1611.00448, 2016.

[36] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes, 2015.

[37] J. Yang, S. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis, 2016.

8 Appendix

8.1 Disentanglement

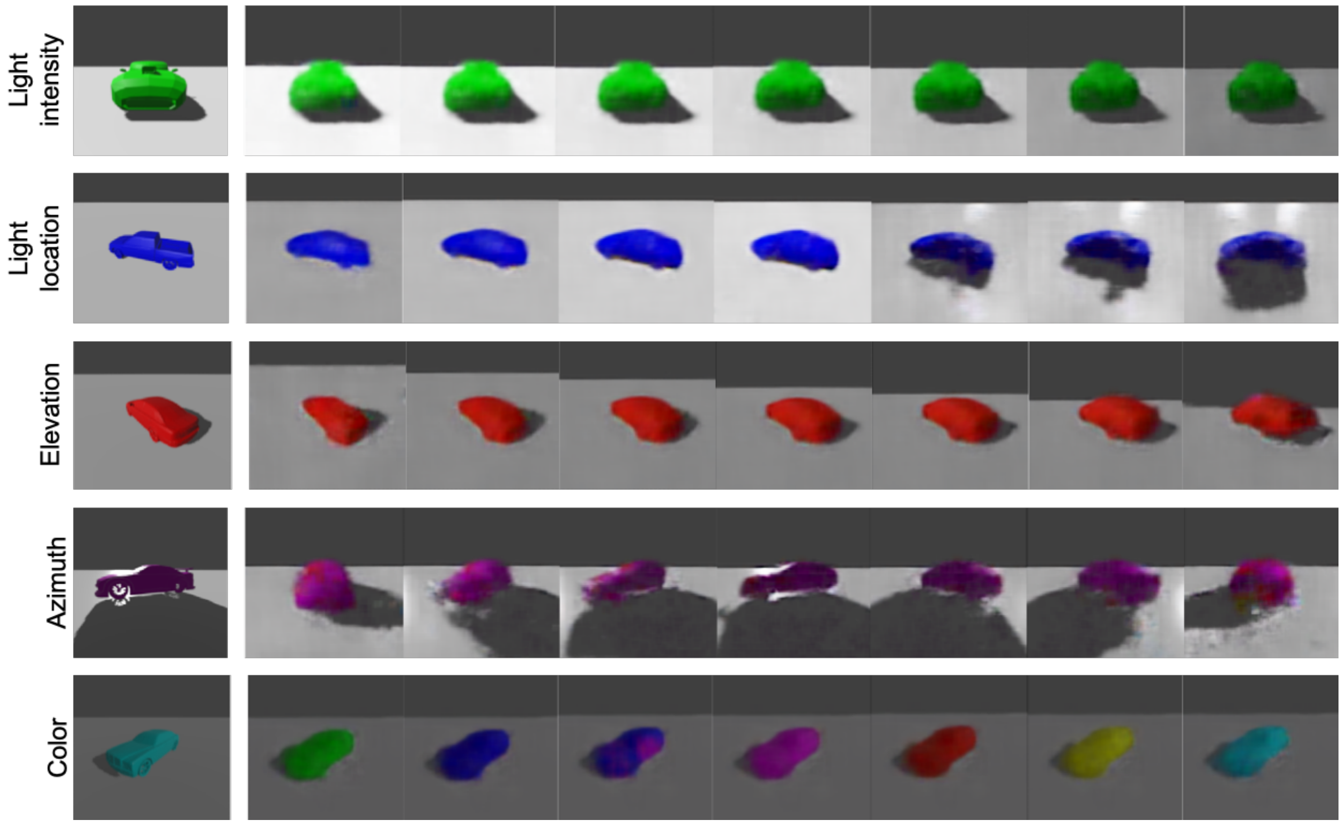

Figure 18: Smooth latent Space Traversal Reconstructed Images

		Generative Factors f_k				
		light intensity	light location	elevation	azimuth	color
Latent Factors z_j	light intensity	0.686	0.057	0.005	0.010	0.004
	light location	0.042	0.426	0.006	0.040	0.007
	elevation	0.039	0.012	0.872	0.013	0.004
	azimuth	0.054	0.037	0.006	0.523	0.005
	color	0.022	0.010	0.007	0.013	0.910
	$H(f_k)$	1.7916	1.7917	1.7917	1.7917	1.7916

Table 6: The additional normalised MI score for the color intrinsic factor implies that this is the best disentangled generative factor. This can be also visually validated in 20 as the separation of the clusters corresponding to true color values is distinctive.

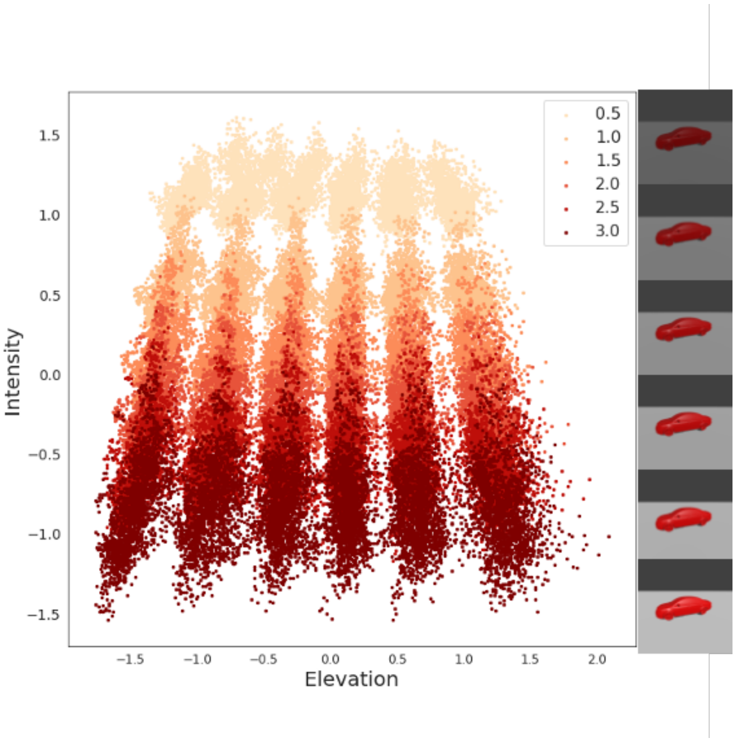

Figure 19: Latent space visualization of Intensity-Elevation *latent* factors. The coloring is based on true intensity values of the images represented by dots. The sequence of the images on the right side are the example images of the true intensity values. Each image denotes the cluster with a particular color.

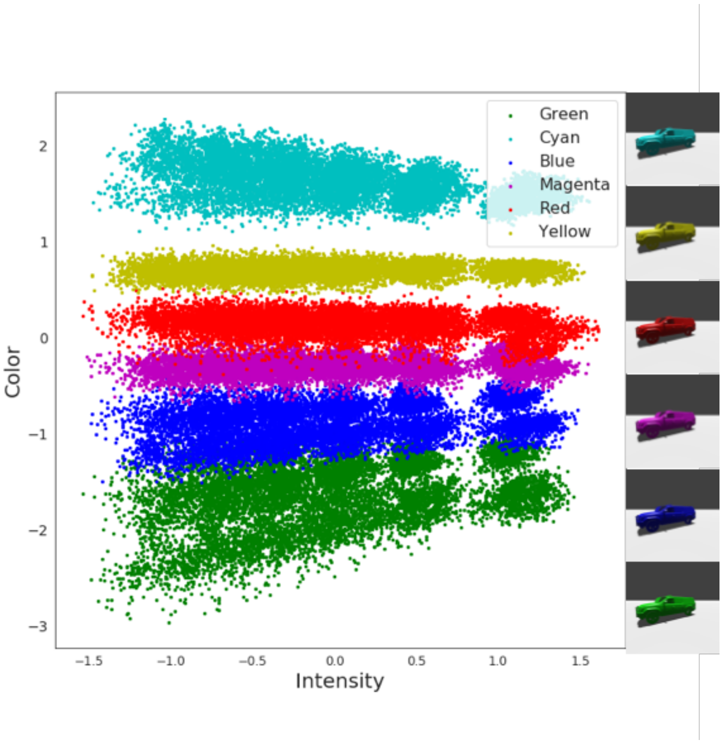

Figure 20: Latent space visualization of Color-Intensity *latent* factors. The coloring is based on true color values of the images represented by dots. The sequence of the images on the right side are the example images of the true color values. Each image denotes the cluster with a particular color.

8.2 Uncertainty Estimation

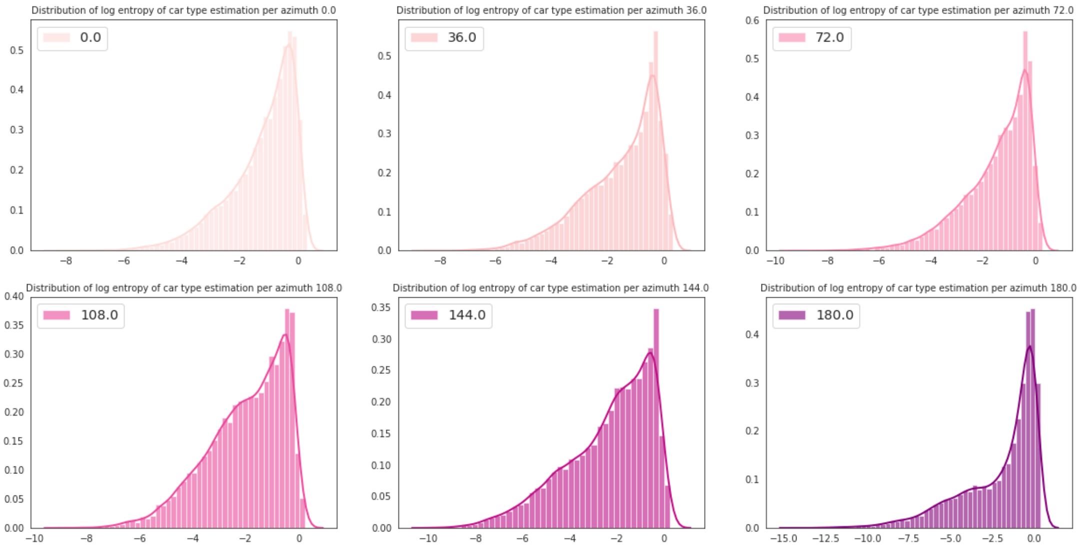

Figure 21: Distribution of Prediction Log-Entropy per Azimuth Angle. We use log entropy values to improve the visualisation since entropy values originally has exponential distribution.

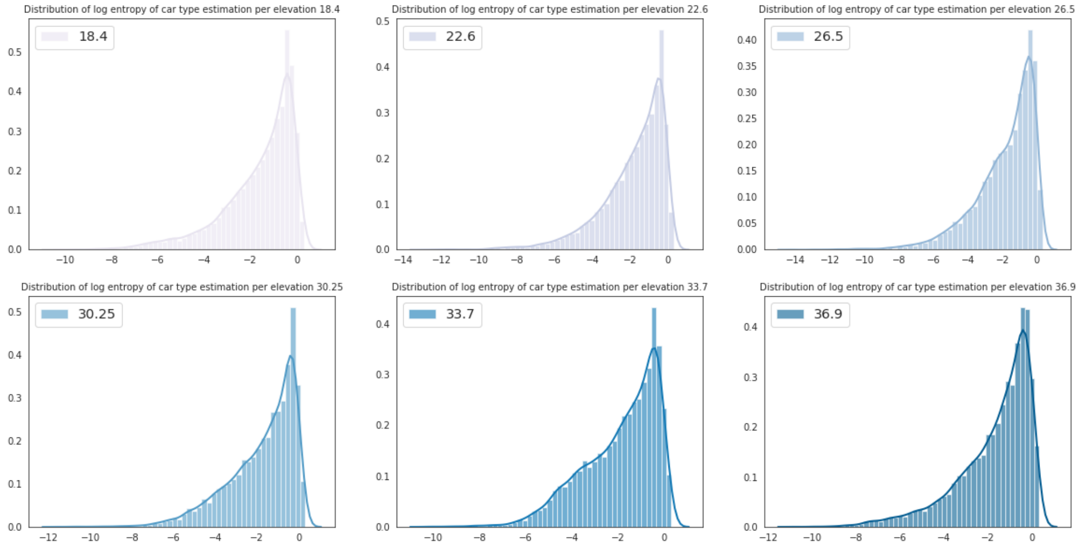

Figure 22: Distribution of Prediction Log-Entropy per Elevation Angle. We use log entropy values to improve the visualisation since entropy values originally has exponential distribution.

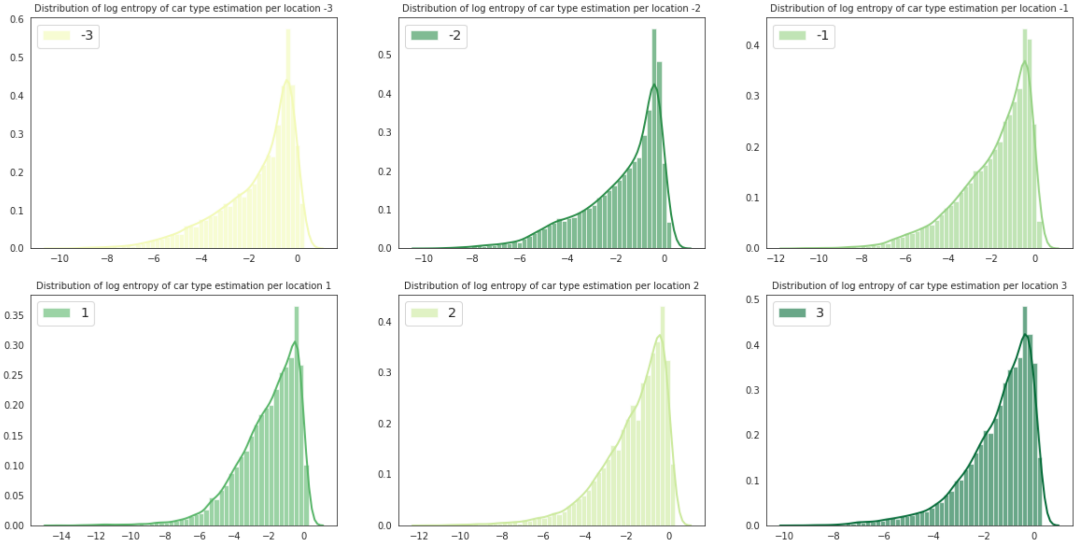

Figure 23: Distribution of Prediction Log-Entropy per Light Location. We use log entropy values to improve the visualisation since entropy values originally has exponential distribution.

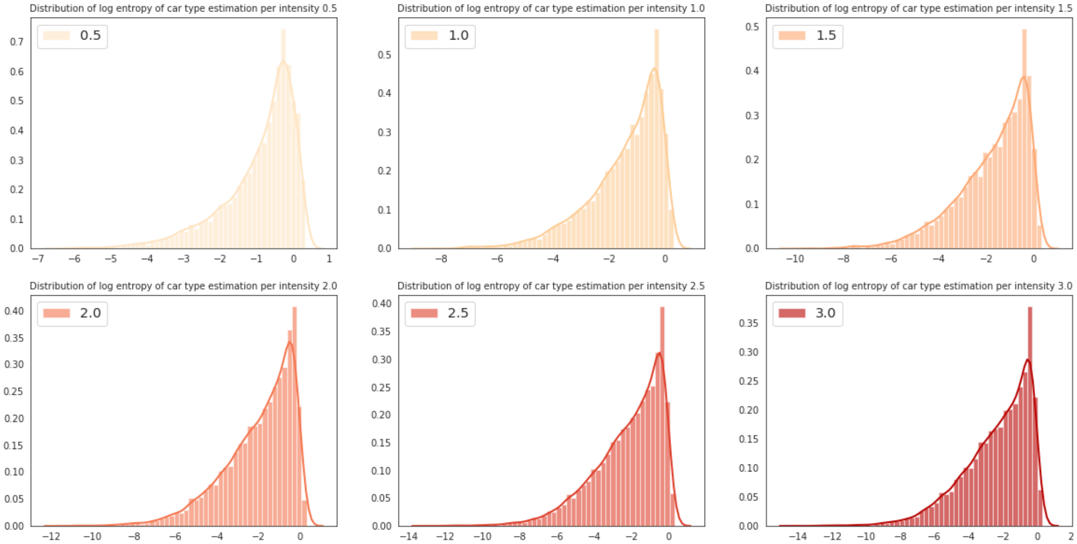

Figure 24: Distribution of Prediction Log-Entropy per Intensity Value. We use log entropy values to improve the visualisation since entropy values originally has exponential distribution.

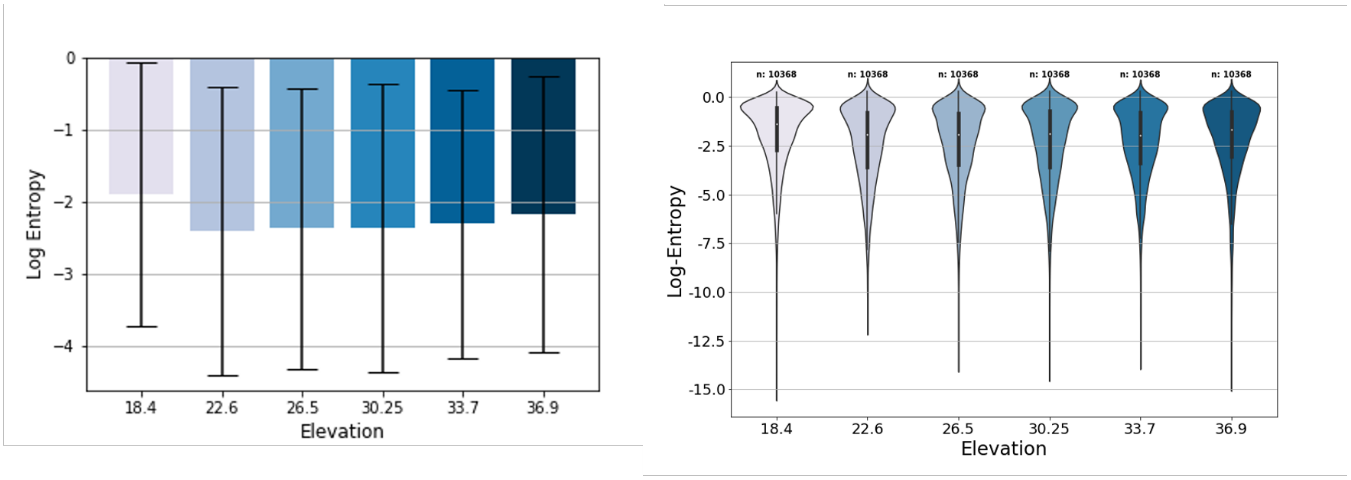

Figure 25: Error-bar and violin plots of log entropy of prediction for the elevation extrinsic factors

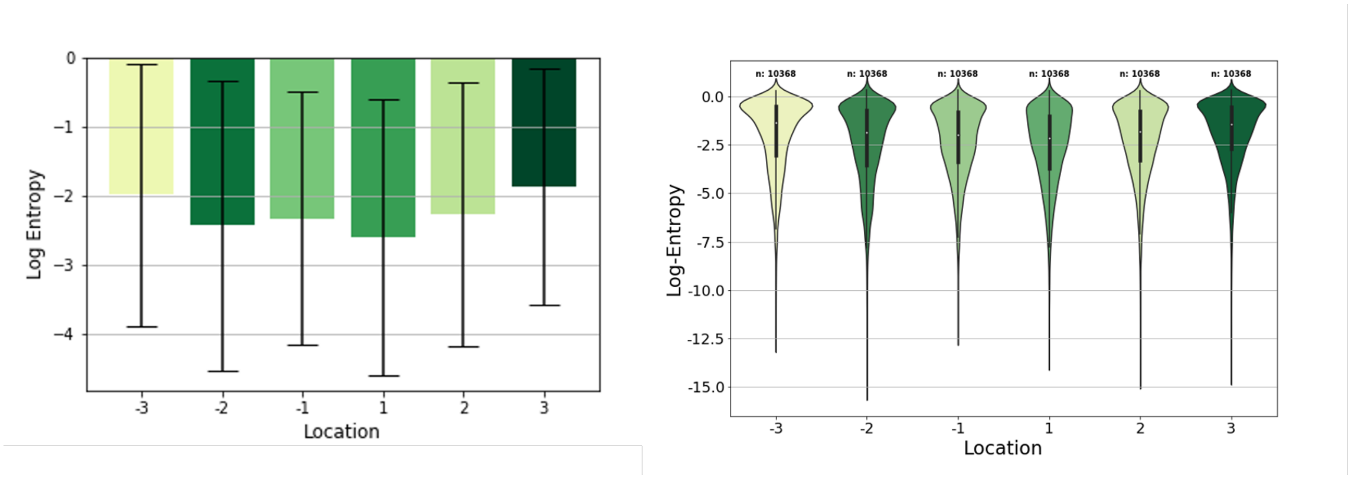

Figure 26: Error-bar and violin plots of log entropy of prediction for the location extrinsic factors

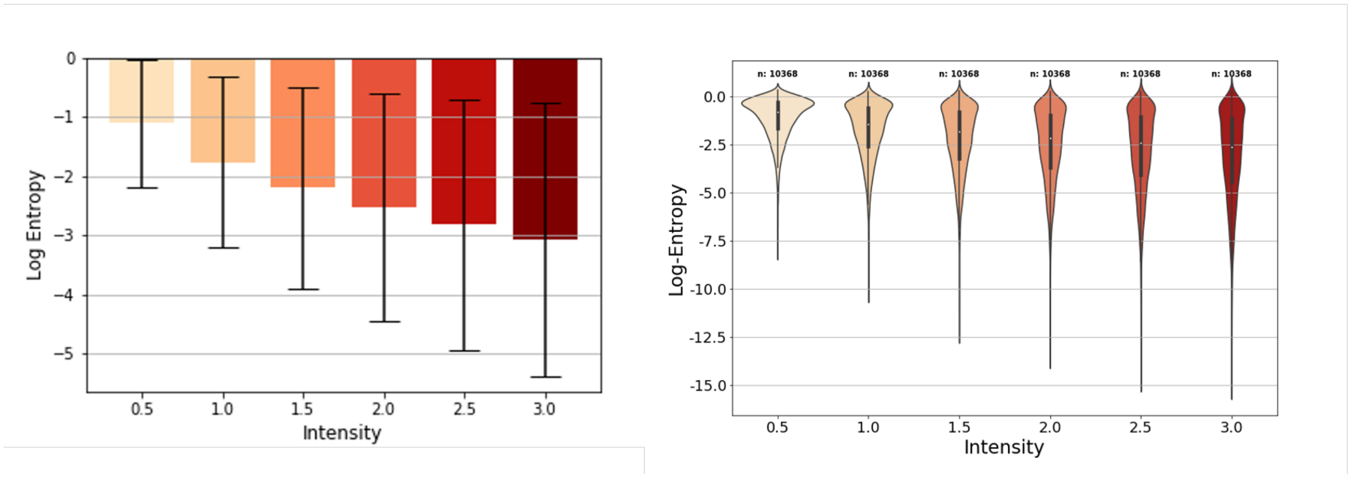

Figure 27: Error-bar and violin plots of log entropy of prediction for the intensity extrinsic factors

Elevation	18.4	22.6	26.5	30.25	33.7	36.9
18.4	-	0.000	0.000	0.000	0.000	0.045
22.6	0.000	-	0.198	0.000	0.000	0.000
26.5	0.000	0.198	-	0.000	0.000	0.000
30.25	0.000	0.000	0.000	-	0.000	0.000
33.7	0.000	0.000	0.000	0.000	-	0.000
36.9	0.045	0.000	0.000	0.000	0.000	-

Table 7: p-values of MWU test of elevation generative factor: The results indicate for *elevation* the prediction uncertainties of all angles are statistically different, except for angles 22.6° and 26.5°

Location	-3	-2	-1	1	2	3
-3	-	0.000	0.000	0.000	0.000	0.000
-2	0.000	-	0.000	0.000	0.142	0.000
-1	0.000	0.000	-	0.000	0.000	0.000
1	0.000	0.000	0.000	-	0.000	0.000
2	0.000	0.142	0.000	0.000	-	0.000
3	0.000	0.000	0.000	0.000	0.000	-

Table 8: p-values of MWU test of location generative factor: For *location*, only for -2 and 2, the test did not find statistically significant difference.

Intensity	0.5	1.0	1.5	2.0	2.5	3.0
0.5	-	0.000	0.000	0.000	0.000	0.000
1.0	0.000	-	0.000	0.000	0.000	0.000
1.5	0.000	0.000	-	0.000	0.000	0.000
2.0	0.000	0.000	0.000	-	0.000	0.000
2.5	0.000	0.000	0.000	0.000	-	0.000
3.0	0.000	0.000	0.000	0.000	0.000	-

Table 9: p-values of MWU Test of intensity generative factor: The differences of uncertainty distributions found significant between each pair of intensity values.

8.3 Inference Pipeline

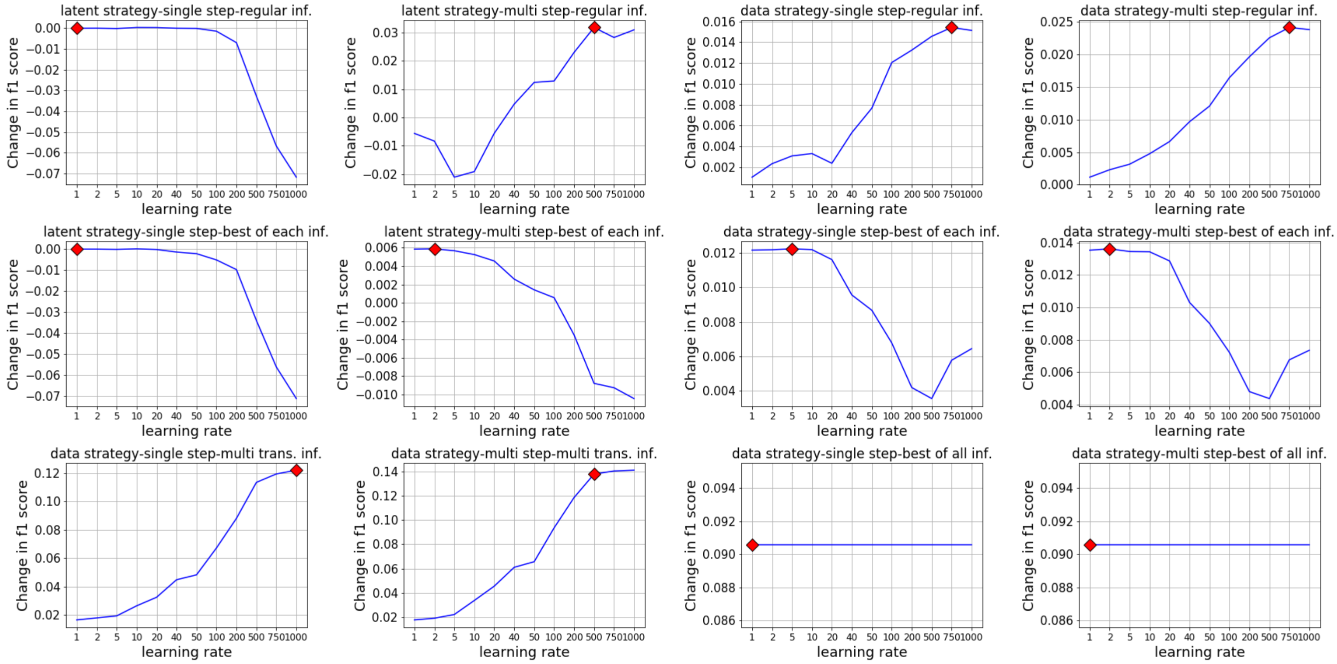

Figure 28: Inference results of each setting with various learning rates

Table 10: Inference results

settings	**learning rate**	**initial correct inference correct**	**initial correct inference wrong**	**initial wrong inference correct**	**initial wrong inference wrong**	**Δ Precision**	**Δ Recall**	**Δ f1**	**max. entropy reduction**	**avg. entropy reduction**	**max. inf. step**	**avg. inf. step**
Original classifier	-	*55444*	-	-	*6764*	*0.91*	*0.89*	*0.89*	-	-	-	-
latent strategy single step regular inf.	1	55442	2	0	6764	-0.000020	-0.000032	-0.000034	0.114021	0.003292	-	-
latent strategy multi step regular inf.	500	50718	4726	2049	4715	-0.041091	-0.043033	-0.045224	1.317900	-0.087570	10	1.31195
data strategy single step regular inf.	750	54097	1347	1748	5016	0.005335	0.006446	0.006041	1.248863	0.074173	-	-
data strategy multi step regular inf.	750	54300	1144	1687	5077	0.007627	0.008729	0.008238	1.317900	0.084894	4	0.429913
latent strategy single step best of each inf.	1	55413	31	19	6745	-0.000147	-0.000193	-0.000215	0.121739	0.007627	-	-
latent strategy multi step best of each inf.	2	53549	1895	2707	4057	0.003157	0.013053	0.012763	1.360743	0.068796	4	1.08062
data strategy single step best of each inf.	5	54679	765	1976	4788	0.013629	0.019467	0.019204	1.360743	0.107427	-	-
data strategy multi step best of each inf.	2	54738	706	2052	4712	0.015568	0.021637	0.021319	1.360743	0.122264	4	0.656829
data strategy single step multi trans. inf.	1000	54301	1143	5182	1582	0.048553	0.064927	0.065606	1.367100	0.118521	-	-
data strategy multi step multi trans. inf.	500	54489	955	5196	1568	**0.051756**	**0.068175**	**0.068890**	1.373707	0.131404	6	1.45086
data strategy single step best of all inf.	1	53132	2312	6243	521	0.052917	0.063191	0.063551	1.374287	0.208947	-	-
data strategy multi step best of all inf.	1	53132	2312	6243	521	0.052917	0.063191	0.063551	1.374287	0.208947	-	-