Article

TURBO: The Swiss Knife of Auto-Encoders

Guillaume Quétant * , **Yury Belousov** , **Vitaliy Kinakh** and **Slava Voloshynovskiy ***

Centre Universitaire d'Informatique, Université de Genève, Route de Drize 7, CH-1227 Carouge, Switzerland; yury.belousov@unige.ch (Y.B.); vitaliy.kinakh@unige.ch (V.K.)
* Correspondence: guillaume.quetant@unige.ch (G.Q.); svolos@unige.ch (S.V.)

Abstract: We present a novel information-theoretic framework, termed as TURBO, designed to systematically analyse and generalise auto-encoding methods. We start by examining the principles of information bottleneck and bottleneck-based networks in the auto-encoding setting and identifying their inherent limitations, which become more prominent for data with multiple relevant, physics-related representations. The TURBO framework is then introduced, providing a comprehensive derivation of its core concept consisting of the maximisation of mutual information between various data representations expressed in two directions reflecting the information flows. We illustrate that numerous prevalent neural network models are encompassed within this framework. The paper underscores the insufficiency of the information bottleneck concept in elucidating all such models, thereby establishing TURBO as a preferable theoretical reference. The introduction of TURBO contributes to a richer understanding of data representation and the structure of neural network models, enabling more efficient and versatile applications.

Keywords: information bottleneck; TURBO; generalisation; auto-encoder; variational approximation; lower bound; mutual information; physical latent space; representations; Kullback–Leibler divergence

1. Introduction

Over the past few years, many deep learning architectures have been proposed and successfully applied to a wide range of problems. However, they are often developed from empirical observations and their theoretical foundations are still not well enough understood. Typical examples of deep learning architectures that have been widely used and revisited multiple times are generative adversarial networks (GANs) [1], variational auto-encoders (VAEs) [2,3] and adversarial auto-encoders (AAEs) [4]. A rigorous interpretation of the concepts and principles behind such machine learning methods is crucial to understanding their strength and limitations, and to guiding the development of new models. A concrete formulation of these concepts, unifying as many models as possible, would be a huge gain for the field.

Showing a promising path towards this goal, bottleneck formulations of neural network training have been extensively studied in many theoretical and experimental works [5–9]. They are based on the fact that one may want to preserve as much relevant information as possible from a given input, removing all unnecessary knowledge. This is called the information bottleneck (IBN) principle [10] and it has a crucial impact in several applications.

It was originally developed to characterise what can be termed "relevant" information in the context of supervised learning [10]. The IBN principle was introduced as an extension to the so-called *rate-distortion* theory [11], leaving the choice of the distortion function open and giving an iterative algorithm for finding the optimal compressed representation of the data. This first description of the IBN principle paves the way for machine-learning-oriented formulations of supervised learning [5], as well as for the bounded information bottleneck auto-encoder (BIB-AE) framework [12] in the context of unsupervised learning. The BIB-AE framework shows that many bottleneck architectures, especially concerning

the VAE family, are generalised by this approach. It can also be used for semi-supervised learning [7], where the framework allows the impact of several well-known techniques to be studied in a better and more interpretable way. More recently, the IBN principle has been used as an attempt to explain the success of self-supervised learning (SSL) [13,14]. In this context, the bottleneck manifests itself as a compression of information between the learnt representation and the distorted image, usually obtained by applying augmentations to the original input. This is achieved by minimising the mutual information between the representation and the distorted image, while maximising the mutual information between the representation and the original image. However, a key limitation rises from this formulation of SSL, since it is largely based on the assumption that all the relevant information for a given downstream task is shared by both the original and the distorted images. If this is not the case, which may occur whenever one replaces the distorted image by a second meaningful representation of the data, the IBN principle struggles to provide a satisfactory explanation of SSL.

The IBN principle represents a significant theoretical advance towards explainable machine learning, but it has several inherent limitations [8]. For example, while the IBN provides a solid and comprehensive theoretical interpretation for the VAE family of models, it can only address the GAN family with an intricate formulation and it encounters difficulties in providing a compelling justification for the AAE family. The core issue emerges when the objective is not to achieve maximum disentanglement of the latent space from the input data, a scenario that is particularly salient with AAE models. The training of an AAE encoder is oriented to maximise the informational content shared between the input data and the latent representation, in direct contradiction to the premises of the IBN principle. This shift in approach is not simply a mathematical manoeuvre to articulate new models: it fundamentally alters the desired methodology of data interpretation. When dealing with input data that admit two or more relevant representations, and where one representation is designated as a physically meaningful latent space, it becomes natural to construct a framework that maximises the mutual information between these two highly correlated modalities.

On top of AAE type models, several other architectures cannot be interpreted via the IBN principle, but rather need a new paradigm. Image-to-image translation models such as pix2pix [15] and CycleGAN [16] played an important role in the development of machine learning. As we will demonstrate, they directly fall into the category of models that maximise the mutual information between two representations of the data, which is a case that the IBN principle fails to address. Furthermore, normalising flows [17] suffer from the same intricate formulation as GANs when attempting to explain them in terms of the IBN principle. Additionally, other models such as ALAE [18] do not show a bottleneck architecture and are thus unconvincingly interpretable via the IBN. All these points speak for the necessity of developing a new framework capable of addressing and explaining these methods theoretically.

In this work, we present a powerful formalism called Two-way Uni-directional Representations by Bounded Optimisation (TURBO) that complements the IBN principle, giving a rigorous interpretation of an additional wide range of neural network architectures. It is based on the maximisation of the mutual information between various random variables involved in an auto-encoder architecture.

The structure of the paper is following this logic: in Section 2, we define a unified language in order to make the understanding and the comparison of the various frameworks clearer. In Section 3, we explain the BIB-AE framework using these notations, exposing its main advantages and drawbacks, and the reasons for considering the TURBO alternative. In Section 4, we detail the TURBO framework and its generalisation power. In Section 5, we highlight successful applications of TURBO for solving several practical tasks. For the ease of reading, Table 1 shows a summary comparison of the BIB-AE and the TURBO frameworks. The reader who is already familiar with the IBN principle is advised to proceed directly to Section 3.2.

Table 1. A summary of the main differences between the BIB-AE and the TURBO frameworks.

	BIB-AE	TURBO
Paradigm	**Minimising** the mutual information between the input space and the latent space, while maximising the mutual information between the latent space and the output space	**Maximising** the mutual information between the input space and the latent space, and maximising the mutual information between the latent space and the output space
	One-way encoding	**Two-way** encoding
	Data and latent space distributions are considered **independently**	Data and latent space distributions are considered **jointly**
Targeted tasks	• Data compression, privacy, classification • Representation learning	• Linking relevant modalities • Transcoding/translation between modalities
Advantages	• Theoretical basis for both supervised and un-supervised tasks • Allows for easy sampling	• Interpretable latent space • Seamlessly handles paired, unpaired and partially paired data • The encoder can represent a physical system, while the decoder can represent a learnable model
Drawbacks	• Not suited for data translation • Enforces a distribution for the latent space • Struggles to map discontinuous data distributions to continuous latent space distributions	• More hyperparameters to tune • More modules increases training complexity
Particular cases	VAE, GAN, VAE/GAN	AAE, GAN, pix2pix, SRGAN, CycleGAN, Flows
Related models	InfoVAE, CLUB	ALAE

Our main contributions in this work are:

- Highlighting the main limitations of the IBN principle and the need for a new framework;
- Introducing and explaining the details of the TURBO framework, and motivating several use cases;
- Reviewing well-known models with the lens of the TURBO framework, showing how it is a straightforward generalisation of them;
- Linking the TURBO framework to additional related models, opening the door to additional studies and applications;
- Showcasing several applications where the TURBO framework gives either state-of-the-art or competing results compared to existing methods.

2. Notations and Definitions

Before discussing the various frameworks that we study in this work, we define a common basis for the notations. Since most of the considered models will be viewed as auto-encoders or as parts of auto-encoders, we shape our notations to fit this framework. Most of the quantities found throughout the paper are defined below and a table summarising all symbols and naming used can be found in Appendix A.

We consider two random variables $\mathbf{X}$ and $\mathbf{Z}$ with marginal distributions $\mathbf{X} \sim p(\mathbf{x})$ and $\mathbf{Z} \sim p(\mathbf{z})$, respectively, and either a known or unknown joint distribution $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z}) \, p(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}) \, p(\mathbf{x})$. Notice that $\mathbf{X}$ and $\mathbf{Z}$ can be independent variables, in which case their joint distribution simplifies to the product of their marginal distributions. The two unknown conditional distributions can be parametrised by two neural networks, usually called *encoder* $q_\phi(\mathbf{z}|\mathbf{x}) \approx p(\mathbf{z}|\mathbf{x})$ and *decoder* $p_\theta(\mathbf{x}|\mathbf{z}) \approx p(\mathbf{x}|\mathbf{z})$, where the parameters of the networks are generically denoted by ϕ and θ. Once chained together as shown in

Figure 1, they form a so-called *auto-encoder*. We thus define two approximations of the joint distribution

$$q_\phi(\mathbf{x}, \mathbf{z}) := q_\phi(\mathbf{z}|\mathbf{x}) \overbrace{p(\mathbf{x})}^{\substack{\text{real} \\ \text{data}}} = q_\phi(\mathbf{x}|\mathbf{z}) \overbrace{\tilde{q}_\phi(\mathbf{z})}^{\substack{\text{synthetic} \\ \text{data}}}, \tag{1}$$

$$p_\theta(\mathbf{x}, \mathbf{z}) := \underbrace{p_\theta(\mathbf{x}|\mathbf{z})}_{\substack{\text{known} \\ \text{networks}}} p(\mathbf{z}) = \underbrace{p_\theta(\mathbf{z}|\mathbf{x})}_{\substack{\text{unknown} \\ \text{networks}}} \tilde{p}_\theta(\mathbf{x}), \tag{2}$$

where the rightmost expressions are reparametrisations with unknown networks and the approximated marginal distributions $\tilde{q}_\phi(\mathbf{z}) = \int q_\phi(\mathbf{x}, \mathbf{z}) \, d\mathbf{x}$ and $\tilde{p}_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z}) \, d\mathbf{z}$ corresponding to the latent spaces synthetic variables. Two approximated marginal distributions, also used throughout this work, corresponding to the reconstructed spaces synthetic variables can be defined as $\hat{q}_\phi(\mathbf{z}) = \int \tilde{p}_\theta(\mathbf{x}) \, q_\phi(\mathbf{z}|\mathbf{x}) \, d\mathbf{x}$ and $\hat{p}_\theta(\mathbf{x}) = \int \tilde{q}_\phi(\mathbf{z}) \, p_\theta(\mathbf{x}|\mathbf{z}) \, d\mathbf{z}$.
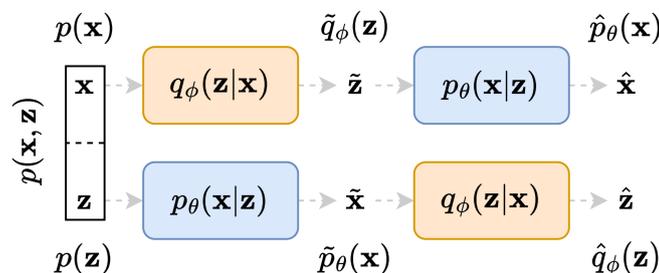
Figure 1. All considered random variable manifolds and the related notations for their probability distributions. The upper part of the diagram is an auto-encoder for the random variable $\mathbf{X}$ while the lower part is a symmetrical formulation for the random variable $\mathbf{Z}$. The two random variables $\mathbf{X}$ and $\mathbf{Z}$ might be independent, so $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}) \, p(\mathbf{z})$.

Since mutual information between different variables is extensively used in this paper, we give a brief definition of it. We also showcase in Figure 2 the notations that we employ when computing the mutual information between diverse random variables, corresponding to diverse information flows in the networks. The mutual information for two random variables $\mathbf{X}$ and $\mathbf{Z}$ following a joint distribution $p(\mathbf{x}, \mathbf{z})$ is defined as

$$I(\mathbf{X}; \mathbf{Z}) := \mathbb{E}_{p(\mathbf{x}, \mathbf{z})} \left[\log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x}) p(\mathbf{z})} \right], \tag{3}$$

where $\mathbb{E}[\cdot]$ is the mathematical expectation with respect to the given distribution $p(\mathbf{x}, \mathbf{z})$ and where $p(\mathbf{x})$ and $p(\mathbf{z})$ denote the corresponding marginal distributions. Therefore, to exemplify our notations, the mutual information between $\mathbf{X}$ and the random variable $\tilde{\mathbf{Z}}$ defined by the marginalisation of the encoder $q_\phi(\mathbf{z}|\mathbf{x})$ outputs $\tilde{\mathbf{Z}} \sim \tilde{q}_\phi(\mathbf{z})$ would be defined by $I_\phi(\mathbf{X}; \tilde{\mathbf{Z}}) = \mathbb{E}_{q_\phi(\mathbf{x}, \mathbf{z})}[\log q_\phi(\mathbf{x}, \mathbf{z})/p(\mathbf{x})\tilde{q}_\phi(\mathbf{z})]$. On the other hand, in order to compute the mutual information between the random variable defined by the marginalisation of the decoder $p_\theta(\mathbf{x}|\mathbf{z})$ outputs $\tilde{\mathbf{X}} \sim \tilde{p}_\theta(\mathbf{x})$ and $\mathbf{Z}$, its expression would be $I_\theta(\tilde{\mathbf{X}}; \mathbf{Z}) = \mathbb{E}_{p_\theta(\mathbf{x}, \mathbf{z})}[\log p_\theta(\mathbf{x}, \mathbf{z})/\tilde{p}_\theta(\mathbf{x})p(\mathbf{z})]$.

$$I_\phi(\mathbf{X}; \tilde{\mathbf{Z}})$$

$$I_\theta(\tilde{\mathbf{X}}; \mathbf{Z})$$
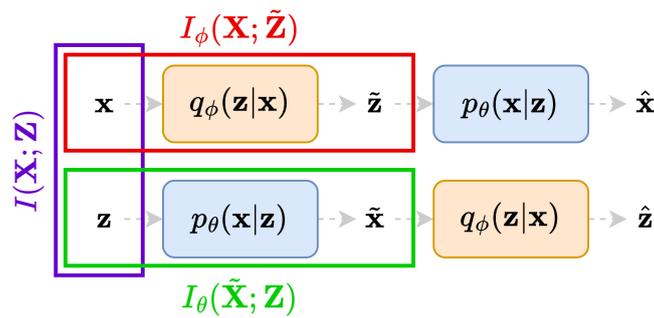
Figure 2. The notations for the mutual information computed between different random variables. The leftmost purple rectangle highlights the true mutual information between $\mathbf{X}$ and $\mathbf{Z}$. The upper red and the lower green rectangles highlight the mutual information when the joint distribution is approximated by $q_\phi(\mathbf{x}, \mathbf{z})$ and $p_\theta(\mathbf{x}, \mathbf{z})$, respectively.

Lastly, we use several other common information-theoretic quantities such as the Kullback–Leibler divergence (KLD) between two distributions denoted by $D_{\mathrm{KL}}(\cdot\|\cdot)$, the entropy of a distribution denoted by $H(\cdot)$, the conditional entropy of a distribution denoted by $H(\cdot|\cdot)$ and the cross-entropy between two distributions denoted by $H(\cdot; \cdot)$.

3. Background: From IBN to TURBO

In this section, for the completeness of our analysis, we briefly review the BIB-AE framework [12], which addresses a variational formulation of the IBN principle for an auto-encoding setting. We then redefine the two founding models at the root of many deep learning studies, the so-called VAE and GAN, in order to unite them under the BIB-AE umbrella and our notations. Finally, we explain why this formulation, even though well suited to many problems and showing several advantages, is not applicable to applications with a physical latent space where the data compression is not needed as such.

3.1. Min-Max Game: Or Bottleneck Training

The IBN is based on a compression principle where all targeted task-irrelevant information should be filtered out from the input data, i.e., the input data are *compressed*. The difference with the classical compression addressed in the rate-distortion theory, which ensures the best source reconstruction under a given rate, is that the IBN compressed data contain only sufficient statistics for a given downstream task. The main targeted application of the IBN is classification [5,19], where the intermediate or latent representation contains only the information related to the provided class labels. Recently, the IBN has also been extended to cover a broad family of privacy-related issues through the complexity–leakage–utility bottleneck (CLUB) [20]. For example, it can be used to disentangle some sensitive data from a latent representation in such a way that no private information remains, while useful features are still available for downstream tasks. Bottleneck networks are also used in anomaly detection in order to get rid of all the useless features contained in the data, keeping only what helps in identifying background from signal [21–23]. A parallel usage of such models is for the generation of new data from a given manifold [24,25]. Indeed, as the encoded latent space should ideally be a disentangled representation of the data, it might be shaped towards any desired random distribution. Typically, this distribution would be Gaussian so that one can generate new samples by passing Gaussian noise through the trained decoder, which should have learnt to recover the data manifold from the minimal amount of information kept by such latent representation. More generally, formalisms derived from the IBN principle describe ways to create a mapping between some input data and a defined distribution.

The BIB-AE framework formulated for both unsupervised [12] and semi-supervised settings [7] is a variational formulation of the IBN principle [5]. It relies on a trade-off

between the amount of information lost when encoding data into a latent space and the amount of information kept for proper decoding from this latent space. It can be formally expressed as an optimisation problem where one tries to balance minimisation and maximisation of the mutual information between the data space random variable $\mathbf{X}$ and the latent space random variable $\tilde{\mathbf{Z}}$. Therefore, the BIB-AE loss has the form

$$\mathcal{L}^{\text{BIB-AE}}(\phi, \theta) = I_\phi(\mathbf{X}; \tilde{\mathbf{Z}}) - \lambda_B \mathcal{I}^{\text{x}}_{\phi,\theta}(\mathbf{X}; \tilde{\mathbf{Z}}), \tag{4}$$

where $\mathcal{I}^{\text{x}}_{\phi,\theta}(\mathbf{X}; \tilde{\mathbf{Z}})$ is a parametrised lower bound to $I_\phi(\mathbf{X}; \tilde{\mathbf{Z}})$ as demonstrated in Appendix B. The positive weight λ_B controls the trade-off between the minimisation and maximisation of the two terms. Once Equation (4) expands, the resulting loss becomes

$$\mathcal{L}^{\text{BIB-AE}}(\phi, \theta) = \mathbb{E}_{p(\mathbf{x})}\left[D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))\right] - D_{\text{KL}}(\tilde{q}_\phi(\mathbf{z})\|p(\mathbf{z})) \\ - \lambda_B \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] + \lambda_B D_{\text{KL}}(p(\mathbf{x})\|\hat{p}_\theta(\mathbf{x})). \tag{5}$$

Notice that we intentionally abuse the notation of $D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{X} = \mathbf{x})\|p(\mathbf{z}))$ here and in other analogous expressions in order to lighten the equations since the expectations always solve the ambiguity. The full derivation of Equation (5) is given in Appendix B. Next, we showcase how the BIB-AE formalism can be related to the VAE and GAN families.

3.1.1. VAE from BIB-AE Perspectives

In the wide range of machine learning methods, VAE [2,3] has a particular place as it is among the first deep learning generative models developed. VAE has been extensively studied to decorrelate a given data space, mapping it to a Gaussian latent distribution while reconstructing back the data space from it. This is a typical example of the IBN principle and therefore it easily falls under the BIB-AE formalism. Keeping only the first and third terms of Equation (5) directly leads to the VAE loss

$$\mathcal{L}^{\text{VAE}}(\phi, \theta) = \mathbb{E}_{p(\mathbf{x})}\left[D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))\right] - \lambda_B \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})], \tag{6}$$

where we already allow for the weight λ_B to appear in order to generalise to a broader family called β-VAE [26]. In the literature, the λ_B weight is often written β, hence the name β-VAE. The usual VAE loss would correspond to $\lambda_B = 1$.

Typically, the encoder is designed in such a way as to output two values that are used to parametrise the mean and the variance of a Gaussian distribution from which latent points are drawn. Therefore, the conditional latent space distribution $q_\phi(\mathbf{z}|\mathbf{x})$ is a conditional Gaussian distribution. For $p(\mathbf{z})$ chosen to be a standard normal distribution, the KLD term has a closed form and can be exactly and efficiently computed. For the second term of the loss, it is typical to assume that the conditional decoded space shows exponential deviations from the original data leading to $\log p_\theta(\mathbf{x}|\mathbf{z}) = -\alpha\|\hat{\mathbf{x}} - \mathbf{x}\|_p^p + C$, where α and C are positive constants.

A successful extension called information maximising VAE (InfoVAE) [27] proposes to add the mutual information term $I_\phi(\mathbf{X}; \tilde{\mathbf{Z}})$ to the VAE objective and to maximise it. This is exactly the first term of the BIB-AE loss in Equation (4), which is, however, minimised, so that the InfoVAE objective is actually counter-balancing it. The expression of the InfoVAE loss is given by the first, second and third terms of Equation (5) as well as an additional weight factor λ_{Info}

$$\mathcal{L}^{\text{InfoVAE}}(\phi, \theta) = \mathbb{E}_{p(\mathbf{x})}\left[D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))\right] - (1 - \lambda_B\lambda_{\text{Info}})D_{\text{KL}}(\tilde{q}_\phi(\mathbf{z})\|p(\mathbf{z})) \\ - \lambda_B \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})], \tag{7}$$

where the usual VAE loss would correspond to $\lambda_B = \lambda_{\text{Info}} = 1$. Notice that, in the original InfoVAE loss [27], the weights are denoted by α and λ, with $\lambda_B = 1/(1-\alpha)$ and $\lambda_{\text{Info}} = \lambda$, where α is the factor in front of the added mutual information term $I_\phi(\mathbf{X}; \tilde{\mathbf{Z}})$ and where λ is

artificially inserted. It should be emphasised that, for positive λ_B and λ_{Info}, the InfoVAE loss is not generalised by the BIB-AE formalism. The additional $\lambda_B \lambda_{\text{Info}} D_{\text{KL}}(\tilde{q}_\phi(\mathbf{z}) \| p(\mathbf{z}))$ term is, however, very similar to a term that we will later find in the TURBO formalism, which is due to its mutual information maximisation origin. The success of the InfoVAE framework is a strong incentive for the development of the TURBO formalism.

3.1.2. GAN from BIB-AE Perspectives

On the other side of the widely used yet very related deep learning generative models stand GANs [1]. The principle of a basic GAN is simple as it may be summarised by a decoder network that is trained to map a random input latent sample to an output sample compatible with the data. Typically, the latent space again follows a Gaussian distribution to ensure simple sampling, similar to VAEs. However, since there is no encoder in the usual GAN formulation, there is no need to include any shaping of the latent space by means of some loss terms. The sole role of the decoder is to transform the Gaussian distribution into the data distribution $p(\mathbf{x})$. Therefore, the training objective of a GAN can be expressed as the fourth term of Equation (5)

$$\mathcal{L}^{\text{GAN}}(\theta) = D_{\text{KL}}(p(\mathbf{x}) \| \hat{p}_\theta(\mathbf{x})), \tag{8}$$

where we omit the λ_B weight as it has no impact here. This term ensures the closeness of the true data distribution $p(\mathbf{x})$ and the generated data distribution $\hat{p}_\theta(\mathbf{x})$.

In contrast to the KLD term in the latent space of the VAE formulation, the GAN loss cannot be expressed in a closed form because the data distribution $p(\mathbf{x})$ that the model tries to approximate with $\hat{p}_\theta(\mathbf{x})$ is by definition unknown. Facing an intractable KLD is unfortunately a common scenario in machine learning optimisation problems. In practice, the loss is usually replaced by any differentiable metric suited to the comparison of two distributions using samples from both. This includes, but is not limited to, density ratio estimation through a discriminator network and optimal transport through Wasserstein distance approximations. We refer the interested reader to [20,28] for an overview of different methods that allow us to tackle this problem in a practical way.

It is worth noting that, in principle, the reconstructed marginal distribution $\hat{p}_\theta(\mathbf{x})$ involves the latent marginal distribution $\tilde{q}_\phi(\mathbf{z})$ in its definition. However, since there is no encoder network in the case of a GAN, it must be understood as the true latent distribution $\tilde{q}_\phi(\mathbf{z}) \equiv p(\mathbf{z})$, which would correspond to a perfect encoder $q_\phi(\mathbf{z}|\mathbf{x}) \equiv p(\mathbf{z}|\mathbf{x})$. The TURBO formalism will later provide deeper insights into GANs.

For the sake of completeness, we also quote here the hybrid VAE/GAN loss [29]. It is hybrid in the sense that it uses both the VAE-like latent space regularisation and the GAN-like reconstruction space distribution matching. The VAE/GAN loss can be expressed as the first, third and fourth terms of Equation (5):

$$\begin{aligned}
\mathcal{L}^{\text{VAE/GAN}}(\phi, \theta) = {} & \mathbb{E}_{p(\mathbf{x})}\left[D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) \right] \\
& - \lambda_B \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] + \lambda_B D_{\text{KL}}(p(\mathbf{x}) \| \hat{p}_\theta(\mathbf{x})),
\end{aligned} \tag{9}$$

where we do not take into account the refinement of the reconstruction error term proposed in the original work. Indeed, the VAE/GAN loss replaces the likelihood term $p_\theta(\mathbf{x}|\mathbf{z})$ with a Gaussian distribution on the outputs of the hidden layers of the discriminator network.

3.1.3. CLUB

Another noticeable extension of BIB-AE is the CLUB model [20]. This model extends the IBN principle by providing a unified generalised framework for information-theoretic privacy models, and by establishing a new interpretation of popular generation, discrimination and compression models. Using our notations, the CLUB objective function can be expressed as

$$\mathcal{L}^{\text{CLUB}}(\phi, \theta, \theta_s) = I_\phi(\mathbf{X}; \tilde{\mathbf{Z}}) - \lambda_B \mathcal{I}^{\text{x}}_{\phi,\theta}(\mathbf{X}; \tilde{\mathbf{Z}}) + \lambda_S \mathcal{I}^{\text{s}}_{\phi,\theta_s}(\mathbf{S}; \tilde{\mathbf{Z}}), \tag{10}$$

which is the BIB-AE loss of Equation (4) plus the additional privacy term $\lambda_S\,\mathcal{I}^s_{\phi,\theta_s}(\mathbf{S};\tilde{\mathbf{Z}})$ controlled by the positive weight λ_S. The $\mathbf{S}$ variable denotes the *sensitive attribute* linked to the data $\mathbf{X}$, which has to be kept secret for privacy purposes. The additional minimised term $\mathcal{I}^s_{\phi,\theta_s}(\mathbf{S};\tilde{\mathbf{Z}})$ is an upper bound to the mutual information between $\mathbf{S}$ and $\tilde{\mathbf{Z}}$, taking into account an attacker network with parameters θ_s, which aims at inferring the sensitive attribute from the latent variable $\tilde{\mathbf{Z}}$. The trade-off between the three terms of the CLUB loss can be understood as an attacker–defender game. Indeed, the first term tries to globally reduce the shared information between the input data and the latent space, while the second and third terms compete in order to keep and get rid of, respectively, the information needed to reconstruct $\hat{\mathbf{X}}$ from $\tilde{\mathbf{Z}}$ and to infer $\hat{\mathbf{S}}$ from $\tilde{\mathbf{Z}}$.

3.2. Max-Max Game: Or Physically Meaningful Latent Space

So far, we have presented the IBN principle, whose main objective is to find the optimal way to compress data into a latent representation while preserving enough information for the task at hand. What if one does not need any compression of the data? Alternatively, what if the latent space should not be Gaussian in order to facilitate the compression and the sampling? In other words, what if the latent space does not have to be *latent*? Indeed, except if partially removing the information contained in the data serves a particular purpose, one could want to retain all the information and just map the data to a latent space of the desired shape. More precisely, one can get rid of the trade-off expressed in Equation (4) and train a network to only maximise the mutual information between the data and latent spaces. It should be noticed that the oxymoronic phrasing, namely a *physical latent space*, is used to emphasise the questions raised here. Since it is a common name for the intermediate representation in an auto-encoder setting, we choose to keep using the word *latent* for denoting this space.

An AAE [4], for example, is based on an adversarial loss computed in the latent space, very much like the GAN loss in the data space. This is a first hint that this loss term might come from the maximisation of mutual information rather than minimisation. It can also be understood as the second term of Equation (5) but with the opposite sign. This is a second hint that goes towards the same conclusion. In the following section, we will detail the TURBO formalism, which will provide a rigorous explanation of AAEs where the BIB-AE formulation fails to do so.

The formalisms that do not show a bottleneck architecture are also highly relevant in several contexts. Indeed, in many cases, the latent space should not be seen as a mere compression of the data but rather as a physically meaningful representation of them. Figure 3 shows a comparison of two auto-encoders settings, one with a virtual latent space and one with a physical latent space. For example, a virtual latent variable could represent a class label, while a physical latent variable would be a noisy picture captured by some camera. In the physical setting, the complex yet unknown physical measurement is approximated by the parametrised network corresponding to the encoder $q_\phi(\mathbf{z}|\mathbf{x})$, which produces the observation $\tilde{\mathbf{z}}$ as output. The parametrised decoder $p_\theta(\mathbf{x}|\mathbf{z})$ reconstructs back $\hat{\mathbf{x}}$ or converts it to a suitable form for further analysis or storage. The latent space variable $\tilde{\mathbf{z}}$ is thus defined by the physics of the experiment rather than being fixed to follow a Gaussian, a categorical or any other simple distribution.

One may even compose these settings as shown in Figure 4. For example, the latent variable $\tilde{\mathbf{z}}$ can be understood as a generic name encapsulating both a virtual $\tilde{\mathbf{z}}_v$ and a physical $\tilde{\mathbf{z}}_p$ meaning. On the other hand, the encoder and decoder themselves could be composed of their own auto-encoder-like networks, each processing some kind of internal latent variable $\mathbf{y}_e$ and $\mathbf{y}_d$, respectively. Combining the BIB-AE and the TURBO formalisms in order to create such deep networks can lead to a very broad family of models.
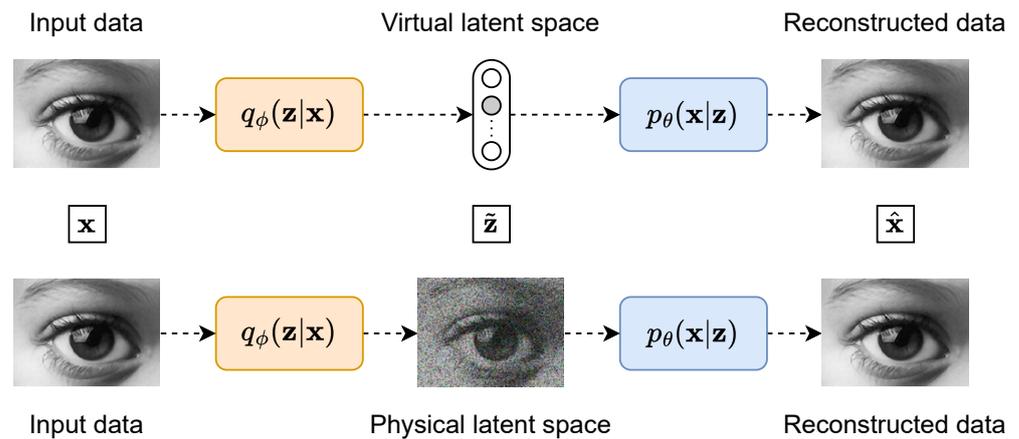
Figure 3. Auto-encoders with a virtual latent space or a physical latent space. In the virtual setting, the latent variable z̃ does not have any physical meaning, while in the physical setting, this latent variable represents a part of the physical observation/measurement chain.
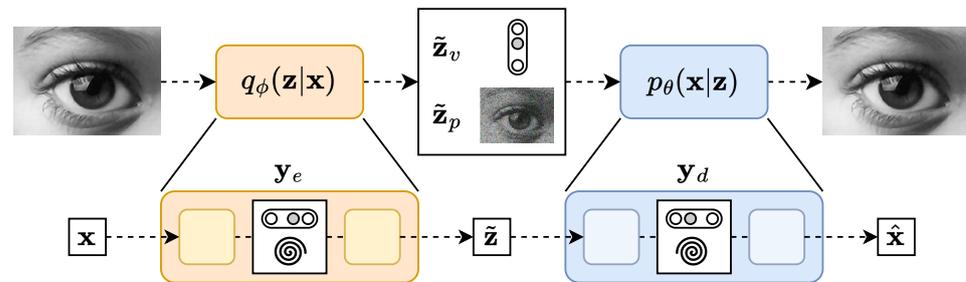
Figure 4. Composition of different latent space settings and several auto-encoder-like networks as internal components of a global auto-encoder architecture. The global latent variable $\tilde{\mathbf{z}}$ can contain both virtual $\tilde{\mathbf{z}}_v$ and physical $\tilde{\mathbf{z}}_p$ parts. The global encoder and decoder can be nested auto-encoders with internal latent variables of any kind $\mathbf{y}_e$ and $\mathbf{y}_d$, respectively.

Treating the latent variable as a second representation of the data in a different space is useful for several applications, as sketched in Figure 5. This is the case, for example, with the optimal-transport-based unfolding and simulation (OTUS) method [30] in the context of high-energy physics. Here, the latent space is the four-momenta of the particles produced in a proton–proton collider experiment, while the data space is the detector response to the passage of these particles. A sketch of the two representations is shown in the top row of Figure 5.

Another example might be the image-to-image translation of a given portion of the sky pictured by two different telescopes. In this case, the latent and data spaces are both images, but one is the image of the sky as seen by the first telescope (e.g., the Hubble Space Telescope), while the other is the image of the sky as seen by the second telescope (e.g., the James Webb Space Telescope). A sketch of the two representations is shown in the middle row of Figure 5.

A third example of a problem where two representations of the data are available is the analysis of copy detection patterns (CDPs) for anti-counterfeiting applications. Templates of CDPs printed on products can be scanned by a device such as a phone camera. The original pattern and the digitally acquired one form a pair of meaningful representations of the same data. A sketch of the two representations is shown in the bottom row of Figure 5.
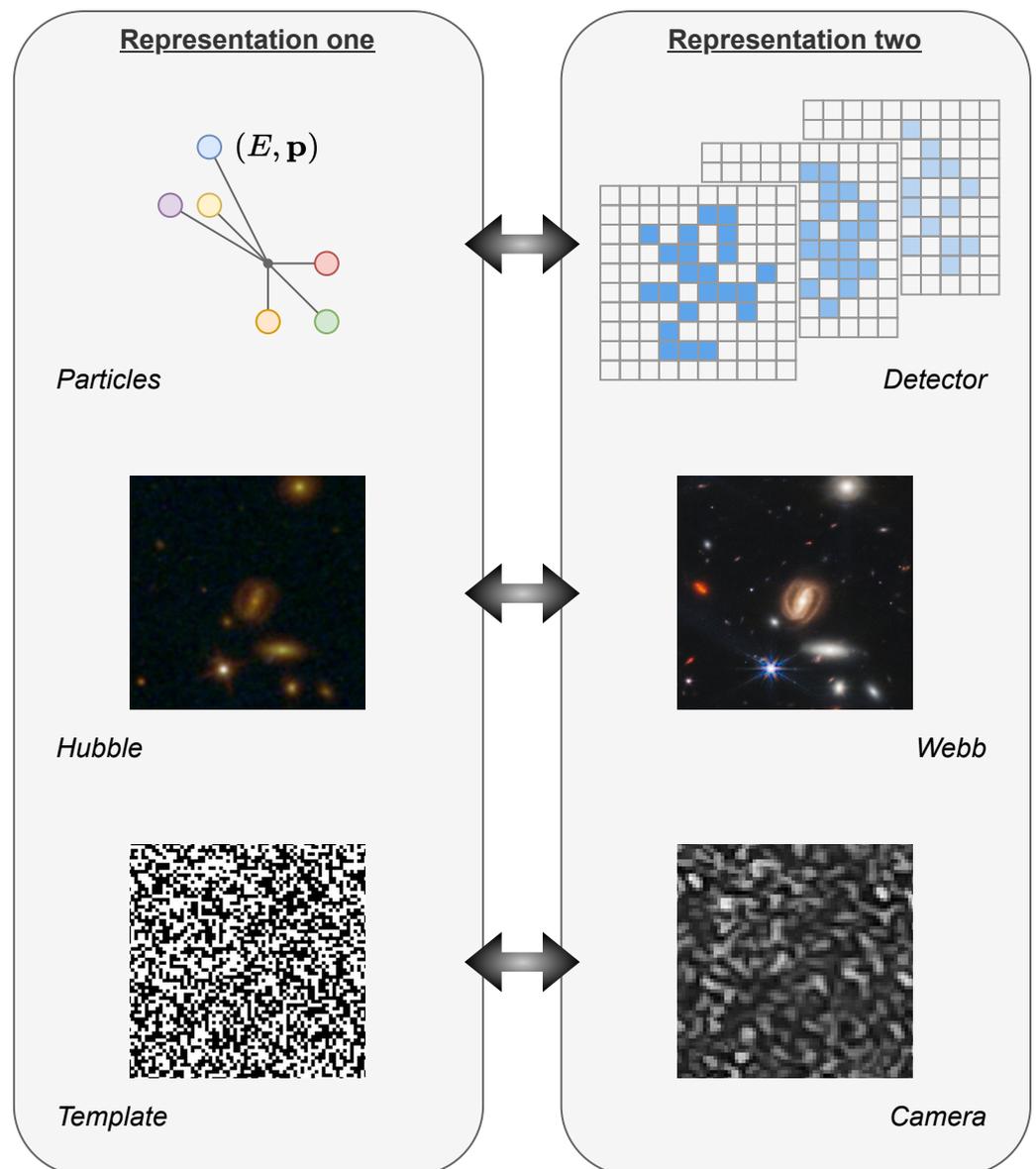
Figure 5. Three different applications that fit into the TURBO formalism. For each domain, two representations of the data are shown, each of which can be associated with one of the two spaces considered, given by the variables **X** and **Z**. The top row shows a high-energy physics example, where particles with given four-momenta are created in a collider experiment and detected by a detector. The middle row shows a galaxy imaging example, where two pictures of the same portion of the sky are taken by two different telescopes. The bottom row shows a counterfeiting detection example, where a digital template is acquired by a phone camera.

In summary, in many physical observation or measurement systems, we can consider the measured data as a latent space. The encoder therefore reflects the nature of a measurement or a sampling process. Further extraction of useful information can be considered as decoding the latent measured data and the overall system can be interpreted as a physical auto-encoder. In such a setting, the measured data are in general not Gaussian. Moreover, the data sensors usually being designed to provide as much information as possible about some events or phenomena leads to a maximisation of mutual information between the studied phenomena and their observations rather than a minimisation. There are many situations where the latent space has a physical meaning as relevant as the data space. In such domains, it may be highly relevant to keep as much information as possible, if not all

of it, in the latent space by maximising its mutual information with the data space. This is the most important concept leading to the TURBO formalism.

4. TURBO

In this section, we present the TURBO framework, which is formulated as a generalised auto-encoder framework. The main ingredient of TURBO is its general loss derived from the maximisation of various lower bounds to several mutual information expressions as opposed to the variational IBN counterpart. Additionally, instead of considering a single-way direction of information flow from $\mathbf{x}$ to $\tilde{\mathbf{z}}$ and back to $\hat{\mathbf{x}}$, TURBO considers a two-way uni-directional information flow that also includes the flow from $\mathbf{z}$ to $\tilde{\mathbf{x}}$ and back to $\hat{\mathbf{z}}$. It is important to note that TURBO interprets the random variables $\mathbf{X}$ and $\mathbf{Z}$ as following the joint distribution $p(\mathbf{x}, \mathbf{z})$ instead of treating them independently as in the BIB-AE formulation. Furthermore, once the general TURBO loss is developed, turning on and off the different terms allows us to recover many existing models, such as AAE [4], GAN [1], WGAN [31], pix2pix [15], SRGAN [32], CycleGAN [16] and even normalising flows [17]. With minor extensions, it also allows us to recover other models such as ALAE [18]. Maybe even more importantly, new models could also be uncovered by using new combinations of the TURBO loss terms. Therefore, the TURBO framework not only summarises existing systems that cannot be explained by the traditional IBN but also creates paths for the development of new ones.

4.1. General Objective Function

The starting point of TURBO is to express several forms of the mutual information between the data space and the latent space in the auto-encoder formulation. Three types of mutual information expressions, highlighted in Figure 2, are studied. We consider the mutual information for the real dataset $(\mathbf{x}, \mathbf{z}) \sim p(\mathbf{x}, \mathbf{z})$, which is denoted by $I(\mathbf{X}; \mathbf{Z})$. We also consider the two evolving mutual information expressions given by the encoded dataset $(\mathbf{x}, \tilde{\mathbf{z}}) \sim q_\phi(\mathbf{x}, \mathbf{z})$ and the decoded dataset $(\tilde{\mathbf{x}}, \mathbf{z}) \sim p_\theta(\mathbf{x}, \mathbf{z})$, denoted by $I_\phi(\mathbf{X}; \tilde{\mathbf{Z}})$ and $I_\theta(\tilde{\mathbf{X}}; \mathbf{Z})$, respectively. In the general case, the computation of mutual information in high-dimensional space for real data is infeasible. To make this problem tractable, we introduce four different lower bounds to these expressions. The technical details about the derivations can be found in Appendix C. The objective function of TURBO is based on these lower bounds and consists of their maximisation with respect to the parameters of the encoder and decoder networks.

For convenience, the objective function is translated into a loss minimisation problem. It involves eight terms, whose short notations are defined here for ease of reading:

$$\mathcal{L}_{\tilde{z}}(\mathbf{z}, \tilde{\mathbf{z}}) := -\mathbb{E}_{p(\mathbf{x}, \mathbf{z})}\big[\log q_\phi(\mathbf{z}|\mathbf{x})\big] \qquad \mathcal{D}_{\tilde{z}}(\mathbf{z}, \tilde{\mathbf{z}}) := D_{\mathrm{KL}}(p(\mathbf{z})\|\tilde{q}_\phi(\mathbf{z}))$$

$$\mathcal{L}_{\hat{x}}(\mathbf{x}, \hat{\mathbf{x}}) := -\mathbb{E}_{q_\phi(\mathbf{x}, \mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \qquad \mathcal{D}_{\hat{x}}(\mathbf{x}, \hat{\mathbf{x}}) := D_{\mathrm{KL}}(p(\mathbf{x})\|\hat{p}_\theta(\mathbf{x}))$$

$$\mathcal{L}_{\tilde{x}}(\mathbf{x}, \tilde{\mathbf{x}}) := -\mathbb{E}_{p(\mathbf{x}, \mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \qquad \mathcal{D}_{\tilde{x}}(\mathbf{x}, \tilde{\mathbf{x}}) := D_{\mathrm{KL}}(p(\mathbf{x})\|\tilde{p}_\theta(\mathbf{x}))$$

$$\mathcal{L}_{\hat{z}}(\mathbf{z}, \hat{\mathbf{z}}) := -\mathbb{E}_{p_\theta(\mathbf{x}, \mathbf{z})}\big[\log q_\phi(\mathbf{z}|\mathbf{x})\big] \qquad \mathcal{D}_{\hat{z}}(\mathbf{z}, \hat{\mathbf{z}}) := D_{\mathrm{KL}}(p(\mathbf{z})\|\hat{q}_\phi(\mathbf{z})).$$

The four terms in the left column correspond to conditional cross-entropies while the four terms in the right column represent KLDs between the true marginals and the marginals in the latent or reconstruction spaces. It should be noted that every KLD term is forward, meaning that the expected values are taken over the true data distribution. The losses involving such terms are therefore called *mean-seeking* as opposed to *mode-seeking*. While, in general, the choice of the order of the KLD follows empirical observations, the TURBO framework sets it beforehand. However, in practice, the KLD is usually approximated with expressions that often appear to lose this asymmetry property.

The conditional cross-entropy terms reflect the pair-wise relationships while the unpaired KLD terms characterise the correspondences between the distributions. Again, a common choice of the conditional distributions in the cross-entropies is to assume exponen-

tial deviations, leading to the ℓ_2-norm or the ℓ_1-norm in the special cases of multi-variate Gaussian or Laplacian, respectively. Therefore, the conditional cross-entropy and KLD terms can be computed in practice, providing the corresponding bound to the considered mutual information terms.

Each auto-encoder has constraints on the latent and reconstruction spaces, but the latent space of TURBO is not *fictitious* as in the IBN framework. It is rather linked to observable variables and that is why, instead of the IBN information minimisation in the latent space, TURBO considers information maximisation.

In contrast to the traditional single-way uni-directional IBN, the TURBO framework considers two flows of information named as *direct* and *reverse* paths. TURBO assumes that the real observable data follow the joint distribution $(\mathbf{x}, \mathbf{z}) \sim p(\mathbf{x}, \mathbf{z})$. Therefore, the two-way uni-directional nature of TURBO reflects the fact that this joint distribution can be decomposed in two different ways using the chain rule for probability distributions $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x})p(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$. Each path of TURBO corresponds to its own auto-encoder setting as shown in Figures 6 and 7. Each auto-encoder consists of two parametrised networks, $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$, which are shared between the *direct* and *reverse* paths. Only the order in which they are used is changed.
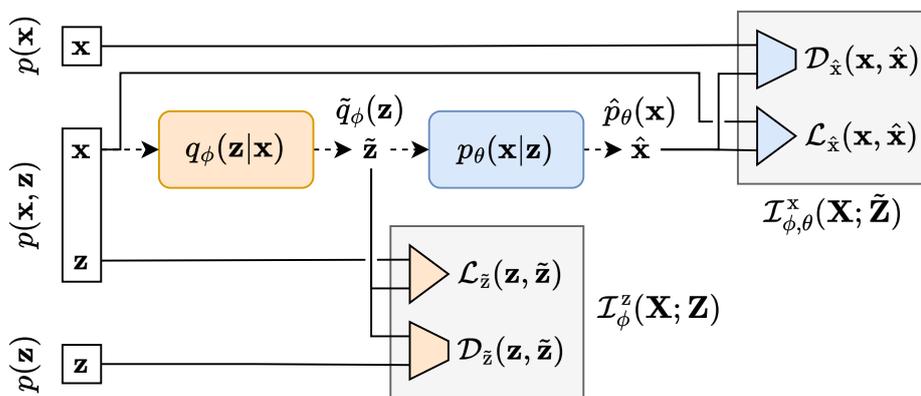
Figure 6. The *direct* path of the TURBO framework. Samples from the $\mathbf{X}$ space are encoded following the $q_\phi(\mathbf{z}|\mathbf{x})$ parametrised conditional distribution. A reconstruction loss term and a distribution matching loss term can be computed here. Then, the latent samples are decoded following the $p_\theta(\mathbf{x}|\mathbf{z})$ parametrised conditional distribution. Another pair of reconstruction and distribution matching loss terms can be computed at this step.
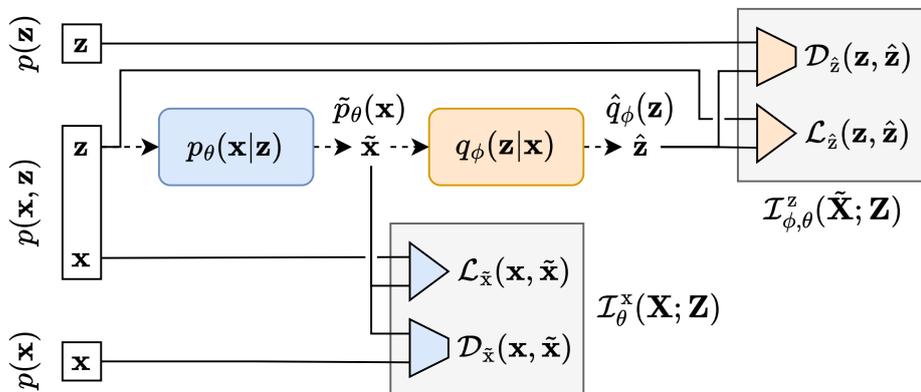
Figure 7. The *reverse* path of the TURBO framework. Samples from the $\mathbf{Z}$ space are decoded following the $p_\theta(\mathbf{x}|\mathbf{z})$ parametrised conditional distribution. A reconstruction loss term and a distribution matching loss term can be computed here. Then, the latent samples are decoded following the $q_\phi(\mathbf{z}|\mathbf{x})$ parametrised conditional distribution. Another pair of reconstruction and distribution matching loss terms can be computed at this step.

The *direct* path loss of TURBO corresponding to the encoding of the variable $\mathbf{x}$ into $\tilde{\mathbf{z}}$ and then the decoding of it back into $\hat{\mathbf{x}}$ as shown in Figure 6 is defined as

$$
\begin{aligned}
\mathcal{L}^{\text{direct}}(\phi, \theta) &= -\mathcal{I}_\phi^z(\mathbf{X}; \mathbf{Z}) - \lambda_D \mathcal{I}_{\phi,\theta}^x(\mathbf{X}; \tilde{\mathbf{Z}}) \\
&= \mathcal{L}_{\tilde{z}}(\mathbf{z}, \tilde{\mathbf{z}}) + \mathcal{D}_{\tilde{z}}(\mathbf{z}, \tilde{\mathbf{z}}) + \lambda_D \mathcal{L}_{\hat{x}}(\mathbf{x}, \hat{\mathbf{x}}) + \lambda_D \mathcal{D}_{\hat{x}}(\mathbf{x}, \hat{\mathbf{x}}),
\end{aligned}
\tag{11}
$$

where λ_D is a hyperparameter controlling the relative importance of the two mutual information bounds $\mathcal{I}_\phi^z(\mathbf{X}; \mathbf{Z})$ and $\mathcal{I}_{\phi,\theta}^x(\mathbf{X}; \tilde{\mathbf{Z}})$ derived in Appendixes C.1 and C.2, respectively. The formulation in Equation (11) is expressed in terms of a *loss* that is typically minimised in machine learning applications. That is why both terms have a minus sign in front of them. It should still be non-ambiguously considered as the maximisation of mutual information. The encoder part of the *direct* path loss ensures that the latent space variable $\tilde{\mathbf{z}}$ produced by the encoder $q_\phi(\mathbf{z}|\mathbf{x})$ matches its observable counterpart $\mathbf{z}$ for a given pair $(\mathbf{x}, \mathbf{z})$, according to the $\mathcal{L}_{\tilde{z}}(\mathbf{z}, \tilde{\mathbf{z}})$ term, while their marginals should be as close as possible according to the $\mathcal{D}_{\tilde{z}}(\mathbf{z}, \tilde{\mathbf{z}})$ term. The decoder part of the loss ensures the pair-wise correspondence between the reconstructed variable $\hat{\mathbf{x}}$ produced by the decoder $p_\theta(\mathbf{x}|\mathbf{z})$ from $\tilde{\mathbf{z}}$, according to the term $\mathcal{L}_{\hat{x}}(\mathbf{x}, \hat{\mathbf{x}})$, while the $\mathcal{D}_{\hat{x}}(\mathbf{x}, \hat{\mathbf{x}})$ term guarantees the matching of the reconstructed and true data distributions.

The *reverse* path loss of TURBO corresponding to the decoding of the variable $\mathbf{z}$ into $\tilde{\mathbf{x}}$ and then the encoding of it back into $\hat{\mathbf{z}}$ as shown in Figure 7 is defined as

$$
\begin{aligned}
\mathcal{L}^{\text{reverse}}(\phi, \theta) &= -\mathcal{I}_\theta^x(\mathbf{X}; \mathbf{Z}) - \lambda_R \mathcal{I}_{\phi,\theta}^z(\tilde{\mathbf{X}}; \mathbf{Z}) \\
&= \mathcal{L}_{\tilde{x}}(\mathbf{x}, \tilde{\mathbf{x}}) + \mathcal{D}_{\tilde{x}}(\mathbf{x}, \tilde{\mathbf{x}}) + \lambda_R \mathcal{L}_{\hat{z}}(\mathbf{z}, \hat{\mathbf{z}}) + \lambda_R \mathcal{D}_{\hat{z}}(\mathbf{z}, \hat{\mathbf{z}}),
\end{aligned}
\tag{12}
$$

where λ_R is another hyperparameter controlling the relative importance of the two mutual information bounds $\mathcal{I}_\theta^x(\mathbf{X}; \mathbf{Z})$ and $\mathcal{I}_{\phi,\theta}^z(\tilde{\mathbf{X}}; \mathbf{Z})$ derived in Appendixes C.3 and C.4, respectively. The interpretation of these four terms is analogous to the *direct* path.

The complete TURBO loss is finally defined as the weighted sum of the *direct* and *reverse* paths losses

$$
\mathcal{L}^{\text{TURBO}}(\phi, \theta) = \mathcal{L}^{\text{direct}}(\phi, \theta) + \lambda_T \mathcal{L}^{\text{reverse}}(\phi, \theta),
\tag{13}
$$

where a hyperparameter λ_T controls the relative importance of the two terms. It is worth noting that the full loss contains four bounds to mutual information expressions that involve both network sets of parameters ϕ and θ. In general, the optimal parameters that would maximise these four terms separately do not coincide. Therefore, the global solution of the complete optimisation problem usually shows deviations from the said optimal parameters, which is strongly dependent on the trade-off weights that balance the different bounds. Moreover, in practice, it is often impossible to calculate all the terms of the TURBO loss due to the nature of the data considered. For example, pairwise comparisons are particularly affected when there is no labelled correspondence between the $\mathbf{X}$ and $\mathbf{Z}$ variable domains. This results in multiple relevant architectures, whose objective functions lead to different optimal parameters.

4.2. Generalisation of Many Models

The complete loss being defined, we can now relate it to several well-known models. This means that the TURBO framework is a generalisation of these models that gives a uniform interpretation of their respective objective functions and creates a common basis towards explainable machine learning.

4.2.1. AAE

The inability of the BIB-AE framework to explain the celebrated family of AAEs [4] was among the many motivation factors in developing the new TURBO framework. Indeed, the AAE loss can now simply be expressed as the second and third terms of Equation (11):

$$\mathcal{L}^{\text{AAE}}(\phi, \theta) = \mathcal{D}_{\tilde{z}}(\mathbf{z}, \tilde{\mathbf{z}}) + \lambda_D \mathcal{L}_{\hat{\mathbf{x}}}(\mathbf{x}, \hat{\mathbf{x}}), \tag{14}$$

where we recognise the adversarial loss in the latent space $\mathbf{Z}$ and the reconstruction loss in the data space $\mathbf{X}$. The latent space distribution is controlled by the imposed prior $p(\mathbf{z})$. Figure 8 shows a schematic representation of the AAE architecture in the TURBO framework. Notice that the AAE, as a representative of the classical auto-encoder family, only uses the *direct* path.
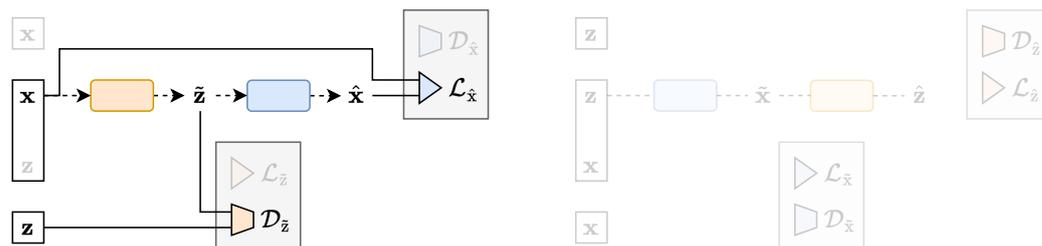
Figure 8. The AAE architecture expressed in the TURBO framework.

4.2.2. GAN and WGAN

As stated previously, GANs [1] are included into the BIB-AE formalism via a convoluted explanation of the distributions used to compute the adversarial loss. On the other hand, they can be easily expressed in the TURBO framework using only the second term of Equation (12):

$$\mathcal{L}^{\text{GAN}}(\theta) = \mathcal{D}_{\tilde{\mathbf{x}}}(\mathbf{x}, \tilde{\mathbf{x}}), \tag{15}$$

which much more naturally involves the data marginal distribution $p(\mathbf{x})$ and the approximated marginal distribution $\tilde{p}_\theta(\mathbf{x})$. In this formulation, the $\mathbf{Z}$ space is a placeholder used to represent any input to the decoder. It can be pure random noise, as for the StyleGAN model [33], or also include additional information such as class labels, as for the BigGAN [34] and StyleGAN-XL [35] large-scale models, which nowadays still compete with other modern frameworks [36,37]. As stated previously, the KLD term $\mathcal{D}_{\tilde{\mathbf{x}}}(\mathbf{x}, \tilde{\mathbf{x}})$ can be replaced by Wasserstein distance approximations, leading to the so-called Wasserstein GAN (WGAN) model [31]. Figure 9 shows a schematic representation of the GAN architecture in the TURBO framework. Notice that the classical GAN family does not make use of an encoder network and thus is better interpreted in the *reverse* path.
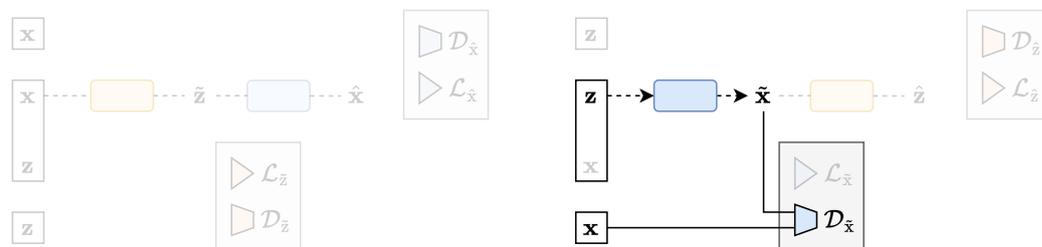
Figure 9. The GAN architecture expressed in the TURBO framework.

4.2.3. pix2pix and SRGAN

The pix2pix [15] and SRGAN [32] architectures are conditional GANs initially developed for image-to-image translation and image super-resolution, respectively. During training, both pix2pix and SRGAN assume the presence of N training pairs $\{\mathbf{x}_i, \mathbf{z}_i\}_{i=1}^N$, allowing one to use a paired loss for the translation network or the decoder optimisation.

However, since the typically used losses, such as ℓ_2-norm, do not cope with the statistical features of natural images, such a decoder produces poor results. This is why the training loss is additionally complemented by an adversarial term, the goal of which is to ensure that the translated images $\tilde{\mathbf{x}}$ are on the same manifold as the training data $\mathbf{x}$. Such an adversarial loss does not require paired data and is similar to the GAN family.

The considered paired systems can be expressed in the TURBO framework using the first and the second terms of Equation (12):

$$\mathcal{L}^{\text{pix2pix}}(\theta) = \mathcal{L}_{\tilde{\mathbf{x}}}(\mathbf{x}, \tilde{\mathbf{x}}) + \mathcal{D}_{\tilde{\mathbf{x}}}(\mathbf{x}, \tilde{\mathbf{x}}), \tag{16}$$

where the $\mathbf{Z}$ space now represents the image to be translated, while the additional random noise also used as input has to be implicitly understood. Figure 10 shows a schematic representation of the pix2pix and the SRGAN architectures in the TURBO framework. It is important to note that pix2pix and SRGAN do not consider the back reconstruction of $\hat{\mathbf{z}}$ from the generated data $\tilde{\mathbf{x}}$ as present in the full TURBO framework. Nevertheless, the fact that such systems have been proposed in the prior art and have produced state-of-the-art results for image-to-image translation and image super-resolution problems motivated us to consider them from the point of view of the TURBO generalisation.
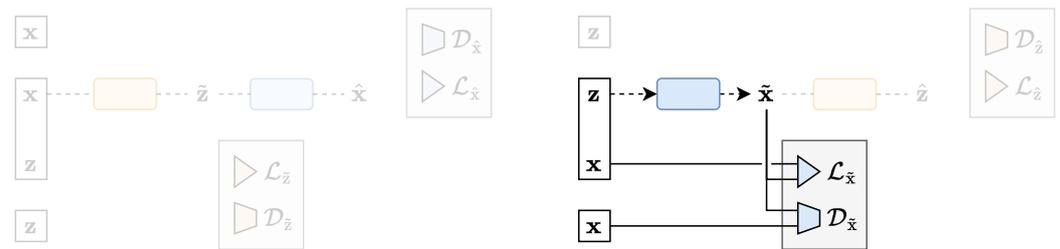
Figure 10. The pix2pix and SRGAN architectures expressed in the TURBO framework.

4.2.4. CycleGAN

The CycleGAN [16] architecture is an image-to-image translation model as well, but designed for unpaired data. It can be thought of as an AAE trained in two ways, where the $\mathbf{X}$ and $\mathbf{Z}$ spaces represent the two domains to be translated into each other. The CycleGAN loss can therefore be expressed in the TURBO framework using the second and the third terms of Equation (11) plus the second and the third terms of Equation (12):

$$\mathcal{L}^{\text{CycleGAN}}(\phi, \theta) = \mathcal{D}_{\tilde{\mathbf{z}}}(\mathbf{z}, \tilde{\mathbf{z}}) + \lambda_D \mathcal{L}_{\hat{\mathbf{x}}}(\mathbf{x}, \hat{\mathbf{x}}) + \lambda_T \mathcal{D}_{\tilde{\mathbf{x}}}(\mathbf{x}, \tilde{\mathbf{x}}) + \lambda_T \lambda_R \mathcal{L}_{\hat{\mathbf{z}}}(\mathbf{z}, \hat{\mathbf{z}}), \tag{17}$$

where $\lambda_T = 1$ and $\lambda_D = \lambda_R$ in the original loss formulation. Figure 11 shows a schematic representation of the CycleGAN architecture in the TURBO framework.
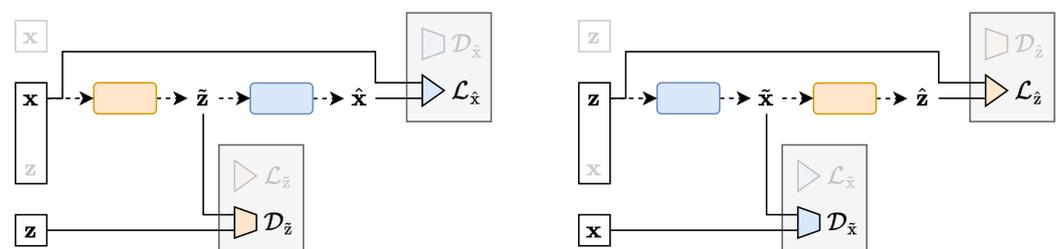
Figure 11. The CycleGAN architecture expressed in the TURBO framework.

4.2.5. Flows

In order to map a tractable base distribution to a complex data distribution, normalising flows [17] learn a series of invertible transformations, creating an expressive deterministic invertible function. The optimal function is usually found by maximising the likelihood of the data under the flow transformation. Sampling points from the base distribution and applying the transformation yields samples from the data distribution. Flows also allow one to evaluate the likelihood of a data sample by evaluating the likelihood of the corresponding base sample given by the inverse transformation.

At first look, normalising flows do not seem to fit into an auto-encoder framework. However, a flow can be thought of as a special case of an auto-encoder where the decoder is the deterministic parametrised invertible function, denoted by $T(\mathbf{z})$, while the encoder is its inverse $T^{-1}(\mathbf{x})$. Actually, the very principle of an auto-encoder is precisely to approximate this ideal case, usually denoting $\tilde{\mathbf{z}} = f_\phi(\mathbf{x}) = T^{-1}(\mathbf{x})$ for the encoder output and $\hat{\mathbf{x}} = g_\theta(\tilde{\mathbf{z}}) = T(\tilde{\mathbf{z}})$ for the decoder output. The approximated conditional distributions defined by Equations (1) and (2) thus read $p_\theta(\mathbf{x}|\mathbf{z}) = \delta(\mathbf{x} - T(\mathbf{z}))$ and $q_\phi(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - T^{-1}(\mathbf{x}))$, where $\delta(\cdot)$ is the Dirac distribution. Moreover, minimising the KLD between the true and approximated data marginal distributions is equivalent to maximising the likelihood of the data under the flow transformation [38]. The flow loss can therefore be expressed in the TURBO framework using only the second term of Equation (12):

$$\mathcal{L}^{\text{Flow}}(\theta) = \mathcal{D}_{\tilde{x}}(\mathbf{x}, \tilde{\mathbf{x}}), \tag{18}$$

which looks very much like the GAN loss of Equation (15). The differences reside in the way the KLD is computed or approximated, and in the parametrisations of the encoder and the decoder. The **Z** and **X** variables represent the base and data samples, respectively. Figure 12 shows a schematic representation of the flow architecture in the TURBO framework.
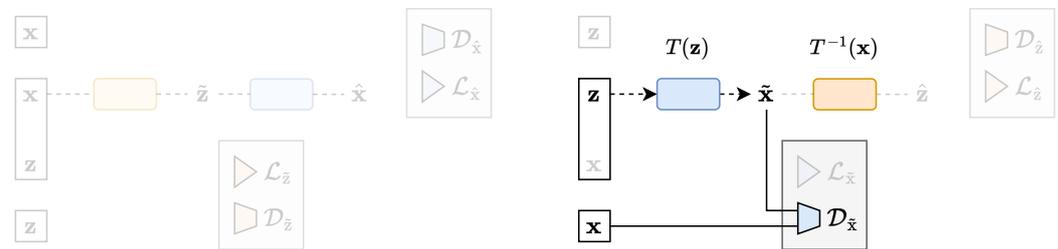
Figure 12. The flow architecture expressed in the TURBO framework.

4.3. Extension to Additional Models

In addition to the aforementioned models, other architectures can be expressed in the TURBO framework, provided some minor extensions are made. We give here an example of such an extension.

ALAE

The adversarial latent auto-encoder (ALAE) [18] is a model that tries to leverage the advantages of GANs, still using an auto-encoder architecture for better representation learning. The main novelty is to exclusively work in the latent space in order to disentangle this data representation as much as possible. The aim is to facilitate the manipulation of the latent space in downstream tasks, keeping a high quality of the generated data. The ALAE loss can be expressed in the TURBO framework using the third and a minor modification of the fourth terms of Equation (12):

$$\mathcal{L}^{\text{ALAE}}(\phi, \theta) = \mathcal{L}_{\hat{z}}(\mathbf{z}, \hat{\mathbf{z}}) + \bar{\mathcal{D}}_{\hat{z}}(\tilde{\mathbf{z}}, \hat{\mathbf{z}}). \tag{19}$$

This last term allows for cross-communication between the *direct* and the *reverse* paths. The derivation of this modified term, still based on mutual information maximisation, is detailed in Appendix D. Figure 13 shows a schematic representation of the ALAE architecture in the TURBO framework.
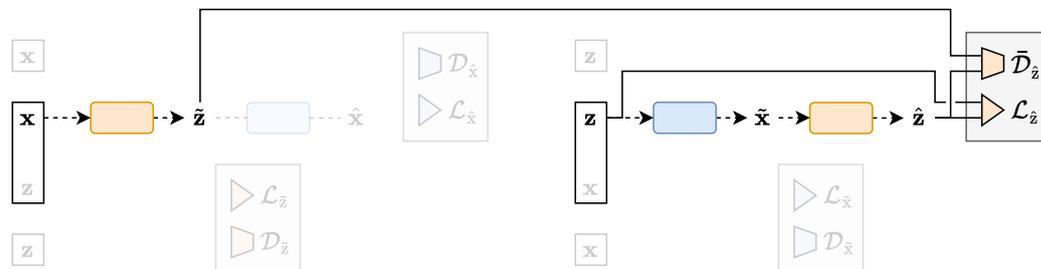
Figure 13. The ALAE architecture expressed in the TURBO framework.

5. Applications

In this section, we present several applications of the TURBO framework in studies about different domains. We highlight that these are summary presentations aiming to showcase how the method can be used and to demonstrate its potential. The complete studies as well as all the details are left to dedicated papers. These applications are somewhat disconnected, and TURBO is applied to diverse data with varying dimensionalities and statistics. Nevertheless, as reported in the corresponding studies, TURBO has additional benefits beyond interoperability, such as a superior performance with respect to the models compared, as well as more stable and more efficient training.

5.1. TURBO in High-Energy Physics: Turbo-Sim

The TURBO formalism has been successfully applied to a problem of particles into particles transformation in high-energy physics through the Turbo-Sim model [39]. The task is to transform the real four-momenta of a set of particles created by the collision of two protons in a collider experiment into the observed four-momenta of the particles captured by detectors, and vice versa. A clever interpretation of the problem is to think of the real and the observed spaces as two different representations of the same physical system, and to consider them as the **Z** and **X** spaces, respectively [30,40]. In such a case, the **Z** and **X** spaces are both physically meaningful and maximising the mutual information between them is highly relevant, making the TURBO formalism a natural choice for the problem.

The complete TURBO formalism as depicted in Figures 6 and 7 and formalised in Equation (13) is implemented in the Turbo-Sim model and compared to the OTUS model [30], the former being composed of two fully connected dense networks as the encoder and decoder. Moreover, we do not implement any of the physical constraints considered in the OTUS model. An example of the distributions generated by the TURBO model is shown in Figure 14 and a subset of the metrics used to evaluate the model is shown in Table 2. One can observe that the method is able to give good results up to uncertainties and even outperforms the OTUS method for several crucial observables. It is worth emphasising that the Turbo-Sim model uses very basic internal encoder and decoder networks, showcasing the strength of the TURBO formalism on its own.
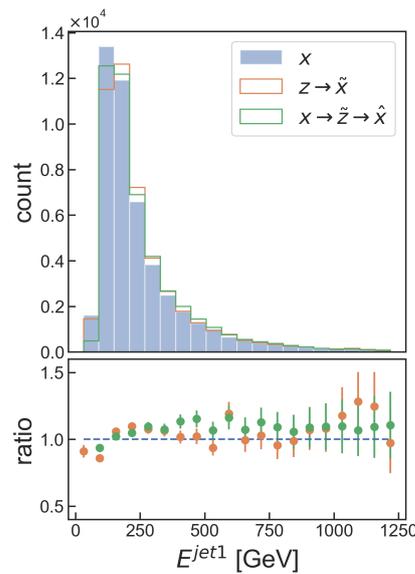
Figure 14. Selected example of distributions generated by the Turbo-Sim model. The histogram shows the distributions of the energy of a given observed particle, which, here, is a shower created by a chain of disintegration called *jet*, for the specific process of top-quark pair production. The blue bars correspond to the original data simulation, the orange line corresponds to the Turbo-Sim transformation from the real particle and the green line corresponds to the Turbo-Sim auto-encoded reconstruction.

Table 2. Selected subset of the metrics used to evaluate the Turbo-Sim model. The table shows the Kolmogorov–Smirnov distance [$\times 10^{-2}$] between the original data simulation and samples generated by the model. A lower value means a higher accuracy and **bold** highlights the best value per observable. One observable is shown per space. The energy of a real particle, a b-quark, is shown for the **Z** space, while the energy of the leading jet is shown for the **X** space. The Rec. column corresponds to unstable particles decaying into the real ones before flying through the detectors to be observed. The observables of these particles must be reconstructed from the combinations of the observed ones. Therefore, the quantity assesses whether the model has learnt the correlations between the variables well enough to make predictions about the underlying physics. In this specific process, two top-quarks are initially produced, and the observable is the invariant mass of the pair.

Model	Z space E^b	X space E^{jet1}	Rec. space m_{tt}
Turbo-Sim	3.96	**4.43**	**2.97**
OTUS	**2.76**	5.75	15.8

5.2. TURBO in Astronomy: Hubble-to-Webb

The advanced TURBO framework has been used in the domain of astronomy, specifically for sensor-to-sensor translation. The challenge involves using TURBO as an image-to-image translation framework to generate simulated images of the James Webb Space Telescope from observed images of the Hubble Space Telescope and vice versa. This application of TURBO, concisely called Hubble-to-Webb, is conducted on paired images of the galaxy cluster SMACS 0723.

In Figure 15, we showcase a side-by-side comparison of astronomical imagery. Several additional demos of Hubble images translated into Webb images by various models are available at https://hubble-to-webb.herokuapp.com/ (accessed on 20 September 23). The leftmost image is sourced from the Hubble Space Telescope, providing us with a crisp and detailed depiction of a celestial region. This Hubble image serves as the input to our TURBO image-to-image translation model. The middle image presents the target representation,

captured by the James Webb Space Telescope. The main objective of our model is to predict this high-fidelity Webb image using the Hubble input. On the right, we observe the image generated by the TURBO model, which displays a commendable attempt to replicate the intricate features of the actual Webb image. It is evident from the generated image that the TURBO model has made notable strides in bridging the differences between the two telescopes' observational capacities.
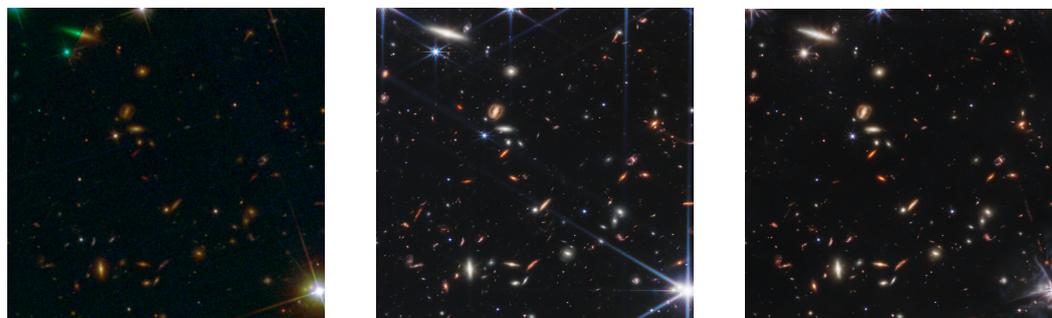
Figure 15. Comparison of images captured by the Hubble Space Telescope (**left**), the James Webb Space Telescope (**middle**) and generated by our TURBO image-to-image translation model (**right**).

A comparison is made between three methods, namely CycleGAN, pix2pix and TURBO. The same network architecture is used for all three methods; only the objective function is changed accordingly to reflect Equations (13), (16) and (17). The TURBO approach shows remarkable efficacy, demonstrating a state-of-the-art performance in terms of both the learned perceptual image patch similarity (LPIPS) and the Fréchet inception distance (FID) metrics within the domain of sensor-to-sensor translation, as compared to traditional image-to-image translation frameworks. The results of the Hubble-to-Webb translation are detailed in Table 3. Upon examination of the results table, it can be observed that TURBO supersedes other methods in terms of both the LPIPS and the FID metrics, while being competitive for the mean squared error (MSE), the structural similarity (SSIM) and the peak signal-to-noise ratio (PSNR) metrics. These metrics, although not directly related to per-pixel accuracy, are well-established indicators of image fidelity.

Table 3. Hubble-to-Webb sensor-to-sensor computed metrics. All results are obtained on a validation set of the Galaxy Cluster SMACS 0723. **Bold** highlights the best value per metric.

Model	MSE ↓	SSIM ↑	PSNR ↑	LPIPS ↓	FID ↓
CycleGAN	0.0097	0.83	20.11	0.48	128.1
pix2pix	**0.0021**	**0.93**	**26.78**	0.44	54.58
TURBO	0.0026	0.92	25.88	**0.41**	**43.36**

5.3. TURBO in Anti-Counterfeiting: Digital Twin

The TURBO framework has been employed to model the printing-imaging channel by leveraging a machine-learning-based *digital twin* for CDPs [41]. CDPs serve as a modern anti-counterfeiting technique in numerous applications. The process involves printing a highly detailed digital template **z** using an industrial high-resolution printer, resulting in a printed template **x**. The goal of the model is to accurately estimate the complex stochastic process of printing and to generate predictions x̃ of how a digital template would appear once printed, as well as to reverse the process and predict the original digital template z̃ from the printed one.

The same three methods, namely CycleGAN, pix2pix and TURBO, are also compared in this study. Network architectures are shared by the three methods and the objective functions of Equations (13), (16) and (17) constitute again the key differences between them. The study demonstrates that, regardless of various architectural factors, discriminators and hyperparameters, the TURBO framework consistently outperforms widely used image-to-

image translation models. A subset of the results is provided in Table 4. The TURBO model shows better results in almost all metrics, staying competitive in the other. In addition, a UMAP projection [42] of real and generated samples is shown in Figure 16. We can see that synthetic samples $\tilde{z}$ and $\tilde{x}$ are close to the corresponding real ones z and x, respectively. We can also observe two distinct clusters, one for digital and one for printed templates.
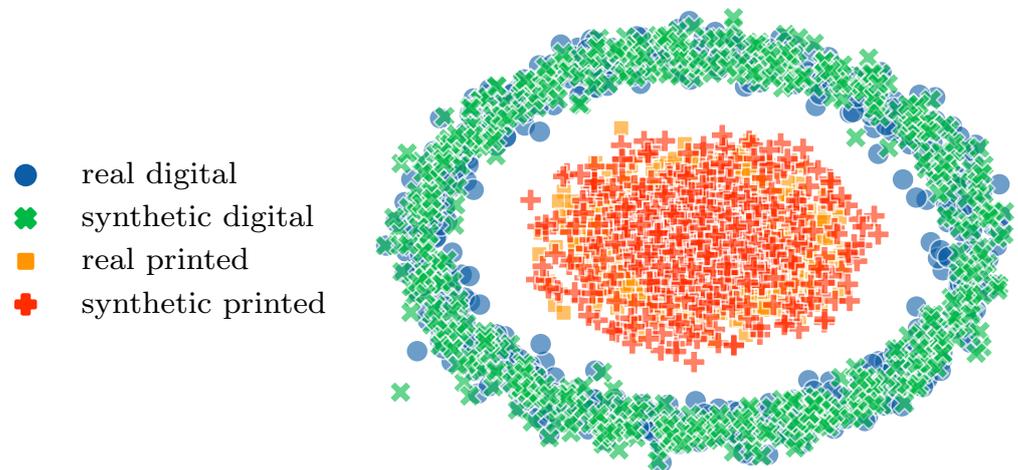
Figure 16. UMAP visualisation of synthetically generated digital and printed templates $\tilde{z}$ and $\tilde{x}$, respectively, superimposed on the corresponding real counterparts z and x.

Table 4. Digital twin estimation results. The performances of the models are evaluated on a test split of a dataset acquired by a scanner. The Hamming metric corresponds to the Hamming distance between the z and $\tilde{z}$ samples, while MSE and SSIM are computed between the x and $\tilde{x}$ samples. The FID metric is calculated in both directions. **Bold** highlights the best value per metric and *italic* is reserved to the case with direct comparison without any processing of the data.

Model	$\text{FID}_{x \rightarrow \tilde{z}} \downarrow$	$\text{FID}_{z \rightarrow \tilde{x}} \downarrow$	Hamming $\downarrow$	MSE $\downarrow$	SSIM $\uparrow$
W/O processing	*304*	*304*	*0.24*	*0.18*	*0.48*
CycleGAN	3.87	**4.45**	0.15	0.05	0.73
pix2pix	3.37	8.57	0.11	0.05	0.76
TURBO	**3.16**	6.60	**0.09**	**0.04**	**0.78**

In Figure 17, we show a visual comparison of template images. The image on the left is a randomly selected digital template, which acts as the input of our TURBO image-to-image translation model. The middle image is the corresponding printed template captured by a scanner. On the right, we display the image generated by the TURBO model. Visually, the synthetic sample looks almost indistinguishable from its real counterpart, meaning that the TURBO model is meritoriously capable of replicating the stochastic printing-capturing process.

In a recent extension of the work, it is shown that the TURBO framework outperforms the other methods not only on data acquired by a scanner but also on data captured using mobile phones. This demonstrates the robustness and versatility of the TURBO approach across different acquisition devices, highlighting its effectiveness in handling different scenarios and its superiority over traditional methods for both high-resolution scanner data and mobile phone data.
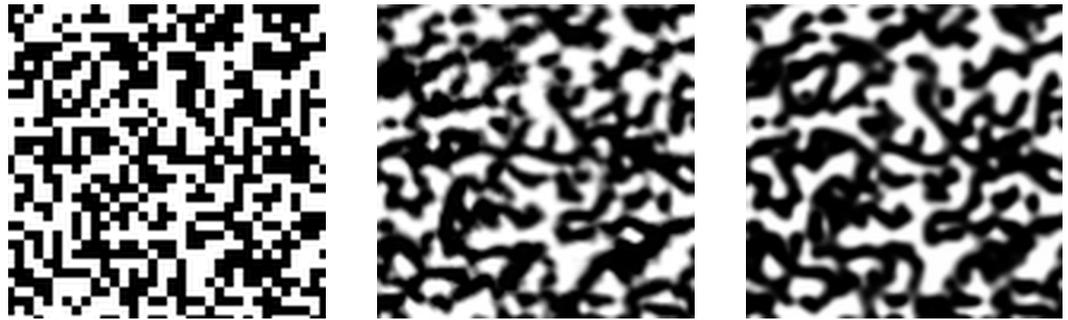
Figure 17. Comparison of a digital template (**left**), printed template (**middle**) and estimation generated by our TURBO image-to-image translation model (**right**). For better visualisation, we display a centrally cropped region that is equal to a quarter of the dimensions of the full image.

6. Conclusions

In this work, we have presented a new formalism, called TURBO, for the description of a class of auto-encoders that do not require a bottleneck structure. The foundation of this formalism is the maximisation of the mutual information between different representations of the data. We argue that this is a powerful paradigm that is worth considering in the design of many machine learning models in general. Indeed, we have shown that TURBO can be used to derive a number of existing models and that simple extensions also based on mutual information maximisation can lead to even more models. We have also highlighted several practical use cases where the TURBO formalism is either state-of-the-art or competitive with other models, demonstrating its versatility and robustness.

Our formulation of TURBO is based on the optimisation of multiple lower bounds to several mutual information terms, but it is important to note that other decompositions of such terms exist. We believe that many more modern machine learning architectures can be interpreted as maximising some form of mutual information. For example, SSL methods are not convincingly described by the IBN principle, and their understanding could certainly benefit from the new perspective provided by the TURBO formalism. Moreover, although general enough to allow for any stochastic neural network design, how to meaningfully bring stochasticity into the TURBO framework is left to future work. Another direct extension of the work presented in this paper would be to test TURBO in other relevant applications, namely any problem for which two modalities of the same underlying physical phenomenon are available. We leave the exploration of these ideas to future work and hope that our study will inspire further research in this direction as well, since having a common and interpretable general theory of deep learning is key to its comprehension.

Author Contributions: Conceptualisation, G.Q. and S.V.; formal analysis, G.Q., Y.B., V.K. and S.V.; investigation, G.Q., Y.B. and V.K.; software, G.Q., Y.B. and V.K.; supervision, S.V.; writing—original draft, G.Q.; writing—review and editing, G.Q., Y.B., V.K. and S.V. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: No data has been used nor created for this study. The information about the data relative to the applications in Section 5 is provided in the corresponding papers.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TURBO	Two-way Uni-directional Representations by Bounded Optimisation
IBN	Information Bottleneck
BIB-AE	Bounded Information Bottleneck Auto-Encoder
GAN	Generative Adversarial Network
WGAN	Wasserstein GAN
VAE	Variational Auto-Encoder
InfoVAE	Information maximising VAE
AAE	Adversarial Auto-Encoder
pix2pix	Image-to-Image Translation with Conditional GAN
SRGAN	Super-Resolution GAN
CycleGAN	Cycle-Consistent GAN
ALAE	Adversarial Latent Auto-Encoder
KLD	Kullback–Leibler Divergence
OTUS	Optimal-Transport-based Unfolding and Simulation
LPIPS	Learned Perceptual Image Patch Similarity
FID	Fréchet Inception Distance
MSE	Mean Squared Error
SSIM	Structural SIMilarity
PSNR	Peak Signal-to-Noise Ratio
CDP	Copy Detection Pattern
UMAP	Uniform Manifold Approximation and Projection

Appendix A. Notations Summary

Table A1. Summary of all symbols and naming used throughout the paper.

Notation	Description
$p(\mathbf{x}, \mathbf{z}), p(\mathbf{x}|\mathbf{z}), p(\mathbf{z}|\mathbf{x}), p(\mathbf{x}), p(\mathbf{z})$	Data joint, conditional and marginal distributions. Short notations for $p_{\mathsf{x},\mathsf{z}}(\mathbf{x}, \mathbf{z})$, $p_{\mathsf{x}}(\mathbf{x})$, etc.
$q_\phi(\mathbf{x}, \mathbf{z}), q_\phi(\mathbf{x}|\mathbf{z}), q_\phi(\mathbf{z}|\mathbf{x})$	Encoder joint and conditional distributions as defined in Equation (1).
$\tilde{q}_\phi(\mathbf{z}) := \int p(\mathbf{x})\, q_\phi(\mathbf{z}|\mathbf{x})\, \mathrm{d}\mathbf{x}$	Approximated marginal distribution of synthetic data in the encoder latent space.
$\hat{q}_\phi(\mathbf{z}) := \int \tilde{p}_\theta(\mathbf{x})\, q_\phi(\mathbf{z}|\mathbf{x})\, \mathrm{d}\mathbf{x}$	Approximated marginal distribution of synthetic data in the encoder reconstructed space.
$p_\theta(\mathbf{x}, \mathbf{z}), p_\theta(\mathbf{x}|\mathbf{z}), p_\theta(\mathbf{z}|\mathbf{x})$	Decoder joint and conditional distributions as defined in Equation (2).
$\tilde{p}_\theta(\mathbf{x}) := \int p(\mathbf{z})\, p_\theta(\mathbf{x}|\mathbf{z})\, \mathrm{d}\mathbf{z}$	Approximated marginal distribution of synthetic data in the decoder latent space.
$\hat{p}_\theta(\mathbf{x}) := \int \tilde{q}_\phi(\mathbf{z})\, p_\theta(\mathbf{x}|\mathbf{z})\, \mathrm{d}\mathbf{z}$	Approximated marginal distribution of synthetic data in the decoder reconstructed space.
$I(\mathbf{X}; \mathbf{Z}), I_\phi(\mathbf{X}; \tilde{\mathbf{Z}}), I_\theta(\tilde{\mathbf{X}}; \mathbf{Z})$	Mutual information as defined in Equation (3) and below. Subscripts mean that parametrised distributions are involved in the space denoted by a tilde.
$\mathcal{I}_\phi^{\mathsf{z}}(\mathbf{X}; \mathbf{Z}), \mathcal{I}_{\phi,\theta}^{\mathsf{x}}(\mathbf{X}; \tilde{\mathbf{Z}}),$ $\mathcal{I}_\theta^{\mathsf{x}}(\mathbf{X}; \mathbf{Z}), \mathcal{I}_{\phi,\theta}^{\mathsf{z}}(\tilde{\mathbf{X}}; \mathbf{Z})$	Lower bounds to mutual information as derived in Appendix C. Superscripts denote for which variable the corresponding loss terms are computed, subscripts denote the involved parametrised distributions and tildes follow the notations of the bounded mutual information.

Appendix B. BIB-AE Full Derivation

In this appendix, we detail the full derivation of all the BIB-AE loss terms expressed in our notations. It relies on a minimisation and maximisation trade-off for the mutual information between the data space and the latent space versus the latent space and the reconstructed data space, formally expressed in Equation (4).

Appendix B.1. Minimised Terms

Here, we detail the derivation of the loss computed between the random variables $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$. We start from the evolving mutual information between $\mathbf{X}$ and $\tilde{\mathbf{Z}}$:

$$I_\phi(\mathbf{X}; \tilde{\mathbf{Z}}) = \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{x},\mathbf{z})}{p(\mathbf{x})\tilde{q}_\phi(\mathbf{z})}\right] = \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{\tilde{q}_\phi(\mathbf{z})}\right], \tag{A1}$$

in which we inject the marginal distribution $p(\mathbf{z})$ and reorganise the expression as

$$\begin{aligned}
I_\phi(\mathbf{X}; \tilde{\mathbf{Z}}) &= \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{\tilde{q}_\phi(\mathbf{z})} \cdot \frac{p(\mathbf{z})}{p(\mathbf{z})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right] - \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{\tilde{q}_\phi(\mathbf{z})}{p(\mathbf{z})}\right] \\
&= \mathbb{E}_{p(\mathbf{x})}\left[D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))\right] - D_{\mathrm{KL}}(\tilde{q}_\phi(\mathbf{z})\|p(\mathbf{z})),
\end{aligned} \tag{A2}$$

where we use the two decompositions of Equation (1) to get to the two KLD formulas. This gives the two terms that are minimised in the BIB-AE original formulation.

Appendix B.2. Maximised Terms

The derivation of the BIB-AE loss computed between the random variables $\mathbf{X}$ and $\hat{\mathbf{X}}$ follows the exact same steps as detailed in Appendix C.2 for the TURBO loss. For completeness, we quote the final result here and refer the reader to Appendix C.2 for the details. It uses the lower bound to the evolving mutual information

$$I_\phi(\mathbf{X}; \tilde{\mathbf{Z}}) \geq \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(p(\mathbf{x})\|\hat{p}_\theta(\mathbf{x})) =: \mathcal{I}_{\phi,\theta}^{\mathrm{x}}(\mathbf{X}; \tilde{\mathbf{Z}}), \tag{A3}$$

which gives the two terms that are maximised in the BIB-AE original formulation. The first term corresponds to the paired data reconstruction consistency constraint while the second term requires the unpaired or distribution-wise consistency between the training data and the reconstructed data.

Appendix C. TURBO Full Derivation

In this appendix, we detail the full derivation of all the TURBO loss terms. Each pair of terms is derived starting from the mutual information between two given random variables and injecting the proper distribution approximations. We present both the *direct* and the *reverse* paths that auto-encode $\mathbf{X}$ and $\mathbf{Z}$, respectively. For the two cases, a pair of terms is derived in the encoder space and another pair is derived in the decoder space.

Appendix C.1. Direct Path, Encoder Space

Here, we detail the derivation of the loss computed between the random variables $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$. We start from the mutual information between $\mathbf{X}$ and $\mathbf{Z}$:

$$I(\mathbf{X}; \mathbf{Z}) = \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}\right] = \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right], \tag{A4}$$

in which we inject the parametrised approximated conditional distribution $q_\phi(\mathbf{z}|\mathbf{x})$ before reorganising the expression as

$$\begin{aligned}
I(\mathbf{X}; \mathbf{Z}) &= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} \cdot \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right] + \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right] + \mathbb{E}_{p(\mathbf{x})}\left[D_{\mathrm{KL}}(p(\mathbf{z}|\mathbf{x})\|q_\phi(\mathbf{z}|\mathbf{x}))\right].
\end{aligned} \tag{A5}$$

Since both the KLD and any probability distribution are non-negative quantities, the last term in Equation (A5) is non-negative and the mutual information is lower bounded by

$$I(\mathbf{X};\mathbf{Z}) \geq \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right]. \tag{A6}$$

We now inject the approximated marginal distribution $\tilde{q}_\phi(\mathbf{z})$ and reorganise the expression further as

$$\begin{aligned}
I(\mathbf{X};\mathbf{Z}) &\geq \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\cdot\frac{\tilde{q}_\phi(\mathbf{z})}{\tilde{q}_\phi(\mathbf{z})}\right]\\
&= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right] - \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log\frac{p(\mathbf{z})}{\tilde{q}_\phi(\mathbf{z})}\right] - \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log\tilde{q}_\phi(\mathbf{z})\right] \qquad (A7)\\
&= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right] - D_{\mathrm{KL}}(p(\mathbf{z})\|\tilde{q}_\phi(\mathbf{z})) + H(p(\mathbf{z});\tilde{q}_\phi(\mathbf{z})).
\end{aligned}$$

Since the cross-entropy is a non-negative quantity, the last term in Equation (A7) is non-negative. Notice that in general, this statement is only true for discrete random variables. For continuous random variables, certain conditions should be satisfied. For example, the entropy of a Gaussian random variable X with variance σ^2 reads $h(X) = 1/2\log(2\pi e\sigma^2)$ and is non-negative if $\sigma^2 \geq 1/2\pi e$. This condition is relatively easy to fulfil in practice. The mutual information can therefore be lower bounded by

$$I(\mathbf{X};\mathbf{Z}) \geq \underbrace{\mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right]}_{=:\,-\mathcal{L}_{\tilde{z}}(\mathbf{z},\tilde{\mathbf{z}})} - \underbrace{D_{\mathrm{KL}}(p(\mathbf{z})\|\tilde{q}_\phi(\mathbf{z}))}_{=:\,\mathcal{D}_{\tilde{z}}(\mathbf{z},\tilde{\mathbf{z}})} =: \mathcal{I}_\phi^z(\mathbf{X};\mathbf{Z}), \tag{A8}$$

where we define $\mathcal{I}_\phi^z(\mathbf{X};\mathbf{Z})$ as a parametrised lower bound to $I(\mathbf{X};\mathbf{Z})$ and where we define the two loss terms $\mathcal{L}_{\tilde{z}}(\mathbf{z},\tilde{\mathbf{z}})$ and $\mathcal{D}_{\tilde{z}}(\mathbf{z},\tilde{\mathbf{z}})$.

A crucial property of this lower bound is that its unique maximum with respect to ϕ, corresponding to the optimal parameter $\phi^* = \arg\max_\phi \mathcal{I}_\phi^z(\mathbf{X};\mathbf{Z})$, is reached when $q_{\phi^*}(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x})$, i.e., when the encoder perfectly reflects the real physical process of data transformation from the $\mathbf{X}$ manifold to the $\mathbf{Z}$ manifold.

Indeed, notice that, for a given dataset with $(\mathbf{x},\mathbf{z}) \sim p(\mathbf{x},\mathbf{z})$, the mutual information $I(\mathbf{X};\mathbf{Z})$ is constant. When we insert the parametrised distribution in Equation (A5), the final expression seems to depend on ϕ, but the terms actually compensate each other and the value eventually does not depend on ϕ. In addition, the KLD term in Equation (A5) vanishes if and only if the two compared distributions $q_\phi(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$ are identical. Therefore, the equality in Equation (A6) holds if and only if $q_{\phi^*}(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x})$ and it is the unique maximum with respect to ϕ of the right-hand side expression.

Further, notice that, in this case, $\tilde{q}_{\phi^*}(\mathbf{z}) = p(\mathbf{z})$, so one would have $H(p(\mathbf{z});\tilde{q}_{\phi^*}(\mathbf{z})) = H(p(\mathbf{z}))$ in Equation (A7). Since, for any two dissimilar distributions $\tilde{q}_\phi(\mathbf{z}) \neq p(\mathbf{z})$, one has $H(p(\mathbf{z});\tilde{q}_\phi(\mathbf{z})) > H(p(\mathbf{z}))$, the solution $q_{\phi^*}(\mathbf{z}|\mathbf{x})$ also coincides with the unique maximum of $\mathcal{I}_\phi^z(\mathbf{X};\mathbf{Z})$ with respect to ϕ. Indeed, if a higher maximum was reached for another distribution $q_{\bar{\phi}}(\mathbf{z}|\mathbf{x})$, it should be designed so that $H(p(\mathbf{z});\tilde{q}_{\bar{\phi}}(\mathbf{z})) < H(p(\mathbf{z}))$, which is impossible, in order to keep satisfying the inequalities in Equation (A6) and Equation (A7). Moreover, if multiple maxima would exist and would satisfy $H(p(\mathbf{z});\tilde{q}_{\phi^*}(\mathbf{z})) = H(p(\mathbf{z}))$, the uniqueness of the saturating point of Equation (A6) would be broken.

Appendix C.2. Direct Path, Decoder Space

Here, we detail the derivation of the loss computed between the random variables $\mathbf{X}$ and $\hat{\mathbf{X}}$. We start from the evolving mutual information between $\mathbf{X}$ and $\tilde{\mathbf{Z}}$:

$$I_\phi(\mathbf{X};\tilde{\mathbf{Z}}) = \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log\frac{q_\phi(\mathbf{x},\mathbf{z})}{p(\mathbf{x})\tilde{q}_\phi(\mathbf{z})}\right] = \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log\frac{q_\phi(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}\right], \tag{A9}$$

in which we inject the parametrised approximated conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$ before reorganising the expression as

$$
\begin{aligned}
I_\phi(\mathbf{X};\tilde{\mathbf{Z}}) &= \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})} \cdot \frac{p_\theta(\mathbf{x}|\mathbf{z})}{p_\theta(\mathbf{x}|\mathbf{z})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}\right] + \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{x}|\mathbf{z})}{p_\theta(\mathbf{x}|\mathbf{z})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}\right] + \mathbb{E}_{\tilde{q}_\phi(\mathbf{z})}\left[D_{\mathrm{KL}}(q_\phi(\mathbf{x}|\mathbf{z})\|p_\theta(\mathbf{x}|\mathbf{z}))\right].
\end{aligned}
\tag{A10}
$$

The last term in Equation (A10) is non-negative and the mutual information is lower bounded by

$$
I_\phi(\mathbf{X};\tilde{\mathbf{Z}}) \geq \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}\right].
\tag{A11}
$$

We now inject the approximated marginal distribution $\hat{p}_\theta(\mathbf{x})$ and reorganise the expression further as

$$
\begin{aligned}
I_\phi(\mathbf{X};\tilde{\mathbf{Z}}) &\geq \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})} \cdot \frac{\hat{p}_\theta(\mathbf{x})}{\hat{p}_\theta(\mathbf{x})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x})}{\hat{p}_\theta(\mathbf{x})}\right] - \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log \hat{p}_\theta(\mathbf{x})] \\
&= \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(p(\mathbf{x})\|\hat{p}_\theta(\mathbf{x})) + H(p(\mathbf{x});\hat{p}_\theta(\mathbf{x})).
\end{aligned}
\tag{A12}
$$

The last term in Equation (A12) is non-negative and the mutual information is lower bounded by

$$
I_\phi(\mathbf{X};\tilde{\mathbf{Z}}) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{=: -\mathcal{L}_{\hat{\mathbf{x}}}(\mathbf{x},\hat{\mathbf{x}})} - \underbrace{D_{\mathrm{KL}}(p(\mathbf{x})\|\hat{p}_\theta(\mathbf{x}))}_{=: \mathcal{D}_{\hat{\mathbf{x}}}(\mathbf{x},\hat{\mathbf{x}})} =: \mathcal{I}_{\phi,\theta}^{\mathrm{x}}(\mathbf{X};\tilde{\mathbf{Z}}),
\tag{A13}
$$

where we define $\mathcal{I}_{\phi,\theta}^{\mathrm{x}}(\mathbf{X};\tilde{\mathbf{Z}})$ as a parametrised lower bound to $I_\phi(\mathbf{X};\tilde{\mathbf{Z}})$ and where we define the two loss terms $\mathcal{L}_{\hat{\mathbf{x}}}(\mathbf{x},\hat{\mathbf{x}})$ and $\mathcal{D}_{\hat{\mathbf{x}}}(\mathbf{x},\hat{\mathbf{x}})$.

With a proof analogous to the one given in Appendix C.1, we find that the unique maximum of this lower bound, corresponding to the optimal parameter $\theta^* = \arg\max_\theta \mathcal{I}_{\phi,\theta}^{\mathrm{x}}(\mathbf{X};\tilde{\mathbf{Z}})$, is reached when $p_{\theta^*}(\mathbf{x}|\mathbf{z}) = q_\phi(\mathbf{x}|\mathbf{z})$, i.e., when the decoder is a perfect distribution-wise inverse of the encoder. Notice that, in this case, $\hat{p}_{\theta^*}(\mathbf{x}) = \int \tilde{q}_\phi(\mathbf{z}) p_{\theta^*}(\mathbf{x}|\mathbf{z})\,\mathrm{d}\mathbf{z} = \int \tilde{q}_\phi(\mathbf{z}) q_\phi(\mathbf{x}|\mathbf{z})\,\mathrm{d}\mathbf{z} = \int p(\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x})\,\mathrm{d}\mathbf{z} = p(\mathbf{x})$ as necessary for the proof to hold.

Furthermore, once optimised with respect to θ, the lower bound satisfies $\mathcal{I}_{\phi,\theta^*}^{\mathrm{x}}(\mathbf{X};\tilde{\mathbf{Z}}) = I_\phi(\mathbf{X};\tilde{\mathbf{Z}}) - H(p(\mathbf{x}))$. Therefore, maximising $\mathcal{I}_{\phi,\theta^*}^{\mathrm{x}}(\mathbf{X};\tilde{\mathbf{Z}})$ with respect to ϕ is equivalent to maximising the evolving mutual information $I_\phi(\mathbf{X};\tilde{\mathbf{Z}})$ between $\mathbf{X}$ and $\tilde{\mathbf{Z}}$.

It is worth noting that maximising $I_\phi(\mathbf{X};\tilde{\mathbf{Z}})$ with respect to ϕ does not give any guarantee that $q_\phi(\mathbf{x},\mathbf{z})$ would match $p(\mathbf{x},\mathbf{z})$, meaning that, in general, one has $\phi^* = \arg\max_\phi \mathcal{I}_\phi^{\mathrm{z}}(\mathbf{X};\mathbf{Z}) \neq \arg\max_\phi \mathcal{I}_{\phi,\theta^*}^{\mathrm{x}}(\mathbf{X};\tilde{\mathbf{Z}})$. Only the combination of the two objectives is meaningful in this sense.

Appendix C.3. Reverse Path, Decoder Space

Here, we detail the derivation of the loss computed between the random variables $\mathbf{X}$ and $\tilde{\mathbf{X}}$. We start from the mutual information between $\mathbf{X}$ and $\mathbf{Z}$:

$$
I(\mathbf{X};\mathbf{Z}) = \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}\right] = \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}\right],
\tag{A14}
$$

in which we inject the parametrised approximated conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$ before reorganising the expression as

$$
\begin{aligned}
I(\mathbf{X};\mathbf{Z}) &= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})} \cdot \frac{p_\theta(\mathbf{x}|\mathbf{z})}{p_\theta(\mathbf{x}|\mathbf{z})}\right] \\
&= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}\right] + \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x}|\mathbf{z})}{p_\theta(\mathbf{x}|\mathbf{z})}\right] \\
&= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}\right] + \mathbb{E}_{p(\mathbf{z})}[D_{\text{KL}}(p(\mathbf{x}|\mathbf{z})\|p_\theta(\mathbf{x}|\mathbf{z}))],
\end{aligned}
\tag{A15}
$$

The last term in Equation (A15) is non-negative and the mutual information is lower bounded by

$$
I(\mathbf{X};\mathbf{Z}) \geq \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}\right].
\tag{A16}
$$

We now inject the approximated marginal distribution $\tilde{p}_\theta(\mathbf{x})$ and reorganise the expression further as

$$
\begin{aligned}
I(\mathbf{X};\mathbf{Z}) &\geq \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})} \cdot \frac{\tilde{p}_\theta(\mathbf{x})}{\tilde{p}_\theta(\mathbf{x})}\right] \\
&= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x})}{\tilde{p}_\theta(\mathbf{x})}\right] - \mathbb{E}_{p(\mathbf{x},\mathbf{z})}[\log \tilde{p}_\theta(\mathbf{x})] \\
&= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(p(\mathbf{x})\|\tilde{p}_\theta(\mathbf{x})) + H(p(\mathbf{x});\tilde{p}_\theta(\mathbf{x})),
\end{aligned}
\tag{A17}
$$

The last term in Equation (A17) is non-negative and the mutual information is lower bounded by

$$
I(\mathbf{X};\mathbf{Z}) \geq \underbrace{\mathbb{E}_{p(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{=:\,-\mathcal{L}_{\tilde{x}}(\mathbf{x},\tilde{\mathbf{x}})} - \underbrace{D_{\text{KL}}(p(\mathbf{x})\|\tilde{p}_\theta(\mathbf{x}))}_{=:\,\mathcal{D}_{\tilde{x}}(\mathbf{x},\tilde{\mathbf{x}})} =:\, \mathcal{I}_\theta^x(\mathbf{X};\mathbf{Z}),
\tag{A18}
$$

where we define $\mathcal{I}_\theta^x(\mathbf{X};\mathbf{Z})$ as a parametrised lower bound to $I(\mathbf{X};\mathbf{Z})$ and where we define the two loss terms $\mathcal{L}_{\tilde{x}}(\mathbf{x},\tilde{\mathbf{x}})$ and $\mathcal{D}_{\tilde{x}}(\mathbf{x},\tilde{\mathbf{x}})$.

A symmetrical proof to the one given in Appendix C.1 shows that the optimal parameter $\theta^\dagger = \arg\max_\theta \mathcal{I}_\theta^x(\mathbf{X};\mathbf{Z})$ corresponds to $p_{\theta^\dagger}(\mathbf{x}|\mathbf{z}) = p(\mathbf{x}|\mathbf{z})$, i.e., when the decoder perfectly reflects the real physical process of data transformation from the $\mathbf{Z}$ manifold to the $\mathbf{X}$ manifold.

Appendix C.4. Reverse Path, Encoder Space

Here, we detail the derivation of the loss computed between the random variables $\mathbf{Z}$ and $\hat{\mathbf{Z}}$. We start from the evolving mutual information between $\tilde{\mathbf{X}}$ and $\mathbf{Z}$:

$$
I_\theta(\tilde{\mathbf{X}};\mathbf{Z}) = \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{\tilde{p}_\theta(\mathbf{x})p(\mathbf{z})}\right] = \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right],
\tag{A19}
$$

in which we inject the parametrised approximated conditional distribution $q_\phi(\mathbf{z}|\mathbf{x})$ before reorganising the expression as

$$
\begin{aligned}
I_\theta(\tilde{\mathbf{X}};\mathbf{Z}) &= \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} \cdot \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right] + \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right] + \mathbb{E}_{\tilde{p}_\theta(\mathbf{x})}[D_{\text{KL}}(p_\theta(\mathbf{z}|\mathbf{x})\|q_\phi(\mathbf{z}|\mathbf{x}))].
\end{aligned}
\tag{A20}
$$

The last term in Equation (A20) is non-negative and the mutual information is lower bounded by

$$I_\theta(\tilde{\mathbf{X}}; \mathbf{Z}) \geq \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right]. \tag{A21}$$

We now inject the approximated marginal distribution $\hat{q}_\phi(\mathbf{z})$ and reorganise the expression further as

$$
\begin{aligned}
I_\theta(\tilde{\mathbf{X}}; \mathbf{Z}) &\geq \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} \cdot \frac{\hat{q}_\phi(\mathbf{z})}{\hat{q}_\phi(\mathbf{z})}\right] \\
&= \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right] - \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{z})}{\hat{q}_\phi(\mathbf{z})}\right] - \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log \hat{q}_\phi(\mathbf{z})\right] \\
&= \mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right] - D_{\mathrm{KL}}(p(\mathbf{z})\|\hat{q}_\phi(\mathbf{z})) + H(p(\mathbf{z}); \hat{q}_\phi(\mathbf{z})).
\end{aligned}
\tag{A22}
$$

The last term in Equation (A22) is non-negative and the mutual information is lower bounded by

$$I_\theta(\tilde{\mathbf{X}}; \mathbf{Z}) \geq \underbrace{\mathbb{E}_{p_\theta(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right]}_{=: -\mathcal{L}_{\hat{z}}(\mathbf{z},\hat{\mathbf{z}})} - \underbrace{D_{\mathrm{KL}}(p(\mathbf{z})\|\hat{q}_\phi(\mathbf{z}))}_{=: \mathcal{D}_{\hat{z}}(\mathbf{z},\hat{\mathbf{z}})} =: \mathcal{I}^z_{\phi,\theta}(\tilde{\mathbf{X}}; \mathbf{Z}), \tag{A23}$$

where we define $\mathcal{I}^z_{\phi,\theta}(\tilde{\mathbf{X}}; \mathbf{Z})$ as a parametrised lower bound to $I_\theta(\tilde{\mathbf{X}}; \mathbf{Z})$ and where we define the two loss terms $\mathcal{L}_{\hat{z}}(\mathbf{z},\hat{\mathbf{z}})$ and $\mathcal{D}_{\hat{z}}(\mathbf{z},\hat{\mathbf{z}})$.

A symmetrical proof to the one given in Appendix C.2 shows that the optimal parameter $\phi^\dagger = \arg\max_\phi \mathcal{I}^z_{\phi,\theta}(\tilde{\mathbf{X}}; \mathbf{Z})$ corresponds to $q_{\phi^\dagger}(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x})$, i.e., when the encoder is a perfect distribution-wise inverse of the decoder, and that maximising $\mathcal{I}^z_{\phi^\dagger,\theta}(\tilde{\mathbf{X}}; \mathbf{Z})$ with respect to θ is equivalent to maximising $I_\theta(\tilde{\mathbf{X}}; \mathbf{Z})$.

Again, in general, $\theta^\dagger = \arg\max_\theta \mathcal{I}^x_\theta(\mathbf{X}; \mathbf{Z}) \neq \arg\max_\theta \mathcal{I}^z_{\phi^\dagger,\theta}(\tilde{\mathbf{X}}; \mathbf{Z})$ and only the combined objective makes full sense.

Appendix D. ALAE Modified Term

In this appendix, we detail how to obtain the modified adversarial term of the ALAE loss found in Equation (19). It is based on the same mutual information term as defined in Appendix C.2, but following another decomposition:

$$I_\phi(\mathbf{X}; \tilde{\mathbf{Z}}) = \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{x},\mathbf{z})}{p(\mathbf{x})\tilde{q}_\phi(\mathbf{z})}\right] = \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{\tilde{q}_\phi(\mathbf{z})}\right], \tag{A24}$$

in which we inject the parametrised approximated marginal distribution $\hat{q}_\phi(\mathbf{z})$ before reorganising the expression as

$$
\begin{aligned}
I_\phi(\mathbf{X}; \tilde{\mathbf{Z}}) &= \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{\tilde{q}_\phi(\mathbf{z})} \cdot \frac{\hat{q}_\phi(\mathbf{z})}{\hat{q}_\phi(\mathbf{z})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right] - \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{\tilde{q}_\phi(\mathbf{z})}{\hat{q}_\phi(\mathbf{z})}\right] - \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \hat{q}_\phi(\mathbf{z})\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right] - D_{\mathrm{KL}}(\tilde{q}_\phi(\mathbf{z})\|\hat{q}_\phi(\mathbf{z})) + H(\tilde{q}_\phi(\mathbf{z}); \hat{q}_\phi(\mathbf{z})).
\end{aligned}
\tag{A25}
$$

The last term in Equation (A25) is non-negative and the mutual information is lower bounded by

$$I_\phi(\mathbf{X}; \tilde{\mathbf{Z}}) \geq \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right] - D_{\mathrm{KL}}(\tilde{q}_\phi(\mathbf{z})\|\hat{q}_\phi(\mathbf{z})), \tag{A26}$$

where the KLD term is the modified adversarial term $\bar{\mathcal{D}}_{\hat{z}}(\tilde{\mathbf{z}}, \hat{\mathbf{z}})$ of the ALAE loss.

References

1. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. *arXiv* **2014**, arXiv:1406.2661.
2. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2013**, arXiv:1312.6114.
3. Rezende, D.J.; Mohamed, S.; Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In Proceedings of the International Conference on Machine Learning, PMLR, Beijing, China, 21–26 June 2014; pp. 1278–1286.
4. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial Autoencoders. *arXiv* **2015**, arXiv:1511.05644.
5. Tishby, N.; Zaslavsky, N. Deep learning and the information bottleneck principle. In Proceedings of the IEEE Information Theory Workshop, Jerusalem, Israel, 26 April–1 May 2015; pp. 1–5.
6. Alemi, A.A.; Fischer, I.; Dillon, J.V.; Murphy, K. Deep Variational Information Bottleneck. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
7. Voloshynovskiy, S.; Taran, O.; Kondah, M.; Holotyak, T.; Rezende, D. Variational Information Bottleneck for Semi-Supervised Classification. *Entropy* **2020**, *22*, 943. [CrossRef]
8. Amjad, R.A.; Geiger, B.C. Learning Representations for Neural Network-Based Classification Using the Information Bottleneck Principle. *IEEE Trans. Pattern. Anal. Mach. Intell.* **2019**, *42*, 2225–2239. [CrossRef]
9. Uğur, Y.; Arvanitakis, G.; Zaidi, A. Variational Information Bottleneck for Unsupervised Clustering: Deep Gaussian Mixture Embedding. *Entropy* **2020**, *22*, 213. [CrossRef]
10. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. In Proceedings of the Thirty-Seventh Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, USA, 22–24 September 1999; pp. 368–377.
11. Cover, T.M. *Elements of Information Theory*; John Wiley & Sons: Hoboken, NJ, USA, 1999.
12. Voloshynovskiy, S.; Kondah, M.; Rezaeifar, S.; Taran, O.; Hotolyak, T.; Rezende, D.J. Information bottleneck through variational glasses. In Proceedings of the Workshop on Bayesian Deep Learning, NeurIPS, Vancouver, Canada, 13 December 2019.
13. Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In Proceedings of the International Conference on Machine Learning, PMLR, Virtually, 18–24 July 2021; pp. 12310–12320.
14. Shwartz-Ziv, R.; LeCun, Y. To Compress or Not to Compress–Self-Supervised Learning and Information Theory: A Review. *arXiv* **2023**, arXiv:2304.09355.
15. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
16. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
17. Rezende, D.; Mohamed, S. Variational inference with normalizing flows. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 1530–1538.
18. Pidhorskyi, S.; Adjeroh, D.A.; Doretto, G. Adversarial latent autoencoders. In Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE/CVF, Virtually, 14–19 June 2020; pp. 14104–14113.
19. Achille, A.; Soatto, S. Information Dropout: Learning Optimal Representations Through Noisy Computation. *IEEE Trans. Pattern. Anal. Mach. Intell.* **2018**, *40*, 2897–2905. [CrossRef]
20. Razeghi, B.; Calmon, F.P.; Gündüz, D.; Voloshynovskiy, S. Bottlenecks CLUB: Unifying Information-Theoretic Trade-Offs Among Complexity, Leakage, and Utility. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 2060–2075. [CrossRef]
21. Tian, Y.; Pang, G.; Liu, Y.; Wang, C.; Chen, Y.; Liu, F.; Singh, R.; Verjans, J.W.; Wang, M.; Carneiro, G. Unsupervised Anomaly Detection in Medical Images with a Memory-augmented Multi-level Cross-attentional Masked Autoencoder. *arXiv* **2022**, arXiv:2203.11725.
22. Patel, A.; Tudosiu, P.D.; Pinaya, W.H.; Cook, G.; Goh, V.; Ourselin, S.; Cardoso, M.J. Cross Attention Transformers for Multi-modal Unsupervised Whole-Body PET Anomaly Detection. *J. Mach. Learn. Biomed. Imaging* **2023**, *2*, 172–201. [CrossRef]
23. Golling, T.; Nobe, T.; Proios, D.; Raine, J.A.; Sengupta, D.; Voloshynovskiy, S.; Arguin, J.F.; Martin, J.L.; Pilette, J.; Gupta, D.B.; et al. The Mass-ive Issue: Anomaly Detection in Jet Physics. *arXiv* **2023**, arXiv:2303.14134.
24. Buhmann, E.; Diefenbacher, S.; Eren, E.; Gaede, F.; Kasieczka, G.; Korol, A.; Krüger, K. Getting high: High fidelity simulation of high granularity calorimeters with high speed. *Comput. Softw. Big Sci.* **2021**, *5*, 13. [CrossRef]
25. Buhmann, E.; Diefenbacher, S.; Hundhausen, D.; Kasieczka, G.; Korcari, W.; Eren, E.; Gaede, F.; Krüger, K.; McKeown, P.; Rustige, L. Hadrons, better, faster, stronger. *Mach. Learn. Sci. Technol.* **2022**, *3*, 025014. [CrossRef]
26. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; Lerchner, A. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
27. Zhao, S.; Song, J.; Ermon, S. InfoVAE: Information Maximizing Variational Autoencoders. *arXiv* **2017**, arXiv:1706.02262.
28. Mohamed, S.; Lakshminarayanan, B. Learning in Implicit Generative Models. *arXiv* **2016**, arXiv:1610.03483.
29. Larsen, A.B.L.; Sønderby, S.K.; Larochelle, H.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 19–24 June 2016; pp. 1558–1566.
30. Howard, J.N.; Mandt, S.; Whiteson, D.; Yang, Y. Learning to simulate high energy particle collisions from unlabeled data. *Sci. Rep.* **2022**, *12*, 7567. [CrossRef]

31. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 214–223.

32. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.

33. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE/CVF, Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410.

34. Brock, A.; Donahue, J.; Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv* **2018**, arXiv:1809.11096.

35. Sauer, A.; Schwarz, K.; Geiger, A. StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets. In Proceedings of the SIGGRAPH Conference. ACM, Vancouver, BC, Canada, 8–11 August 2022; pp. 1–10.

36. Image Generation on ImageNet 256 × 256. Available online: https://paperswithcode.com/sota/image-generation-on-imagenet-256x256 (accessed on 29 August 2023).

37. Image Generation on FFHQ 256 × 256. Available online: https://paperswithcode.com/sota/image-generation-on-ffhq-256-x-256 (accessed on 29 August 2023).

38. Papamakarios, G.; Nalisnick, E.; Rezende, D.J.; Mohamed, S.; Lakshminarayanan, B. Normalizing Flows for Probabilistic Modeling and Inference. *J. Mach. Learn. Res.* **2021**, *22*, 1–64.

39. Quétant, G.; Drozdova, M.; Kinakh, V.; Golling, T.; Voloshynovskiy, S. Turbo-Sim: A generalised generative model with a physical latent space. In Proceedings of the Workshop on Machine Learning and the Physical Sciences, NeurIPS, Virtually, 13 December 2021.

40. Bellagente, M.; Butter, A.; Kasieczka, G.; Plehn, T.; Rousselot, A.; Winterhalder, R.; Ardizzone, L.; Köthe, U. Invertible networks or partons to detector and back again. *SciPost Phys.* **2020**, *9*, 074. [CrossRef]

41. Belousov, Y.; Pulfer, B.; Chaban, R.; Tutt, J.; Taran, O.; Holotyak, T.; Voloshynovskiy, S. Digital twins of physical printing-imaging channel. In Proceedings of the IEEE International Workshop on Information Forensics and Security, Virtually, 12–16 December 2022; pp. 1–6.

42. McInnes, L.; Healy, J.; Saul, N.; Grossberger, L. UMAP: Uniform Manifold Approximation and Projection. *J. Open Source Softw.* **2018**, *3*, 861. [CrossRef]