Diffusion Based Representation Learning

Sarthak Mittal [*1 2] **Korbinian Abstreiter** [*3] **Stefan Bauer** [4 5] **Bernhard Schölkopf** [6] **Arash Mehrjou** [†3 6]

Abstract

Diffusion-based methods, represented as stochastic differential equations on a continuous-time domain, have recently proven successful as non-adversarial generative models. Training such models relies on denoising score matching, which can be seen as multi-scale denoising autoencoders. Here, we augment the denoising score matching framework to enable representation learning without any supervised signal. GANs and VAEs learn representations by directly transforming latent codes to data samples. In contrast, the introduced diffusion-based representation learning relies on a new formulation of the denoising score matching objective and thus encodes the information needed for denoising. We illustrate how this difference allows for manual control of the level of details encoded in the representation. Using the same approach, we propose to learn an infinite-dimensional latent code that achieves improvements on state-of-the-art models on semi-supervised image classification. We also compare the quality of learned representations of diffusion score matching with other methods like autoencoder and contrastively trained systems through their performances on downstream tasks. Finally, we also ablate with a different SDE formulation for diffusion models and show that the benefits on downstream tasks are still present on changing the underlying differential equation.

1. Introduction

Diffusion-based models have recently proven successful for generating images (Sohl-Dickstein et al., 2015; Song & Ermon, 2020; Song et al., 2020), graphs (Niu et al., 2020), shapes (Cai et al., 2020), and audio (Chen et al., 2020b; Kong et al., 2021). Two promising approaches apply stepwise perturbations to samples of the data distribution until the perturbed distribution matches a known prior (Song & Ermon, 2019; Ho et al., 2020). A model is then trained to estimate the reverse process, which transforms samples of the prior to samples of the data distribution (Saremi et al., 2018). Diffusion models were further refined (Nichol & Dhariwal, 2021; Luhman & Luhman, 2021) and even achieved better image sample quality than GANs (Dhariwal & Nichol, 2021; Ho et al., 2021; Mehrjou et al., 2017). Further, Song et al. showed that these frameworks are discrete versions of continuous-time perturbations modeled by stochastic differential equations and proposed a diffusion-based generative modeling framework on continuous time. Unlike generative models such as GANs and various forms of autoencoders, the original form of diffusion models does not come with a fixed architectural module that captures the representations of the data samples.

Learning desirable representations has been an integral component of generative models such as GANs and VAEs (Bengio et al., 2013; Radford et al., 2016; Chen et al., 2016; van den Oord et al., 2017; Donahue & Simonyan, 2019; Chen et al., 2020a; Schölkopf et al., 2021). Recent works on visual representation learning achieve impressive performance on the downstream task of classification by applying contrastive learning (Chen et al., 2020d; Grill et al., 2020; Chen & He, 2020; Caron et al., 2021; Chen et al., 2020c). However, contrastive learning requires additional supervision of augmentations that preserve the content of the data, and hence these approaches are not directly comparable to representations learned through generative systems like Variational Autoencoders (Kingma & Welling, 2013; Rezende et al., 2014) and the current work which are considered *fully* unsupervised. Moreover, training the encoder to output similar representation for different views of the same image removes information about the applied augmentations, thus the performance benefits are limited to downstream tasks that do not depend on the augmentation, which has to be known beforehand. Hence our proposed algorithm does not restrict the learned representations to specific downstream tasks and solves a more general problem instead. We provide a summary of contrastive learning approaches in Appendix A. Similar to our approach, Denoising Autoen-

[*]Equal contribution, [†] Senior authorship, [1]Mila [2]Université de Montréal [3]ETH Zürich [4]Helmholtz AI [5]Technical University of Munich [6]Max Planck Institute for Intelligent Systems. Correspondence to: Sarthak Mittal <sarthmit@gmail.com>, Arash Mehrjou <arash@distantvantagepoint.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

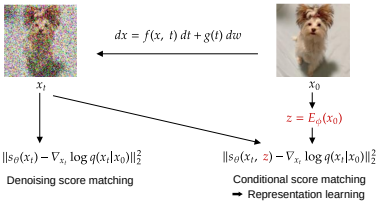


Figure 1. Conditional score matching with a parametrized latent code is representation learning. Denoising score matching estimates the score at each x_t; we add a latent representation z of the clean data x_0 as additional input to the score estimator.

coders (DAE) (Vincent et al., 2008) can be used to encode representations that can be manually controlled by adjusting the noise scale (Geras & Sutton, 2015; Chandra & Sharma, 2014; Zhang & Zhang, 2018). Note that, unlike DAEs, the encoder in our approach does not receive noisy data as input, but instead extracts features based on the clean images. For example, this key difference allows DRL to be used to limit the encoding to fine-grained features when focusing on low noise levels, which is not possible with DAEs.

Recently, there have been some works that rely on additional encoders in the model architecture of diffusion based models (Preechakul et al., 2022; Mittal et al., 2021a; Sinha et al., 2021). Sinha et al. (2021) considers an autoencoder based setup with the diffusion model defining the prior whereas Pandey et al. (2022) considers the opposite where a diffusion model is used to further improve the decoded samples from a VAE. Preechakul et al. (2022) is a concurrent work that is closest to our setup, however, instead of relying on time-conditioned encoder, they rely only on an unconditional encoder. Further, they concentrate more on generation-based tasks while our approach focuses more on evaluating the representations learned for downstream tasks.

The main contributions of this work are

- We present an alternative formulation of the denoising score matching objective, showing that the objective cannot be reduced to zero. We leverage this property to learn representations for downstream tasks.
- We introduce Diffusion-based Representation Learning (DRL), a novel framework for representation learning in diffusion-based generative models. We show how this framework allows for manual control of the level of details encoded in the representation through an infinite-dimensional code. We evaluate the proposed approach on downstream tasks using the learned representations directly as well as using it as a pre-training step for semi-supervised image classification, thereby improving state-of-the-art approaches for the latter.
- We evaluate the effect of the initial noise scale and achieve significant improvements in sampling speed, which is a bottleneck in diffusion-based generative models compared with GANs and VAEs, without sacrificing image quality.

1.1. Diffusion-based generative modeling

We first give a brief overview of the technical background for the framework of the diffusion-based generative model as described in (Song et al., 2021b). The forward diffusion process of the data is modeled as an SDE on a continuous-time domain $t \in [0, T]$. Let $x_0 \in \mathbb{R}^d$ denote a sample from the data distribution $x_0 \sim p_0$, where d is the data dimension. The trajectory $(x_t)_{t\in[0,T]}$ of data samples is a function of time determined by the diffusion process. The SDE is chosen such that the distribution $p_{0T}(x_T|x_0)$ for any sample $x_0 \sim p_0$ can be approximated by a known prior distribution. Notice that the subscript $0T$ of p_{0T} refers to the conditional distribution of the diffused data at time T given the data at time 0. For simplicity we limit the remainder of this paper to the so-called Variance Exploding SDE (Song et al., 2021b), that is,

$$\mathrm{d}x = f(x,t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathrm{w} := \sqrt{\frac{\mathrm{d}[\sigma^2(t)]}{\mathrm{d}t}}\,\mathrm{d}\mathrm{w}, \quad (1)$$

where w is the standard Wiener process. The perturbation kernel of this diffusion process has a closed-form solution being $p_{0t}(x_t|x_0) = \mathcal{N}(x_t; x_0, [\sigma^2(t) - \sigma^2(0)]I)$. It was shown by Anderson (1982) that the reverse diffusion process is the solution to the following SDE:

$$\mathrm{d}x = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]\,\mathrm{d}t + g(t)\,\mathrm{d}\overline{\mathrm{w}}, \quad (2)$$

where $\overline{\mathrm{w}}$ is the standard Wiener process when the time moves backwards. Thus, given the score function $\nabla_x \log p_t(x)$ for all $t \in [0, T]$, we can generate samples from the data distribution $p_0(x)$. In order to learn the score function, the simplest objective is Explicit Score Matching (ESM) (Hyvärinen & Dayan, 2005), that is,

$$\mathbf{E}_{x_t}\left[\|s_\theta(x_t,t) - \nabla_{x_t}\log p_t(x_t)\|_2^2\right]. \quad (3)$$

Since the ground-truth score function $\nabla_{x_t}\log p_t(x_t)$ is generally not known, one can apply denoising score matching (DSM) (Vincent, 2011), which is defined as the following:

$$\begin{aligned} J_t^{DSM}(\theta) =& \mathbf{E}_{x_0}\{\mathbf{E}_{x_t|x_0}[\|s_\theta(x_t,t) \\ &- \nabla_{x_t}\log p_{0t}(x_t|x_0)\|_2^2]\}. \end{aligned} \quad (4)$$

The training objective over all t is augmented by Song et al. (2021b) with a time-dependent positive weighting function $\lambda(t)$, that is, $J^{DSM}(\theta) = \mathbf{E}_t\left[\lambda(t)J_t^{DSM}(\theta)\right]$. One can also achieve class-conditional generation in diffusion-based models by training an additional time-dependent classifier $p_t(y|x_t)$ (Song et al., 2021b)). In particular, the conditional score for a fixed y can be expressed as the sum of

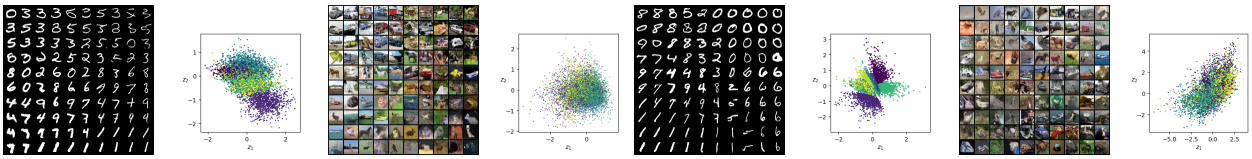


Figure 2. Results of proposed DRL models trained on MNIST and CIFAR-10 with point clouds visualizing the latent representation of test samples, colored according to the digit class. The models are trained with **Left:** uniform sampling of t and **Right:** a focus on high noise levels. Samples are generated from a grid of latent values ranging from -1 to 1.

the unconditional score and the score of the classifier, that is, $\nabla_{x_t} \log p_t(x_t|y) = \nabla_{x_t} \log p_t(x_t) + \nabla_{x_t} \log p_t(y|x_t)$. We take motivation from an alternative way to allow for controllable generation, which, given supervised samples $(x, y(x))$, uses the following training objective for each time t

$$J_t^{CSM}(\theta) = \mathbf{E}_{x_0}\{\mathbf{E}_{x_t|x_0}[\|s_\theta(x_t, t, y(x_0)) - \nabla_{x_t} \log p_{0t}(x_t|x_0)\|_2^2]\}. \tag{5}$$

The objective in Equation 5 is minimized if and only if the model equals the conditional score function $\nabla_{x_t} \log p_t(x_t|y(x_0) = \hat{y})$ for all labels $\hat{y}$.

2. Diffusion-based Representation Learning

We begin this section by presenting an alternative formulation of the Denoising Score Matching (DSM) objective, which shows that this objective cannot be made arbitrarily small. Formally, the formula of the DSM objective can be rearranged as

$$J_t^{DSM}(\theta) = \mathbf{E}_{x_0}\{\mathbf{E}_{x_t|x_0}\big[\|s_\theta(x_t, t) - \nabla_{x_t} \log p_t(x_t)\|_2^2 + \|\nabla_{x_t} \log p_{0t}(x_t|x_0) - \nabla_{x_t} \log p_t(x_t)\|_2^2\big]\}. \tag{6}$$

The above formulation holds, because the DSM objective in Equation 4 is minimized when $\forall x_t : s_\theta(x_t, t) = \nabla_{x_t} \log p_t(x_t)$, and differs from ESM in Equation 3 only by a constant (Vincent, 2011). Hence, the constant is equal to the minimum achievable value of the DSM objective. A detailed proof is included in the Appendix B.

It is noteworthy that the second term in the right-hand side of the Equation 6 does not depend on the learned score function of x_t for every $t \in [0, T]$. Rather, it is influenced by the diffusion process that generates x_t from x_0. This observation has not been emphasized previously, probably because it has no direct effect on the learning of the score function, which is handled by the second term in the Equation 6. However, the additional constant has major implications for finding other hyperparameters such as the function $\lambda(t)$ and the choice of $\sigma(t)$ in the forward SDE. As (Kingma et al., 2021) shows, changing the integration variable from time to signal-to-noise ratio (SNR) simplifies the diffusion loss such that it only depends on the end values of SNR. Hence, the loss is invariant to the intermediate values of the noise schedule. However, the weight functions $\lambda(\cdot)$ is still an important hyper-parameter whose choice might be affected by the non-vanishing constant in Equation 6.

To the best of our knowledge, there is no known theoretical justification for the values of $\sigma(t)$. While these hyperparameters could be optimized in ESM using gradient-based learning, this ability is severely limited by the non-vanishing constant in Equation 6.

Even though the non-vanishing constant in the denoising score matching objective presents a burden in multiple ways such as hyperparameter search and model evaluation, it provides an opportunity for latent representation learning, which will be described in the following sections. We note that this is different from Sinha et al. (2021); Mittal et al. (2021b) as they consider a Variational Autoencoder model followed by diffusion in the latent space, where their representation learning objective is still guided by reconstruction. Contrary to this, our representation learning approach does not utilize a variational autoencoder model and is guided by denoising instead. Our approach is similar to Preechakul et al. (2022) but we also condition the encoder system on the time-step, thereby improving representation capacity and leading to parameterized curve-based representations.

2.1. Learning latent representations

Since supervised data is limited and rarely available, we propose to learn a labeling function $y(x_0)$ at the same time as optimizing the conditional score matching objective in Equation 5. In particular, we represent the labeling function as a trainable encoder $E_\phi : \mathbb{R}^d \to \mathbb{R}^c$, where $E_\phi(x_0)$ maps the data sample x_0 to its corresponding code in the c-dimensional latent space. The code is then used as additional input to the score model. Formally, the proposed learning objective for Diffusion-based Representation Learning (DRL) is the following:

$$J^{DRL}(\theta, \phi) = \mathbf{E}_{t,x_0,x_t}[\lambda(t)\|s_\theta(x_t, t, E_\phi(x_0)) - \nabla_{x_t} \log p_{0t}(x_t|x_0)\|_2^2 + \gamma\|E_\phi(x_0)\|_1] \tag{7}$$

where we add a small amount of L_1 regularization, controlled by γ, on the output of the trainable encoder.

To get a better idea of the above objective, we provide an

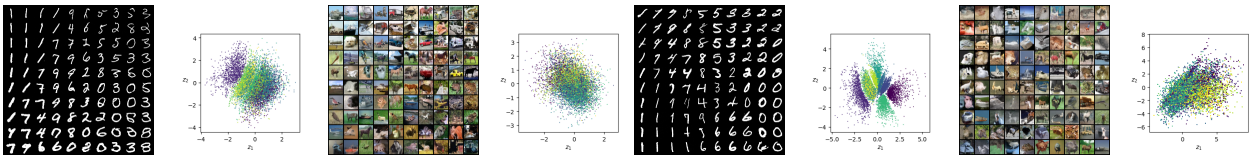


Figure 3. Results of proposed VDRL models trained on MNIST and CIFAR-10 with point clouds visualizing the latent representation of test samples, colored according to the digit class. The models are trained with **Left:** uniform sampling of t and **Right:** a focus on high noise levels. Samples are generated from a grid of latent values ranging from -2 to 2.

intuition for the role of $E_\phi(x_0)$ in the input of the model. The model $s_\theta(\cdot,\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^c \to \mathbb{R}^d$ is a vector-valued function whose output points to different directions based on the value of its third argument. In fact, $E_\phi(x_0)$ selects the direction that best recovers x_0 from x_t. Hence, when optimizing over ϕ, the encoder learns to extract the information from x_0 in a reduced-dimensional space that helps recover x_0 by denoising x_t.

We show in the following that Equation 7 is a valid representation learning objective. The score of the perturbation kernel $\nabla_{x_t} \log p_{0t}(x_t|x_0)$ is a function of only t, x_t and x_0. Thus, the objective can be reduced to zero if all information about x_0 is contained in the latent representation $E_\phi(x_0)$. When $E_\phi(x_0)$ has no mutual information with x_0, the objective can only be reduced up to the constant in Equation 6. Hence, our proposed formulation takes advantage of the non-zero lower-bound of Equation 6, which can only vanish when the encoder $E_\phi(\cdot)$ properly distills information from the unperturbed data into a latent code, which is an additional input to the score model. These properties show that Equation 7 is a valid objective for representation learning.

Our proposed representation learning objective enjoys the continuous nature of SDEs, a property that is not available in many previous representation learning methods (Radford et al., 2016; Chen et al., 2016; Locatello et al., 2019). In DRL, the encoder is trained to represent the information needed to denoise x_0 for different levels of noise $\sigma(t)$. We hypothesize that by adjusting the weighting function $\lambda(t)$, we can manually control the granularity of the features encoded in the representation and provide empirical evidence as support. Note that $t \to T$ is associated with higher levels of noise and the mutual information of x_t and x_0 starts to vanish. In this case, denoising requires all information about x_0 to be contained in the code. In contrast, $t \to 0$ corresponds to low noise levels and hence x_t contains coarse-grained features of x_0 and only fine-grained properties may have been washed out. Hence, the encoded representation learns to keep the information needed to recover these fine-grained details. We provide empirical evidence to support this hypothesis in Section 3.

It is noteworthy that E_ϕ does not need to be a deterministic function and can be a probabilistic map similar to the encoder of VAEs. In principle, it can be viewed as an information channel that controls the amount of information that the diffusion model receives from the initial point of the diffusion process. With this perspective, any deterministic or stochastic function that can manipulate $I(x_t, x_0)$, the mutual information between x_0 and x_t, can be used. This opens up the room for stochastic encoders similar to VAEs which we call Variational Diffusion-based Representation Learning (VDRL). The formal objective of VDRL is

$$\begin{aligned} J^{VDRL}(\theta,\phi) = \mathbf{E}_{t,x_0,x_t}[\mathbf{E}_{z\sim E_\phi(Z|x_0)}[\lambda(t)\|s_\theta(x_t,t,z) \\ - \nabla_{x_t}\log p_{0t}(x_t|x_0)\|_2^2] \\ + \mathcal{D}_{\mathrm{KL}}(E_\phi(Z|x_0)||\mathcal{N}(Z;0,I)] \end{aligned} \tag{8}$$

2.2. Infinite-dimensional representation of data

We now present an alternative version of DRL where the representation is a function of time. Instead of emphasizing on different noise levels by weighting the training objective, as done in the previous section, we can provide the time t as input to the encoder. Formally, the new objective is

$$\begin{aligned} \mathbf{E}_{t,x_0,x_t}[\lambda(t)\|s_\theta(x_t,t,E_\phi(x_0,t)) \\ - \nabla_{x_t}\log p_{0t}(x_t|x_0)\|_2^2 + \gamma\|E_\phi(x_0,t)\|_1] \end{aligned} \tag{9}$$

where $E_\phi(x_0)$ in Equation 7 is replaced by $E_\phi(x_0,t)$. Intuitively, it allows the encoder to extract the necessary information of x_0 required to denoise x_t for any noise level. This leads to richer representation learning since normally in autoencoders or other *static* representation learning methods, the input data $x_0 \in \mathbb{R}^d$ is mapped to a single point $z \in \mathbb{R}^c$ in the latent space. In contrast, we propose a richer representation where the input x_0 is mapped to a curve in $\mathbb{R}^c$ instead of a single point. Hence, the learned latent code is produced by the map $x_0 \to (E_\phi(x_0,t))_{t\in[0,T]}$ where the infinite-dimensional object $(E_\phi(x_0,t))_{t\in[0,T]}$ is the encoding for x_0.

Proposition 2.1. *For any downstream task, the infinite-dimensional code $(E_\phi(x_0,t))_{t\in[0,T]}$ learned using the objective in Equation 9 is at least as good as finite-dimensional static codes learned by the reconstruction of x_0.*

Proof sketch. Let $L_D(z,y)$ be the per-sample loss for a supervised learning task calculated for the pair (z,y) where $z = z(x,t)$ is the representation learned for the input x at

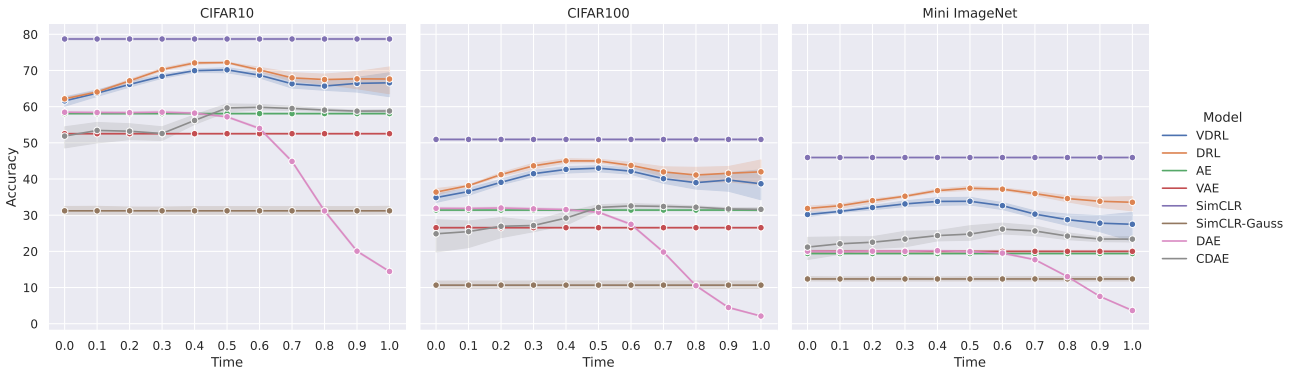


Figure 4. Comparing the performance of the proposed diffusion-based representations (DRL and VDRL) with the baselines that include autoencoder (AE), variational autoencoder (VAE), simple contrastive learning (simCLR) and its restricted variant (simCLR-Gauss) which exclude domain-specific data augmentation from the original simCLR algorithm.

time t and y is the label. The representation function is also a function of the scalar t that takes values from a closed subset U of R. For any value $s \in U$, it is obvious that

$$min_{t \in U} L_D(z(x,t), y) < L_D(z(x,s), y). \tag{10}$$

Taking into account the extra argument t, the representation function $z(x,t)$ can be seen as an infinite dimensional representation. The argument t actually controls which representation of x has to be passed to the downstream task. The conventional representation learning algorithms correspond to choosing the t argument apriori and keep it fixed independent of x. Here, by minimizing over t, the passed representation cannot be worse than the results of conventional representation learning methods. Note that $L_D(\cdot,\cdot)$ here can be any metric that we require, however gradient-based learning and optimization issues can still affect the actual performance achieved .

The score matching objective can be seen as a reconstruction objective of x_0 conditioned on x_t. The terminal time T is chosen large enough so that x_T is independent of x_0, hence the objective for $t = T$ is equal to a reconstruction objective without conditioning. Therefore, there exists a $t \in [0, T]$ where the learned representation $E_\phi(x_0, t)$ is the same representation learned by the reconstruction objective of a vanilla autoencoder. The full proof for Proposition 2.1 can be found in the Appendix C

A downstream task can leverage this rich encoding in various ways, including the use of either the static code for a fixed t, or the use of the whole trajectory $(E_\phi(x_0, t))_{t \in [0,T]}$ as input. We posit the conjecture that the proposed rich representation is helpful for downstream tasks when used for pretraining, where the value of t could either be a model selection parameter or be jointly optimized with other parameters during training. We leave investigations along these directions as important future work. We show the performance of the proposed model on downstream tasks in Section 3.1 and also evaluate it on semi-supervised image classification in Section 3.2.

3. Results

For all experiments, we use the same function $\sigma(t), t \in [0,1]$ as in Song et al. (2021b), which is $\sigma(t) = \sigma_{\min} (\sigma_{\max}/\sigma_{\min})^t$, where $\sigma_{\min} = 0.01$ and $\sigma_{\max} = 50$. Further, we use a 2d latent space for all qualitative experiments (Section 3.3) and 128 dimensional latent space for the downstream tasks (Section 3.1) and semi-supervised image classification (Section 3.2). We also set $\lambda(t) = \sigma^2(t)$, which has been shown to yield the KL-Divergence objective (Song et al., 2021a). Our goal is not to produce state-of-the-art image quality, rather showcase the representation learning method. Because of that and also limited computational resources, we did not carry out an extensive hyperparameter sweep (check Appendix D for details). Note that all experiments were conducted on a single RTX8000 GPU, taking up to 30 hours of wall-clock time, which only amounts to 15% of the iterations proposed in (Song et al., 2021b).

3.1. Downstream Classification

We directly evaluate the representations learned by different algorithms on downstream classification tasks for CIFAR10, CIFAR100, and Mini-ImageNet datasets. The representation is first learned using the proposed diffusion-based method. Then, the encoder (either deterministic or probabilistic) is frozen and a single-layered neural network is trained on top of it for the downstream prediction task. For the baselines, we consider an Autoencoder (AE), a Variational Autoencoder (VAE), two versions of Denoising Autoencoders (DAE and CDAE) and two verisons of Contrastive Learning (SimCLR(Chen et al., 2020c) and SimCLR-Gauss explained below) setup to compare with the proposed methods (DRL and VDRL). Figure 4 shows that DRL and VDRL outperforms autoencoder-styled baselines as well as the restricted contrastive learning baseline.

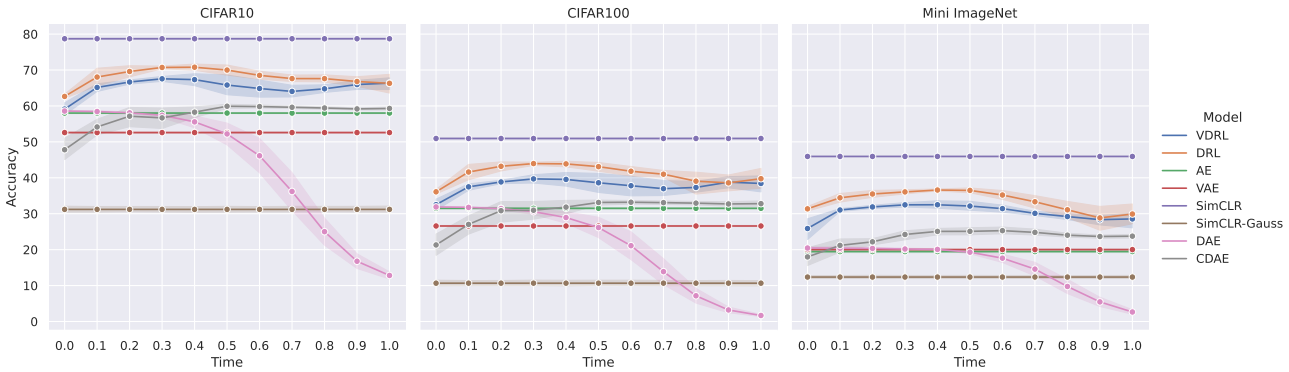


Figure 5. Comparing the performance of the proposed diffusion-based representations (DRL and VDRL) with the baselines that include autoencoder (AE), variational autoencoder (VAE), simple contrastive learning (simCLR) and its restricted variant (simCLR-Gauss) which exclude domain-specific data augmentation from the original simCLR algorithm.

Standard Autoencoders— Standard autoencoders (AE and VAE) rely on learning of representations of the input data using an encoder in such a way that it can be reconstructed back, using a decoder, solely based on the representation learned. Such systems can be trained without any regularization on the representation space (AE), or in a probabilistic fashion which relies on variational inference and ultimately leads to a KL-Divergence based regularization on the representation space (VAE). Figure 4 shows that the time-axis is not meaningful for such training, as expected.

Denoising Autoencoders— While the problem of reconstruction is easily solved given a big enough network (i.e. capable of learning the identity mapping), this problem can be made harder by considering a noisy version of the data as input with the task of predicting its denoised version, as opposed to vanilla reconstruction in standard autoencoders. Such approaches are referred to as Denoising Autoencoders, and we consider its two variants. In the first variant, DAE, a noisy version of the image is given as input x_t (higher t implying more noise) and the task of the model is to predict the denoised version x_0. Since larger t implies learning of representations from more noise, we can see a sharp decline in performance of DAE systems with increasing t in Figure 4. The second variant, CDAE, considers x_t as the noisy input again, but predicts the denoised version based on a representation of x_t combined with a learned time-conditioned representation of the true input $E_\phi(x_0, t)$, similar to the DRL setups. This approach is arguably similar to DRL with the sole difference being that $E_\phi(\cdot, \cdot)$ in DRL had the incentive of predicting the right score function, whereas in CDAE the incentive is to denoise in a single step. As highlighted in Figure 4, the performance increases with increasing t because the encoder $E_\phi(\cdot, \cdot)$ is useless in low-noise settings (as all the data is already there in the input) but becomes increasingly meaningful as noise increases.

Restricted SimCLR— While we compare against the standard SimCLR model, to obtain a fair comparison, we restricted the transformations used by the simCLR method to the additive pixel-wise Gaussian noise (SimCLR-Gauss) as this was the only domain-agnostic transformation in the SimCLR pipeline. The original SimCLR expectedly outperforms the other methods because it uses the privileged information injected by the employed data augmentation methods. For example, random cropping is an inductive bias that reflects the spatial regularity of the images. Even though it is possible to strengthen our method and autoencoder-based baselines such as VAEs with such augmentation-based strategies, it still doesn't provide the additional inductive bias of preservation of high-level information in the presence of these augmentations, which SimCLR directly uses. Thus, we restricted all baselines to the generic setting without this inductive bias and leave the domain-specific improvements for future work.

It is seen that the DRL and VDRL methods significantly outperform the baselines on all the datasets at a number of different time-steps t. We further evaluate the infinite-dimensional representation on few-shot image classification using the representation at different timescales as input. The detailed results are shown in Appendix E. In summary, the representations of DRL and VDRL achieve significant improvements as compared to an autoencoder or VAE for several values of t .

Overall the results align with the theoretical argument of Proposition 2.1 that the rich representation of DRL is at least as good as the static code learned using a reconstruction objective. It further shows that in practice, the infinite-dimensional code is superior to the static (finite-dimensional) representation for downstream applications such as image classification by a significant margin.

As a further analysis, we consider the same experiments when the DRL models are trained on the Variance Preserv-

Pretraining		LaplaceNet		Ours		
		None		DRL		VDRL
Mixup		No	Yes	No	Yes	No
Dataset	#labels					
CIFAR-10	100	73.68	75.29	74.31	64.67	**81.63**
	500	91.31	92.53	**92.70**	92.31	**92.79**
	1000	92.59	93.13	**93.24**	**93.42**	**93.60**
	2000	94.00	93.96	**94.18**	93.91	93.96
	4000	94.73	94.97	94.75	**95.22**	**95.00**
CIFAR-100	1000	55.58	55.24	**55.85**	**55.74**	**56.47**
	4000	67.07	67.25	67.22	**67.47**	**67.54**
	10000	73.19	72.84	**73.31**	**73.66**	**73.50**
	20000	75.80	76.07	**76.46**	**76.88**	**76.64**
Mini ImageNet	4000	58.40	58.84	**58.95**	**59.29**	**59.14**
	10000	66.65	66.80	**67.31**	66.63	**67.46**

Table 1. Comparison of classifier accuracy in % for different pretraining settings. Scores better than the SOTA model (LaplaceNet) are in **bold**. "DRL" pretraining is our proposed representation learning, and "VDRL" the respective version which uses a probabilistic encoder.

ing SDE formulation (Song et al., 2021b).

$$\mathrm{d}x = -\frac{1}{2}\beta(t)x\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}w, \tag{11}$$

Figure 5 shows that even in this formualtion, DRL and VDRL models outperform their autoencoder and denoising autoencoder competitors and perform better than restricted constrastive learning, showing that this approach can be easily adapted to various different diffusion models.

3.2. Semi-Supervised Image Classification

The current state-of-the-art model for many semi-supervised image classification benchmarks is LaplaceNet (Sellars et al., 2021). It alternates between assigning pseudo-labels to samples and supervised training of a classifier. The key idea is to assign pseudo-labels by minimizing the graphical Laplacian of the prediction matrix, where similarities of data samples are calculated on a hidden layer representation in the classifier. Note that LaplaceNet applies *mixup* (Zhang et al., 2017) that changes the input distribution of the classifier. We evaluate our method with and without mixup on CIFAR-10 (Krizhevsky et al., a), CIFAR-100 (Krizhevsky et al., b) and MiniImageNet (Vinyals et al., 2016).

In the following, we evaluate the infinite-dimensional representation $(E_\phi(x_0, t))_{t\in[0,T]}$ on semi-supervised image classification, where we use DRL and VDRL as pretraining for the LaplaceNet classifier. Table 1 depicts the classifier accuracy on test data for different pretraining settings. Details for architecture and hyperparameters are described in Appendix G.

Our proposed pretraining using DRL significantly improves the baseline and often surpasses the state-of-the-art performance of LaplaceNet. Most notable are the results of DRL and VDRL without mixup, which achieve high accuracies without being specifically tailored to the downstream task of classification. Note that pretraining the classifier as part of an autoencoder did not yield any improvements (Table 4 in the Appendix). Combining DRL with mixup yields inconsistent improvements, results are reported in Table 5 of the Appendix. In addition, DRL pretraining achieves much better performances when only limited computational resources are available (Tables 2, 3 in the Appendix).

3.3. Qualitative Results

We first train a DRL model with L_1-regularization on the latent code on MNIST (LeCun & Cortes, 2010) and CIFAR-10. Figure 2 (*left*) shows samples from a grid over the latent space and a point cloud visualization of the latent values $z = E_\phi(x_0)$. For MNIST, we can see that the value of z_1 controls the stroke width, while z_2 weakly indicates the class. The latent code of CIFAR-10 samples mostly encodes information about the background color, which is weakly correlated to the class. The use of a probabilistic encoder (VDRL) leads to similar representations, as seen in Fig. 3 (*left*). We further want to point out that the generative process using the reverse SDE involves randomness and thus generates different samples for a single latent representation. The diversity of samples however steadily decreases with the dimensionality of the latent space, shown in Figure 7 of the Appendix.

Next, we analyze the behavior of the representation when adjusting the weighting function $\lambda(t)$ to focus on higher noise levels, which can be done by changing the sampling distribution of t. To this end, we sample $t \in [0, 1]$ such that $\sigma(t)$ is uniformly sampled from the interval $[\sigma_{\min}, \sigma_{\max}] = [0.01, 50]$. Figure 2 (*right*) shows the resulting representation of DRL and Figure 3 (*right*) for the VDRL results. As expected, the latent representation for MNIST encodes information about classes rather than fine-grained features such as stroke width. This validates our

hypothesis of Section 2.1 that we can control the granularity of features encoded in the latent space. For CIFAR-10, the model again only encodes information about the background, which contains the most information about the image class. A detailed analysis of class separation in the extreme case of training on single timescales is included in Appendix H.

Overall, the difference in the latent codes for varying $\lambda(t)$ shows that we can control the granularity encoded in the representation of DRL. This ability provides a significant advantage when there exists some prior information about the level of detail that we intend to encode in the target representation. We further illustrate how the representation encodes information for the task of denoising in the Appendix (Fig. 6).

We also provide further analysis into the impact of noise scales on generation in Appendix I.

4. Conclusion

We presented Diffusion-based Representation Learning (DRL), a new objective for representation learning based on conditional denoising score matching. In doing so, we turned the original non-vanishing objective function into one that can be reduced arbitrarily close to zero by the learned representation. We showed that the proposed method learns interpretable features in the latent space. In contrast to some of the previous approaches that required specialized architectural changes or data manipulations, denoising score matching comes with a natural ability to control the granularity of features encoded in the representation. We demonstrated that the encoder can learn to separate classes when focusing on higher noise levels and encodes fine-grained features such as stroke-width when mainly trained on smaller noise variance. In addition, we proposed an infinite-dimensional representation and demonstrated its effectiveness for downstream tasks such as few-shot classification. Using the representation learning as pretraining for a classifier, we were able to improve the results of LaplaceNet, a state-of-the-art model on semi-supervised image classification.

Starting from a different origin but conceptually close, contrastive learning as a self-supervised approach could be compared with our representation learning method. We should emphasize that there are fundamental differences both at theoretical and algorithmic levels between contrastive learning and our diffusion-based method. The generation of positive and negative examples in contrastive learning requires the domain knowledge of the applicable invariances. This knowledge might be hard to obtain in scientific domains such as genomics where the knowledge of invariance amounts to the knowledge of the underlying biology which in many cases is not known. However, our diffusion-based representation uses the natural diffusion process that is employed in score-based models as a continuous obfuscation of the information content. Moreover, unlike the loss function of the contrastive-based methods that are specifically designed to learn the invariances of manually augmented data, our method uses the same loss function that is used to learn the score function for generative models. The representation is learned based on a generic information-theoretic concept which is an encoder (information channel) that controls how much information of the input has to be passed to the score function at each step of the diffusion process. We also provided theoretical motivation for this information channel. The algorithm cannot ignore this source of information because it is the only way to reduce a non-negative loss arbitrarily close to zero.

Our experiments on diffusion-based representation learning methods highlight its benefits when compared to *fully* unsupervised models like autoencoders, variational or denoising. The proposed methodology does not rely on additional supervision regarding augmentations, and can be easily adapted to any representation learning paradigm that previously relied on reconstruction-based autoencoder methods.

Acknowledgements

SM would like to acknowledge the support of UNIQUE's and IVADO's scholarships towards his research. This research was enabled in part by compute resources provided by Mila (mila.quebec).

References

Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. ISSN 0304-4149. doi: https://doi.org/10.1016/0304-4149(82)90051-5. URL https://www.sciencedirect.com/science/article/pii/0304414982900515.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Bromley, J., Bentz, J., Bottou, L., Guyon, I., Lecun, Y., Moore, C., Sackinger, E., and Shah, R. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7:25, 08 1993. doi: 10.1142/S0218001493000339.

Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N., and Hariharan, B. Learning gradient fields for shape generation, 2020.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments, 2021.

Chandra, B. and Sharma, R. Adaptive noise schedule for denoising autoencoder. volume 8834, pp. 535–542, 11 2014. ISBN 978-3-319-12636-4. doi: 10.1007/978-3-319-12637-1_67.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *International Conference on Machine Learning*, pp. 1691–1703. PMLR, 2020a.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. Wavegrad: Estimating gradients for waveform generation, 2020b.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020c.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big self-supervised models are strong semi-supervised learners. *CoRR*, abs/2006.10029, 2020d. URL https://arxiv.org/abs/2006.10029.

Chen, X. and He, K. Exploring simple siamese representation learning, 2020.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets, 2016.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis, 2021.

Donahue, J. and Simonyan, K. Large scale adversarial representation learning, 2019.

Geras, K. J. and Sutton, C. Scheduled denoising autoencoders, 2015.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. URL https://arxiv.org/abs/2006.11239.

Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *arXiv preprint arXiv:2106.15282*, 2021.

Hyvärinen, A. and Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis, 2021.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). a. URL http://www.cs.toronto.edu/~kriz/cifar.html.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-100 (canadian institute for advanced research). b. URL http://www.cs.toronto.edu/~kriz/cifar.html.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Luhman, E. and Luhman, T. Knowledge distillation in iterative generative models for improved sampling speed, 2021.

Mehrjou, A., Schölkopf, B., and Saremi, S. Annealed generative adversarial networks. *arXiv preprint arXiv:1705.07505*, 2017.

Mittal, G., Engel, J., Hawthorne, C., and Simon, I. Symbolic music generation with diffusion models. *arXiv preprint arXiv:2103.16091*, 2021a.

Mittal, G., Engel, J. H., Hawthorne, C., and Simon, I. Symbolic music generation with diffusion models. *CoRR*, abs/2103.16091, 2021b. URL https://arxiv.org/abs/2103.16091.

Nichol, A. and Dhariwal, P. Improved denoising diffusion probabilistic models. *CoRR*, abs/2102.09672, 2021. URL https://arxiv.org/abs/2102.09672.

Niu, C., Song, Y., Song, J., Zhao, S., Grover, A., and Ermon, S. Permutation invariant graph generation via score-based generative modeling, 2020.

Pandey, K., Mukherjee, A., Rai, P., and Kumar, A. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. *arXiv preprint arXiv:2201.00308*, 2022.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Preechakul, K., Chatthee, N., Wizadwongsa, S., and Suwajanakorn, S. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10619–10629, 2022.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

Saremi, S., Mehrjou, A., Schölkopf, B., and Hyvärinen, A. Deep energy estimator networks. *arXiv preprint arXiv:1805.08306*, 2018.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, 2021.

Sellars, P., Avilés-Rivero, A. I., and Schönlieb, C. Laplacenet: A hybrid energy-neural model for deep semi-supervised classification. *CoRR*, abs/2106.04527, 2021. URL https://arxiv.org/abs/2106.04527.

Sinha, A., Song, J., Meng, C., and Ermon, S. D2C: diffusion-denoising models for few-shot conditional generation. *CoRR*, abs/2106.06819, 2021. URL https://arxiv.org/abs/2106.06819.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models, 2020.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *CoRR*, abs/1907.05600, 2019. URL http://arxiv.org/abs/1907.05600.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. *CoRR*, abs/2006.09011, 2020. URL https://arxiv.org/abs/2006.09011.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models, 2021a. URL https://arxiv.org/pdf/2101.09258v1.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations, 2021b.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. *CoRR*, abs/1711.00937, 2017. URL http://arxiv.org/abs/1711.00937.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390294. URL https://doi.org/10.1145/1390156.1390294.

Vinyals, O., Blundell, C., Lillicrap, T. P., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. *CoRR*, abs/1606.04080, 2016. URL http://arxiv.org/abs/1606.04080.

Zhang, H., Cissé, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017. URL http://arxiv.org/abs/1710.09412.

Zhang, Q. and Zhang, L. Convolutional adaptive denoising autoencoders for hierarchical feature extraction. *Front. Comput. Sci.*, 12(6):1140–1148, dec 2018. ISSN 2095-2228. doi: 10.1007/s11704-016-6107-0. URL https://doi.org/10.1007/s11704-016-6107-0.

A. Related work on contrastive learning

The core idea of contrastive learning is to learn representations that are similar for different views of the same image and distant for different images. In order to prevent the collapse of representations to a constant, various approaches have been introduced. SimCLRv2 directly includes a loss term repulsing negative image pairs in addition to the attraction of different views of positive pairs (Chen et al., 2020d)). In contrast, BYOL relies solely on positive pairs, preventing collapse by enforcing similarity between the encoded representation of an image and the output of a momentum encoder applied to a different view of the same image (Grill et al., 2020). An additional approach relies on online clustering and was proposed in SwAV (Caron et al., 2021). Training in SwAV is based on enforcing consistency between cluster assignments produced for different views of an image. Each of these methods relies on the foundation of Siamese networks (Bromley et al., 1993), which were shown to be competitive for unsupervised pretraining for classification networks on its own when including a stop-gradient operation on one of the branches (Chen & He, 2020).

B. Denoising Score Matching

The following is the proof for the new formulation of the denoising score matching objective in Equation 6.

Proof. It was shown by (Vincent, 2011) that Equation 4 is equal to explicit score matching up to a constant which is independent of θ, that is,

$$\mathbf{E}_{x_0}\{\mathbf{E}_{x_t|x_0}[\|s_\theta(x_t,t) - \nabla_{x_t}\log p_{0t}(x_t|x_0)\|_2^2]\} \tag{12}$$
$$= \mathbf{E}_{x_t}\left[\|s_\theta(x_t,t) - \nabla_{x_t}\log p_t(x_t)\|_2^2\right] + c. \tag{13}$$

As a consequence, the objective is minimized when the model equals the ground-truth score function $s_\theta(x_t,t) = \nabla_x \log p_t(x)$. Hence we have:

$$\mathbf{E}_{x_0}\{\mathbf{E}_{x_t|x_0}[\|\nabla_{x_t}\log p_t(x_t) - \nabla_{x_t}\log p_{0t}(x_t|x_0)\|_2^2]\} \tag{14}$$
$$= \mathbf{E}_{x_t}\left[\|\nabla_{x_t}\log p_t(x_t) - \nabla_{x_t}\log p_t(x_t)\|_2^2\right] + c \tag{15}$$
$$= c. \tag{16}$$

Combining these results leads to the claimed exact formulation of the Denoising Score Matching objective:

$$J_t^{DSM}(\theta) = \mathbf{E}_{x_0}\{\mathbf{E}_{x_t|x_0}[\|s_\theta(x_t,t) - \nabla_{x_t}\log p_{0t}(x_t|x_0)\|_2^2]\} \tag{17}$$
$$= \mathbf{E}_{x_t}\left[\|s_\theta(x_t,t) - \nabla_{x_t}\log p_t(x_t)\|_2^2\right] + c \tag{18}$$
$$\begin{aligned} &= \mathbf{E}_{x_t}\left[\|s_\theta(x_t,t) - \nabla_{x_t}\log p_t(x_t)\|_2^2\right] \\ &\quad + \mathbf{E}_{x_0}\{\mathbf{E}_{x_t|x_0}[\|\nabla_{x_t}\log p_t(x_t) - \nabla_{x_t}\log p_{0t}(x_t|x_0)\|_2^2]\} \end{aligned} \tag{19}$$
$$\begin{aligned} &= \mathbf{E}_{x_0}\{\mathbf{E}_{x_t|x_0}[\|\nabla_{x_t}\log p_{0t}(x_t|x_0) - \nabla_{x_t}\log p_t(x_t)\|_2^2 \\ &\qquad + \|s_\theta(x_t,t) - \nabla_{x_t}\log p_t(x_t)\|_2^2]\}. \end{aligned} \tag{20}$$

□

C. Representation learning

Here we present the proof for Proposition 2.1, stating that the infinite-dimensional code learned using DRL is at least as good as a static code learned using a reconstruction objective.

Proof. We assume that the distribution of the diffused samples at time $t = T$ matches a known prior $p_T(x_T)$. That is, $\int p(x_0)p_{0T}(x_T|x_0)\,\mathrm{d}x_0 = p_T(x_T)$. In practice T is chosen such that this assumption approximately holds.
Now consider the training objective in Equation 9 at time T, which can be transformed to a reconstruction objective in the

following way:

$$\lambda(T)\mathbf{E}_{x_0,x_T}\left[\|s_\theta(x_T,T,E_\phi(x_0,T))-\nabla_{x_T}\log p_{0T}(x_T|x_0)\|_2^2\right] \tag{21}$$

$$=\lambda(T)\mathbf{E}_{x_0}\mathbf{E}_{x_T\sim p_T(x_T)}\left[\left\|s_\theta(x_T,T,E_\phi(x_0,T))-\frac{x_0-x_T}{\sigma^2(T)}\right\|_2^2\right] \tag{22}$$

$$=\lambda(T)\sigma^{-4}(T)\mathbf{E}_{x_0}\mathbf{E}_{x_T\sim p_T(x_T)}\left[\|D_\theta(E_\phi(x_0,T))-x_0\|_2^2\right] \tag{23}$$

$$=\lambda(T)\sigma^{-4}(T)\mathbf{E}_{x_0}\left[\|D_\theta(E_\phi(x_0,T))-x_0\|_2^2\right], \tag{24}$$

where we replaced the score model with a Decoder model $s_\theta(x_T,T,E_\phi(x_0,T))=\frac{D_\theta(E_\phi(x_0,T))-x_T}{\sigma^2(T)}$ and replaced the score function of the perturbation kernel $\nabla_{x_T}\log p_{0T}(x_T|x_0)$ with its known closed-form solution $\frac{x_0-x_T}{\sigma^2(T)}$ determined by the Forward SDE in Equation 1. Hence the learned code at time $t=T$ is equal to a code learned using a reconstruction objective.

We model a downstream task as a minimization problem of a distance $d:\Omega\times\Omega\to\mathbb{R}$ in the feature space Ω between the true feature extractor $g:\mathbb{R}^d\to\Omega$ which maps data samples x_0 to a features space Ω and a model feature extractor $h_\psi:\mathbb{R}^c\to\Omega$ doing the same given the code as input. The following shows that the infinite-dimensional representation is at least as good as the static code:

$$\inf_t\min_\psi \mathbf{E}_{x_0}[d(h_\psi(E_\phi(x_0,t)),g(x_0))]\leq\min_\psi\mathbf{E}_{x_0}[d(h_\psi(E_\phi(x_0,T)),g(x_0))] \tag{25}$$

□

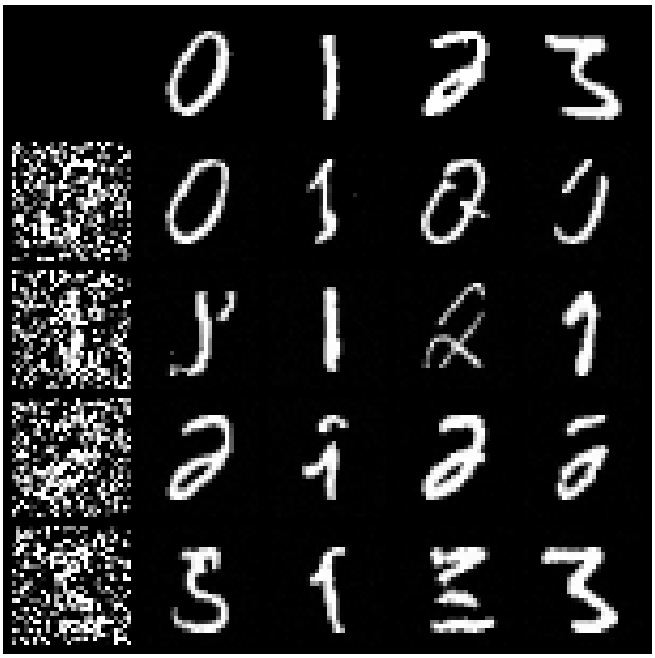
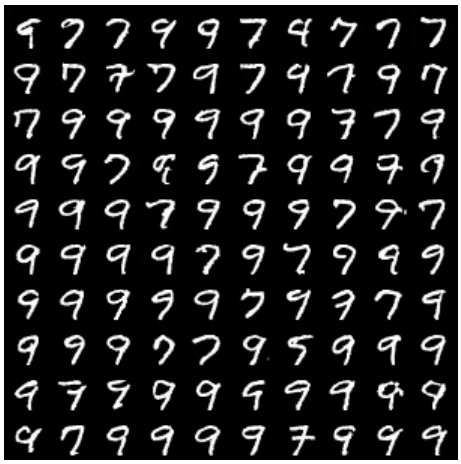


Figure 6. Samples generated starting from x_t (left column) using the diffusion model with the latent code of another x_0 (top row) as input. It shows that samples are denoised correctly only when conditioning on the latent code of the corresponding original image x_0.

D. Architecture and Hyperparameters

The model architecture we use for all experiments is based on "DDPM++ cont. (deep)" used for CIFAR-10 in (Song et al., 2021b). It is composed of a downsampling and an upsampling block with residual blocks at multiple resolutions. We did

(a) 2-dimensional



(b) 4-dimensional

(c) 8-dimensional

(d) 16-dimensional

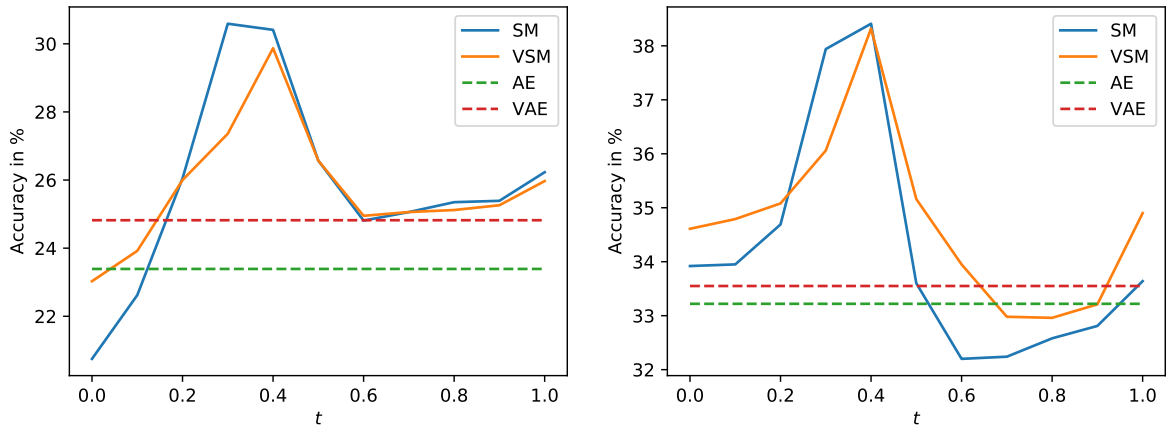
Figure 7. Samples generated using the same latent code for each generation, showing that the randomness of the code-conditional generation of DRL reduces in higher dimensional latent spaces.

not change any of the hyperparameters of the optimizer. Depending on the dataset, we adjusted the number of resolutions, number of channels per resolution, and the number of residual blocks per resolution in order to reduce training time.

For representation learning, we use an encoder with the same architecture as the downsampling block of the model, followed by another three dense layers mapping to a low dimensional latent space. Another four dense layers map the latent code back to a higher-dimensional representation. It is then given as input to the model in the same way as the time embedding. That is, each channel is provided with a conditional bias determined by the representation and time embedding at multiple stages of the downsampling and upsampling block.

Regularization of the latent space For both datasets, we use a regularization weight of 10^{-5} when applying L1-regularization, and a weight of 10^{-7} when using a probabilistic encoder regularized with KL-Divergence.

MNIST hyperparameters Due to the simplicity of MNIST, we only use two resolutions of size $28 \times 28 \times 32$ and $14 \times 14 \times 64$, respectively. The number of residual blocks at each resolution is set to two. In each experiment, the model is trained for $80k$ iterations. For a uniform sampling of σ we trained the models for an additional $80k$ iterations with a frozen encoder and uniform sampling of t.



(a) 100 labels

(b) 1000 labels

Figure 8. Classifier accuracies for few shot learning on given 8-dimensional representations learned using DRL (SM), VDRL (VSM), Autoencoder (AE) and Variational Autoencoder (VAE).

CIFAR-10 hyperparameters For the silhouette score analysis, we use three resolutions of size $32 \times 32 \times 32$, $16 \times 16 \times 32$, and $8 \times 8 \times 32$, again with only two residual blocks at each resolution. Each model is trained for $90k$ iterations.

CIFAR-10 (deep) hyperparameters While representation learning works for small models already, sample quality on CIFAR-10 is poor for models of the size described above. Thus for models used to generate samples, we use eight residual blocks per resolution and the following resolutions: $32 \times 32 \times 32$, $16 \times 16 \times 64$, $8 \times 8 \times 64$, and $4 \times 4 \times 64$. Each model is trained for $300k$ iterations. Note that this number of iterations is not sufficient for convergence, however capable of illustrating the representation learning with limited computational resources.

E. Evaluation of the infinite-dimensional representation

In order to evaluate our infinite-dimensional representation, we conduct an ablation study where we compare our proposed method with Autoencoders (AE) and Variational Autoencoders (VAE) on CIFAR-10 images. We measure the accuracy of an SVM provided by sklearn (Pedregosa et al., 2011) with default hyperparameters trained on the representation of 100 (resp. 1000) training samples and their class labels. For our time-dependent representation, this is done for fixed values of t between 0.0 and 1.0 in steps of 0.1. This is done for both DRL and VDRL, where we use a probabilistic encoder regularized by including an additional KL-Divergence term in the training objective. DRL and AE were regularized using L1-norm, and the regularization weight was optimized for each model independently.

Results for few-shot learning with fixed representations are shown in Figure 8. As expected, the accuracies when training on the score matching representations highly depend on the value of t. Overall our representation achieves much better scores when using the best t, and performs comparable to AE and VAE for $t = 1.0$. This aligns with Proposition 2.1 claiming that our representation learning method for $t = 1.0$ is similar to a static code learned using reconstruction objective. Note that the shape of the time-dependent classifier accuracies resembles the one of the silhouette scores of CIFAR-10 in 12. This is not surprising, since both training on single values of t and learning a time-dependent representation are both trained to find the optimal representation for a given value of t. We further want to point out that representation learning through score matching enjoys the training stability of diffusion-based generative models, which is often not the case in GANs and VAEs.

F. Downstream Image Classification

Architecture and Hyperparameters In all our experiments, we consider the small WideResNet model WRN-28-2 of (Sellars et al., 2021) as the encoder module for all of the different settings: diffusion representation learning, autoencoder and contrastive learning. We sample the time-steps at intervals of 0.1 from the range $0.0 - 1.0$. Corresponding to each time-step, we train a single layered non-linear MLP network for 50 epochs.

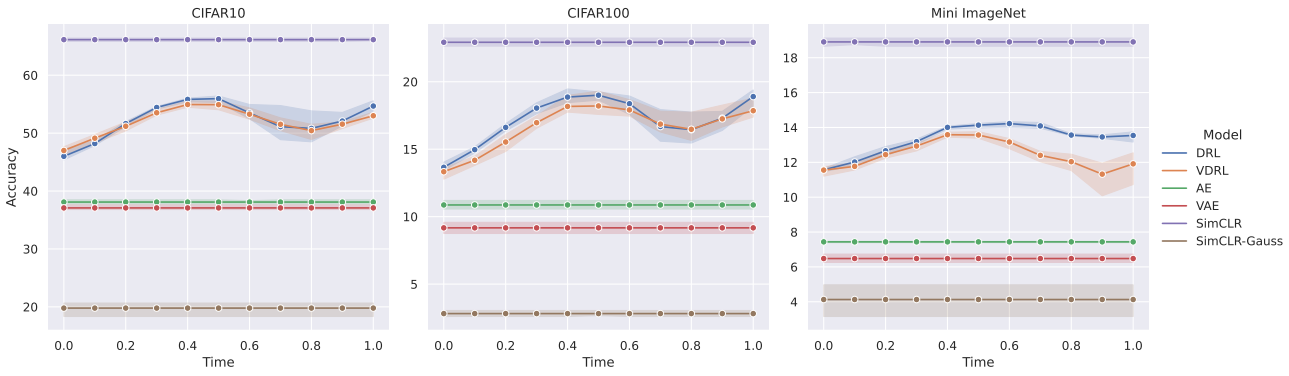


Figure 9. Comparing the low-data regime (1000 labels) downstream performance of the proposed diffusion-based representations (DRL and VDRL) with the baselines that include autoencoder (AE), variational autoencoder (VAE), simple contrastive learning (simCLR) and its restricted variant (simCLR-Gauss) which exclude domain-specific data augmentation from the original simCLR algorithm.

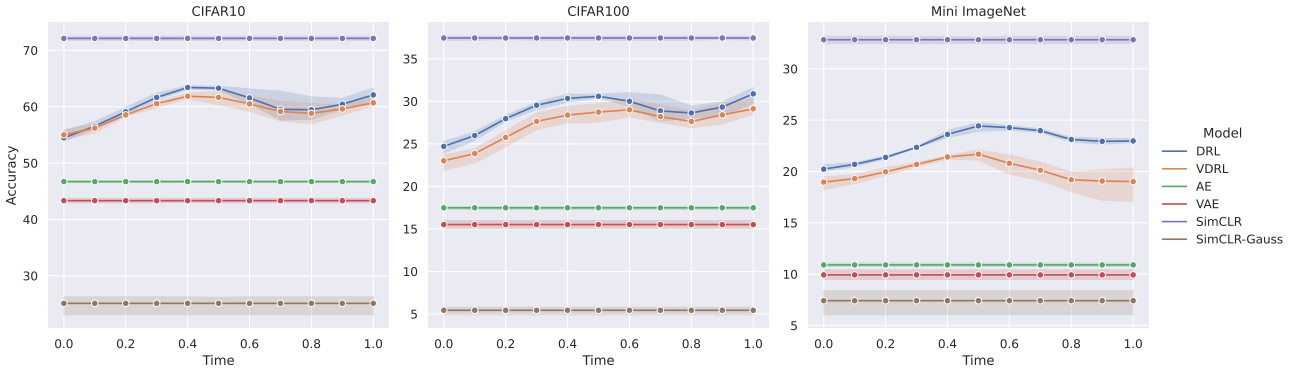


Figure 10. Comparing the low-data regime (5000 labels) downstream performance of the proposed diffusion-based representations (DRL and VDRL) with the baselines that include autoencoder (AE), variational autoencoder (VAE), simple contrastive learning (simCLR) and its restricted variant (simCLR-Gauss) which exclude domain-specific data augmentation from the original simCLR algorithm.

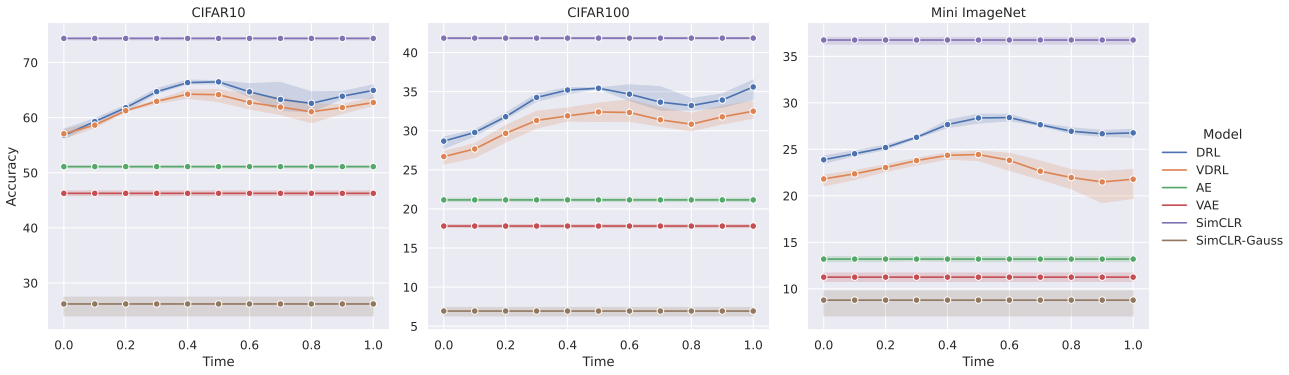


Figure 11. Comparing the low-data regime (10000 labels) downstream performance of the proposed diffusion-based representations (DRL and VDRL) with the baselines that include autoencoder (AE), variational autoencoder (VAE), simple contrastive learning (simCLR) and its restricted variant (simCLR-Gauss) which exclude domain-specific data augmentation from the original simCLR algorithm.

Dataset	#labels	No pretraining	Pretraining using DRL	Improvement
CIFAR-10	100	64.12	69.79	+5.67
	500	86.24	88.28	+2.04
	1000	87.48	88.56	+1.08
	2000	89.99	89.52	-0.47
	4000	90.15	91.13	+0.98
CIFAR-100	1000	45.14	48.04	+2.90
	4000	59.86	60.34	+0.48
	10000	64.83	65.80	+0.97
	20000	65.77	66.39	+0.62
MiniImageNet	4000	47.18	50.75	+3.57
	10000	58.66	58.62	-0.04

Table 2. Classifier accuracy in % with and without DRL as pretraining of the classifier when training for 100 epochs only.

Results with Limited Data We perform additional experiments where the encoder system is as before and kept frozen, but the MLP can only access a fraction of the training set for the downstream supervised classification task. We ablate over three different number of labels provided to the MLP: 1000, 5000 and 10000. The results for the different datasets can be seen in Figures 9-11 which shows that the trends are consistent even in low data regime.

G. Semi-supervised image classification

Architecture and Hyperparameters In all experiments, our encoder has the same architecture as the classifier, where the hidden layer used to measure similarities for assigning pseudo-labels in LaplaceNet is used as the latent code in representation learning. For all experiments, the input t to the encoder is included as a trainable parameter of the model and initialized with $t = 0.5$. As done in the original paper, we train the model for 260 iterations, where each iteration consists of assigning pseudo-labels and one epoch of supervised training on the assigned pseudo-labels. The training is preceded by 100 supervised epochs on the labeled data. We use the small WideResNet model WRN-28-2 of (Sellars et al., 2021) and the same hyperparameters as the authors.

Evaluation with limited computation time In the following we include more detailed analysis of the scenario of a few supervised labels and limited computational resources. Besides LaplaceNet and its version without mixup, we include an ablation study of encoder pretraining as part of an autoencoder using binary cross-entropy as a reconstruction objective. In addition, we propose to improve the search for the optimal value of t by the model selection, since the gradient for t is usually noisy and small. Thus we include additional experiments where we chose the initial t based on the minimum training loss after 100 epochs of supervised training. The optimal t is approximated by calculating the training loss for 11 equally spaced values of t in the interval $[0.001, 1]$. The results are shown in Table 3. While mixup achieves no significant improvement in the few-label case trained using 100 epochs, we can see that a simple autoencoder pretraining consistently improves classifier accuracy. More notably, however, our proposed pretraining based on score matching achieves significantly better results than both random initialization and autoencoder pretraining. In the t-search, we observed that for all datasets, our proposed method selects $t = 0.9$, however it moves towards the interval $[0.4, 0.6]$ during training. While this shows that the approach of selecting t based on supervised training loss is not working, it demonstrates that the parameter t can very well be learned in the training process, making the downstream task performance robust to the initial value of t. In our experiments the final value of t was always in the range $[0.4, 0.6]$, independent of the initial value of t.

H. Training on single timescales

To understand the effect of training DRL on different timescales more clearly, we limit the support of the weighting function $\lambda(t)$ to a single value of t. We analyze the resulting quality of the latent representation for different values of t using the silhouette score with euclidean distance based on the dataset classes (Rousseeuw, 1987). It compares the average distance between a point to all other points in its cluster with the average distance to points in the nearest different cluster. Thus

Pretraining	Options	CIFAR-10 100 labels	CIFAR-100 1000 labels	MiniImageNet 4000 labels
None		64.12	45.14	47.18
None	mixup	54.06	46.28	47.64
DRL		**69.79**	**48.04**	**50.75**
DRL	t-search	67.07	47.08	50.31
Autoencoder		64.99	46.88	48.52

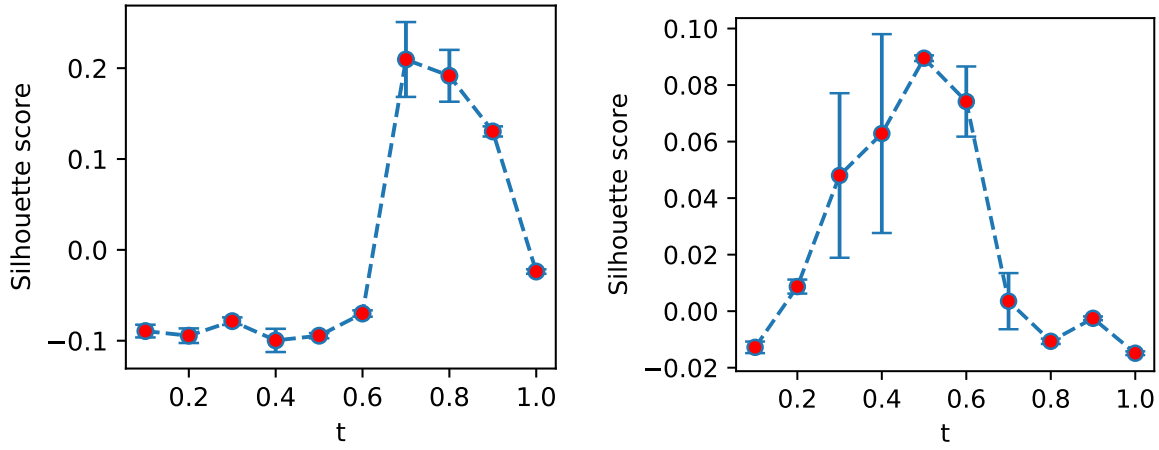
Table 3. Comparison of classifier accuracy in % for different pretraining methods in the case of few supervised labels when training for 100 epochs only.

Pretraining	CIFAR-10 100 labels	CIFAR-100 1000 labels	MiniImageNet 4000 labels
None	73.68	55.58	58.40
DRL	**74.31**	**55.85**	**58.95**
Autoencoder	58.84	55.41	57.93

Table 4. Classifier accuracy in % for autoencoder pretraining compared with the baseline and score matching as pretraining. No mixup is applied for this ablation study.

Pretraining Mixup in sup. training		Ours Basic No	Ours Basic Yes	Ours Mixup-DRL Yes	Ours VDRL No	Ours VDRL Yes
Dataset	#labels					
CIFAR-10	100	74.31	64.67	70.40	**81.63**	77.51
	500	92.70	92.31	92.55	**92.79**	91.46
	1000	93.24	93.42	93.14	**93.60**	93.33
	2000	**94.18**	93.91	93.80	93.96	94.27
	4000	94.75	**95.22**	94.75	95.00	94.87
CIFAR-100	1000	55.85	55.74	55.15	**56.47**	55.65
	4000	67.22	67.47	67.09	**67.54**	67.52
	10000	73.31	73.66	**74.36**	73.50	73.20
	20000	76.46	76.88	**77.04**	76.64	76.68
MiniImageNet	4000	58.95	59.29	**59.46**	59.14	59.36
	10000	67.31	66.63	67.31	**67.46**	66.79

Table 5. Evaluation of classifier accuracy in %, including the setting of using mixup during pretraining (right column). DRL pretraining is our proposed representation learning, and "Mixup-DRL" the respective version which additionally applies mixup during pretraining. "VDRL" instead uses a probabilistic encoder.



(a) MNIST (b) CIFAR-10

Figure 12. Mean and standard deviation of silhouette scores when training a DRL model on MNIST (left) and CIFAR-10 (right) using a single t over three runs.

t_{init}	$\sigma(t_{\text{init}})$	Gaussian FID ↓	Uniform + Gaussian FID ↓
0.5	0.71	218.95	25.02
0.6	1.66	75.11	5.15
0.7	3.88	12.57	2.98
0.8	9.10	3.05	2.99
0.9	21.33	2.97	2.94
1.0	50.00	3.01	2.99

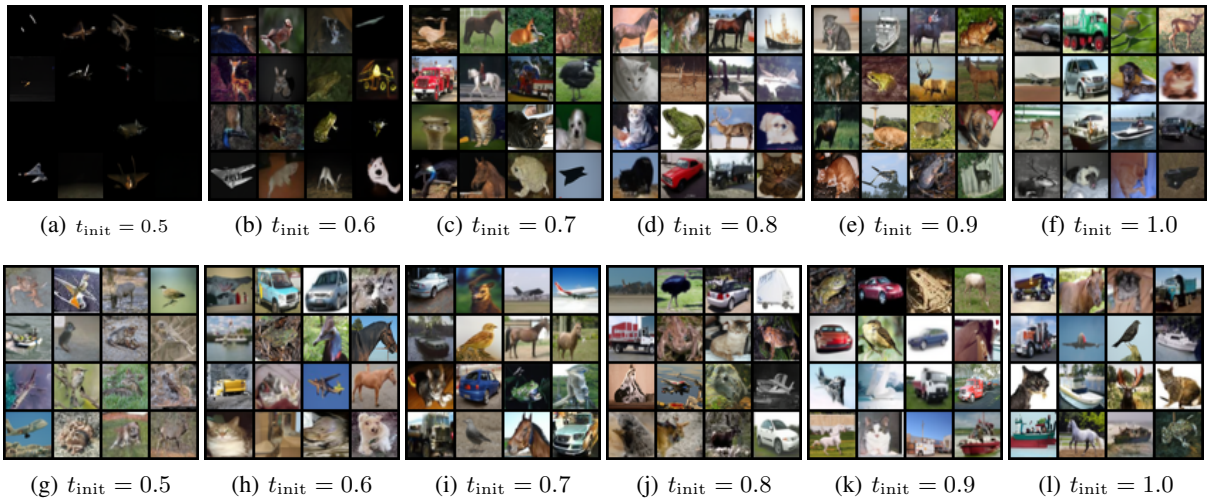
Table 6. FID for different initial noise scales evaluated on 20k generated samples.

we measure how well the latent representation encodes classes, ignoring any other features. Note that after learning the representation with a different distribution of t it is necessary to perform additional training with a uniform sampling of t and a frozen encoder to achieve good sample quality.

Figure 12 shows the silhouette scores of latent codes of MNIST and CIFAR-10 samples for different values of t. In alignment with our hypothesis of Section 2.1, training DRL on a small t and thus low noise levels leads to almost no encoded class information in the latent representation, while the opposite is the case for a range of t which differs between the two datasets. The decline in encoded class information for high values of t can be explained by the vanishing difference between distributions of perturbed samples when t gets large. This shows that the distinction among the code classes represented by the silhouette score is controlled by $\lambda(t)$.

I. The choice of the initial noise scale

In the following, we evaluate image quality and diversity for different initial noise scales for CIFAR-10 dataset. Note that we do not change $\sigma(T)$, but instead evaluate generated images for different initial times t_{init}, which implicitly define the initial noise scale $\sigma(t_{\text{init}})$. This reduces the number of sampling steps per image, which is $1000 \times t_{\text{init}}$ and thus directly proportional to t_{init}. Table 6 shows the FID of generated images for various values of t_{init}. As we can see, the first 200 sampling steps can safely be replaced by approximating the prior directly either with the Gaussian or the additional uniform distribution. Interestingly, using the sum of the uniform and Gaussian random variables as a prior leads to improved image quality. This approximation for $p_{0.7}(x)$ allows us to reduce the number of sampling steps by 30% without sacrificing image quality, which is further supported by the visual quality of generated samples shown in Figure 13. Further, note that FID is occasionally lower for values of $t_{\text{init}} < 1.0$ than for $t_{\text{init}} = 1$. This suggests that up to these timescales, our prior approximates the distribution better than the diffusion model when starting at $t_{\text{init}} = 1.0$.



(a) $t_{\text{init}} = 0.5$ (b) $t_{\text{init}} = 0.6$ (c) $t_{\text{init}} = 0.7$ (d) $t_{\text{init}} = 0.8$ (e) $t_{\text{init}} = 0.9$ (f) $t_{\text{init}} = 1.0$

(g) $t_{\text{init}} = 0.5$ (h) $t_{\text{init}} = 0.6$ (i) $t_{\text{init}} = 0.7$ (j) $t_{\text{init}} = 0.8$ (k) $t_{\text{init}} = 0.9$ (l) $t_{\text{init}} = 1.0$

Figure 13. Generated image samples for different values of t_{init}. Top row ((a)-(f)) uses the Gaussian prior, bottom row ((g)-(l)) uses the version with an additional uniform random variable in the prior.