Representation Learning with Diffusion Models

This Master thesis has been carried out by Jeremias Traub at the IWR, Heidelberg University under the supervision of Prof. Dr. Björn Ommer (Ludwig Maximilian University of Munich & IWR, Heidelberg University) and Robin Rombach (Ludwig Maximilian University of Munich & IWR, Heidelberg University) and Prof. Dr. Tilman Plehn (Institute for Theoretical Physics, Heidelberg University)
Abstract

Diffusion models (DMs) have achieved state-of-the-art results for image synthesis tasks as well as density estimation. Applied in the latent space of a powerful pretrained autoencoder (LDM), their immense computational requirements can be significantly reduced without sacrificing sampling quality. However, DMs and LDMs lack a semantically meaningful representation space as the diffusion process gradually destroys information in the latent variables. We introduce a framework for learning such representations with diffusion models (LRDM). To that end, a LDM is conditioned on the representation extracted from the clean image by a separate encoder. In particular, the DM and the representation encoder are trained jointly in order to learn rich representations specific to the generative denoising process. By introducing a tractable representation prior, we can efficiently sample from the representation distribution for unconditional image synthesis without training of any additional model. We demonstrate that i) competitive image generation results can be achieved with image-parameterized LDMs, ii) LRDMs are capable of learning semantically meaningful representations, allowing for faithful image reconstructions and semantic interpolations. Our implementation is available at https://github.com/jeremiastraub/diffusion.
1 Introduction

As humans, we can visually perceive the world around us, recognize objects, and imagine new scenes with great ease. The actual complexity of the task becomes apparent when trying to make a computer understand images. To computers, images are a big pile of numbers, spatially arranged on a pixel grid with color represented as tuple of 3 numbers in the RGB scheme. Enabling computational models to make sense of, and to solve various tasks on image data is the key challenge in the field of Computer Vision. It comprises many image-related tasks such as image classification, object detection and image segmentation but also that of image synthesis and representation learning. This thesis addresses the topics of visual synthesis (see Sec. 1.1) and representation learning (see Sec. 1.2). In particular, we tackle the problem of learning meaningful representations with diffusion models [1, 2] which have recently shown impressive visual synthesis results [3].

When we visually perceive something, this does not only involve a sensory process but also, and most importantly, the interpretation through the brain, which makes use of previously acquired concepts to make sense of the received signal [4]. Deep neural networks (DNNs) have proven to be a powerful tool for modeling the latter [5]. Due to their deep architecture, they are capable of capturing more abstract features, i.e., high-level interdependencies and patterns, when trained on a large amount of training images [6].

The thesis is structured as follows: Sec. 2 provides an overview of the relevant machine learning methods, particularly of variational autoencoders (Sec. 2.1) and diffusion models (Sec. 2.2). Sec. 3 discusses the connections of this work to related literature. The results for our experiments are presented in Sec. 4.
1.1 Visual Synthesis

The task of Visual Synthesis aims at understanding images by learning to create new, unseen images from the data distribution. The corresponding class of models is called deep generative models. More precisely, given a large amount of training images that follow a distribution $p(x)$, we want to be able to generate new images that could have been samples from the true data distribution. To that end, generative models either explicitly (e.g. variational autoencoders [7]) or implicitly (e.g. generative adversarial networks [8]) perform density estimation such that the model distribution $p_\theta(x)$ approximates the true distribution $p(x)$.

![Figure 1.1: Visual Synthesis with DALL-E 2 [9]. The model generates new images based on text captions. It is based on diffusion models conditioned on CLIP [10] embeddings to allow for guidance through the textual captions. Taken from [9].](image)

In this work, we employ diffusion models (DMs) [1, 2] for image synthesis. DMs have achieved state-of-the-art results in image synthesis [3] and density estimation [11], and have shown great flexibility for various conditional synthesis tasks [3, 12, 9]. For example, the DALL-E 2 model is based on DMs for synthesizing images based on a text caption (see Fig. 1.1).
1.2 Representation Learning

An underlying idea of representation learning is that real image data lies on a lower-dimensional manifold of the pixel-space [13], i.e., that the data distribution has a reduced effective dimensionality. In other words, we assume that there is a set of underlying generative factors that capture the semantics of an image [14] [15]. There are different approaches to learning such representations. A supervised classification model using a DNN learns to map images to a task-specific representation space in the deeper layers where regression can be employed to classify the image [16]. Contrastive learning is a self-supervised [17] or supervised [18] approach where a representation space is learnt through learning similarities and differences between images [10]. Often, however, representation learning is closely linked to generative modeling (i.e., visual synthesis). That way, the learnt representation is directly evaluated during training in terms of its capabilities for recovering an input image from its representation (through a reconstruction loss, as in VAEs [7]) or for creating new, plausible images (through an adversarial loss, as in GANs [8]). Generative models coupled with a meaningful representation space allow for controlled synthesis, such that interpolations in the representation space imply semantic interpolations in the corresponding generated images. Diffusion models [1, 2] are a type of generative model that, by design, lack a meaningful representation space. On the other hand, DMs are powerful generative models, partly due to their particular inductive bias for image-like data [12]. Hence, we seek to extract a just as powerful and rich representation – specific to the generative denoising process – which in turn allows for control over image semantics for synthesis tasks as well as for usage in downstream tasks such as classification. Unfortunately, DMs [1, 2], as such, do not provide a meaningful representation space. In contrast, the generative power lies in the learnt denoising steps in between the thousands of latent variables. In this work, we introduce a framework to equip diffusion models with a semantically structured representation space.
In summary, this work makes the following contributions:

(i) We revisit the image-parameterization for DMs, discuss differences to the commonly used noise-parameterization, and demonstrate that competitive FIDs can be achieved with image-parameterized LDMs.

(ii) We introduce a framework for learning semantically meaningful representations with diffusion models (LRDM). In contrast to previous work [19], we can efficiently sample from a tractable representation prior for unconditional image synthesis, without training an extra model.

(iii) Further, we extend our framework to timestep-conditional (t-LRDM) and class-conditional representation learning. Additionally, we propose a framework for separating style and shape information.
2 Methods

This chapter gives an overview of Variational Autoencoders (Sec. 2.1) and Diffusion Models (Sec. 2.2), which lay the foundation of this thesis.

2.1 Variational Autoencoders

The goal of generative machine learning approaches is to model the data distribution \( p(x) \). The Variational Autoencoder (VAE) \cite{kingma2013auto} does so by learning to reconstruct input images from a compressed latent code. An underlying idea of the model is that real world images can be represented by a relatively small set of higher-lever features, i.e., that the data lies on a lower-dimensional manifold in the image space. Expressed in the framework of probabilistic graphical models, it is assumed that the observed (i.i.d.) data distribution \( p(x) \) is generated by some random process from an unobserved random variable \( z \) (see Fig. 2.1):

\[
p(x) = \int p(x \mid z)p(z)dz  \tag{2.1}
\]

This expression, and thus also the posterior density

\[
p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}  \tag{2.2}
\]

are usually intractable without making strong assumptions about the likelihood function \( p(x \mid z) \), which we do not want to do here in order to train an expressive model. Instead of resorting to sampling-based inference methods (e.g. MCMC), we merely assume the prior \( p(z) = p_\theta(z) \) and the likelihood \( p(x \mid z) = p_\theta(x \mid z) \) to come from a parameterized families of distributions – allowing us to employ NNs as function approximators – and then apply the concept of variational inference.

We then perform maximum likelihood estimation to find the optimal parameters \( \theta \). This means that we seek those parameters \( \theta \) for which the
training data is most probable given the statistical model \( p_\theta(x) \), i.e., which maximize the (log-)likelihood of the data. For that, we first introduce a parameterized encoder \( q_\phi(z \mid x) \) – which, again, can be approximated by a NN – to have a tractable inference model at hand. We will see in the following that we can then avoid computing the intractable integral in Eq. 2.1 by optimizing a lower bound of the log-likelihood.

![Graphical model representation of a VAE. Shaded circles denote latent variables, unshaded circles denote observable variables. Solid lines represent the generative path, dashed lines represent the encoder path.](image)

We rewrite the log-likelihood of the generative model as follows:

\[
\log p_\theta(x) = \log \int p_\theta(x, z) \, dz
\]

(2.3)

\[
= \log \int q_\phi(z \mid x) \frac{p_\theta(x, z)}{q_\phi(z \mid x)} \, dz
\]

(2.4)

\[
\geq \int q_\phi(z \mid x) \log \frac{p_\theta(x, z)}{q_\phi(z \mid x)} \, dz
\]

(2.5)

\[
= \int q_\phi(z \mid x) \log \frac{p_\theta(z \mid x)p_\theta(x)}{q_\phi(z \mid x)} \, dz
\]

(2.6)

\[
= \int q_\phi(z \mid x) \log p_\theta(x) \, dz - \int q_\phi(z \mid x) \log \frac{q_\phi(z \mid x)}{p_\theta(z \mid x)} \, dz
\]

(2.7)

\[
(\ast\ast) \log p_\theta(x) - D_{KL}(q_\phi(z \mid x) \parallel p_\theta(z \mid x))
\]

(2.8)

(2.9)
At (*), Jensen’s inequality is used to get a lower bound estimate. The expression in Eq. 2.5 is called Evidence Lower Bound (ELBO). At (**), we identify the log-likelihood and the Kullback-Leibler divergence $D_{KL}$ between the approximated and the true posterior. The KL-divergence is a measure of distance between two probability distributions, it is non-negative and equals zero if and only if the two distributions match almost everywhere. From Eq. 2.8 we see that the ELBO becomes exact for the right choice of $q_\phi(z \mid x)$.

Since we want to employ NNs as function approximators with the parameters $\{\phi, \theta\}$, we reformulate the maximum likelihood estimation as an optimization problem that can be solved with gradient methods. Rearranging the terms in Eq. 2.5 differently yields

$$\text{ELBO} = \mathbb{E}_{q_\phi(z \mid x)} \left[ \log p_\theta(x \mid z) \right] - D_{KL}(q_\phi(z \mid x) \parallel p_\theta(z)) \quad (2.10)$$

Maximizing the ELBO thereby comprises two concurrent tasks: (1) Maximizing the reconstruction quality and (2) keeping the approximate posterior close to the latent prior. For the latter, the latent prior is typically assumed to be a standard Gaussian distribution $p(z) = \mathcal{N}(z; 0, I)$. For the former, the MSE between input and reconstruction is typically employed as reconstruction loss (assuming a Gaussian distribution again).

In order to be able to compute a low-variance estimate of the gradient $\nabla_{\phi, \theta}\text{ELBO}$, we further need to reparameterize the random variable $z$ as

$$q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x)^2 I) \quad (2.11)$$

That way, we can compute $z$ given an input $x$ as

$$z = \mu_\phi(x) + \sigma_\phi(x) \cdot \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}(\epsilon; 0, I) \quad (2.12)$$

Separating the random nature of $z$ from the deterministic encoder allows for propagating the gradient backwards through the whole model to jointly optimize $\phi$ and $\theta$.

---

1While the concept of reparameterization is a general one, we specifically choose to estimate the mean and variance of a gaussian distribution in order to match the functional form of the latent prior.
VAEs are generative models that readily provide a meaningful latent space. The structure of the data representation is enforced by the second term in Eq. 2.10 which regularizes the information encoded by $q_{\phi}(z \mid x)$ while the (first) reconstruction term encourages unique encodings. It is this trade-off that gives the learnt representation a semantic structure. In terms of an autoencoder-pipeline, the regularization term restricts the size of bottleneck from which the input image is reconstructed. In order to explicitly control the width of the bottleneck, [24] propose to introduce a regularization weighting factor $\lambda$, such that the minimization objective becomes

$$L_{\text{VAE}} = -\mathbb{E}_{q_{\phi}(z \mid x)} \left[ \log p_{\theta}(x \mid z) \right] + \lambda \cdot D_{\text{KL}}(q_{\phi}(z \mid x) \parallel p_{\theta}(z))$$  \hspace{1cm} (2.13)

For $\lambda \to 0$ the VAE approaches a deterministic autoencoder, whereas a larger $\lambda$ narrows down the bottleneck and therefore increases the disentanglement of the representation, i.e., it forces the encoder to restrict itself to encoding only the main latent factors in $z$.

While VAEs proved to be useful for representation learning in the image domain [14, 15], they tend to generate blurry samples [25] and thus, in the task of image synthesis, other generative model types (e.g. GANs, Diffusion Models) often outperform VAEs.
2.2 Diffusion Models

Diffusion Models (DMs) synthesize data by reversing a gradual noising process. A forward diffusion process successively adds small amounts of gaussian noise to the training data while the model learns to recover the destroyed information. Samples can then be generated by passing random noise through the learnt reverse process.

There are various design choices involved when setting up a DM – not all of which are addressed in the following sections. In particular, we only consider discrete diffusion processes. The derivations and notation roughly follow that of [2]. Sec. 2.2.4 extends the classic DM by employing it in the latent space of an autoencoder model. Further, Sec. 2.2.5 introduces a designated representation encoder.

2.2.1 Diffusion Process

A DM can be described as latent variable model where the latent variables form a Markov chain, i.e., each variable $x_t$ only depends on the previous one $x_{t-1}$ (see Fig. 2.2). Under the Markov assumption, we can write the approximate posterior as follows:

$$q(x_{1:T} \mid x_0) := \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$  \hspace{1cm} (2.14)
The forward process transition distributions are assumed to be gaussian

\[ q(x_t \mid x_{t-1}) := \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I) \] (2.15)

with a variance schedule \( \{\beta_t\}_{t=1,...,T} \). In this work, the schedule is fixed to a linear noise schedule \( \beta_t = \frac{T-t}{T-1}\beta_1 + \frac{t-1}{T-1}\beta_T \), with \( t = 1, ..., T \) and \( T = 1000 \). However, other schedules such as the cosine-schedule can be chosen [26], or \( \beta_t \) may also be learned by the model [11]. The scaling factor \( \sqrt{1 - \beta_t} \) of the mean ensures that the variance is bounded, such that for large enough \( T \), \( x_T \) follows a gaussian distribution with unit variance.

The pre-defined forward diffusion process \( q(x_{1:T} \mid x_0) \) gradually transforms the data distribution \( q(x_0) \) to a standard gaussian distribution. To synthesize new images, we are interested in learning the reverse process

\[ p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t) \] (2.16)

The reverse process transition distributions are usually assumed to be gaussian

\[ p_\theta(x_{t-1} \mid x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \] (2.17)

This is justified by the fact that the true reverse process transitions \( q(x_{t-1} \mid x_t) \) are of the same functional form for small enough perturbations \( \beta_t \) [2] [27]. We will see in Sec. 2.2.2 that the training objective can then be reduced to KL-Divergences between gaussian distributions, which have a closed form. For that, let us first make some additional observations about the diffusion process that help making the training objective tractable.

With \( \alpha_t := 1 - \beta_t \) and \( \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s \), it follows from Eq. 2.15:

\[ q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)I) \] (2.18)

By reparameterizing the gaussian distribution we can directly sample \( x_t \) at any timestep \( t \) via

\[ x_t(x_0, \epsilon) = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}(\epsilon; 0, I) \] (2.19)
(The linear noise schedule $\beta_t$ and the relevant parameters for the diffusion process are visualized in Sec. A.2.) This means that the model can be trained by looking at each transition separately and not having to sample along the forward or reverse Markov chain during training.

Moreover, the forward process posteriors are tractable when conditioned on $x_0$:

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \beta_t I) \quad (2.20)$$

where $\tilde{\mu}_t(x_t, x_0) := \sqrt{\alpha_{t-1}} \beta_t x_0 + \frac{\sqrt{\alpha_t(1 - \tilde{\alpha}_{t-1})}}{1 - \tilde{\alpha}_t} x_t$

and $\tilde{\beta}_t := \frac{1 - \tilde{\alpha}_{t-1}}{1 - \tilde{\alpha}_t} \beta_t \quad (2.21)$

### 2.2.2 Training Objective

DMs are trained to minimize the negative log-likelihood by maximizing the ELBO (cf. Sec. 2.1).

$$E[-\log p_\theta(x_0)] \leq E_q \left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right] =: L_{vlb} \quad (2.22)$$

The upper bound $L$ can be rewritten in terms of tractable KL-Divergences (see Sec. A.3 for a derivation):

$$L_{vlb} = E_q \left[ \frac{D_{KL}(q(x_T \mid x_0) \parallel p(x_T))}{L_T} - \log p_\theta(x_0 \mid x_1) \right] + \sum_{t>1} \frac{D_{KL}(q(x_{t-1} \mid x_t, x_0) \parallel p_\theta(x_{t-1} \mid x_t))}{L_{t-1}} \quad (2.23)$$

With the forward process being fixed, $L_T$ is a constant w.r.t. the optimization of the model parameters $\theta$. Moreover, for large enough $T$, the true distribution $q(x_T \mid x_0)$ is very close to the assumed prior distribution $p(x_T) = \mathcal{N}(x_T; 0, I)$ and thus $L_T$ very close to zero. For discrete image data $x_0$, discrete log-likelihoods are required. [2] derive a discrete decoder from
\[ N(x_0; \mu_\theta(x_1, 1), \sigma_1^2 I) \] in order to compute \( L_0 \). Minimizing the \( L_{t-1} \) terms ensures that in each step the model recovers the corresponding lost information. Note that since we can sample \( x_t \) directly for any timestep \( t \) we can sample \( t \) uniformly during training.

From an alternative derivation (see Sec. A.3), it follows that minimizing Eq. 2.23 is equivalent to approximating the unconditional true reverse transitions \( q(x_{t-1}|x_t) \). This is intuitive when considering that \( p_\theta(x_{t-1} | x_t) \) does not have access to the information that is destroyed by the diffusion process up to the timestep \( t \).

### 2.2.3 Parameterizing the Reverse Process

This section discusses different possible parameterizations of the reverse process \( p_\theta(x_{t-1} | x_t) \). First, we fix the reverse process variance to be the same as the forward process variance \( \Sigma_\theta(x_t, t) = \sigma_t^2 I = \beta_t I \). \( \Sigma_\theta(x_t, t) \) could also be trained, as explored in [3]. The most straightforward parameterization is to let the model predict \( \mu_\theta(x_t, t) \). The loss term \( L_{t-1} \) in Eq. 2.23 then becomes

\[
L_{t-1} = \mathbb{E}_q \left[ \frac{1}{2\sigma_t^2} \| \tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t) \|^2 \right] + C \tag{2.24}
\]

with \( C \) being a constant w.r.t. \( \theta \). Another parameterization option is to let the model directly predict the denoised image \( x_{0\theta}(x_t, t) \). Rewriting Eq. 2.24 with Eq. 2.21 we get the following expression:

\[
L_{t-1} - C = \mathbb{E}_q \left[ \frac{\bar{\alpha}_t \beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)^2} \| x_0 - x_{0\theta}(x_t, t) \|^2 \right] \tag{2.25}
\]

Further, using the reparameterization in Eq. 2.19

\[
\tilde{\mu}_t(x_t, x_0) = \mu_t(x_t, \frac{1}{\sqrt{\alpha_t}}(x_t - \sqrt{1 - \alpha_t} \epsilon)) \]
\[
= \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon \right) \tag{2.26}
\]

we can also rewrite \( L_{t-1} \) in terms of a noise predictor \( \epsilon_\theta(x_t, t) \):
\[ L_{t-1} - C = \mathbb{E}_{x_{0}, \epsilon} \left[ \frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)} \| \epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \|^2 \right] \] (2.27)

In this work, we compare the image-parameterization \( x_{0\theta}(x_t, t) \) and the noise-parameterization \( \epsilon_\theta(x_t, t) \) (see Sec. 4.1.1) and their roles for representation learning (see Sec. 2.2.5). Predicting the mean of the reverse transitions lead to unstable training and much worse results in all learning configurations.

The estimator function is realized through a neural network. A convolution-based U-Net architecture \cite{23} is very suitable for this task, since the predicted quantity has the same dimensionality as the input. Moreover, there is no need for a bottleneck (cf. Sec. 2.1) and the spatial inductive bias of the U-Net can be exploited. The timestep information is integrated via a sinusoidal positional embedding, as used in transformer models \cite{21}.

\cite{2} find that the sample quality improves by a large margin when dropping the prefactors in Eq. 2.27. They reweight the loss terms in \( L_{\text{vlb}} \) and propose an alternative minimization objective

\[ L^*_\text{noise} := \mathbb{E}_{t, x_0, \epsilon} \left[ \| \epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \|^2 \right] \] (2.28)

(The subscript denotes the predicted quantity and the superscript asterisk denotes the dropped prefactors compared to the variational lower bound formulation.) We want to consider yet another reweighted objective for the image-parameterization which we obtain by dropping the prefactors in Eq. 2.25

\[ L^*_\text{image} := \mathbb{E}_{t, x_0, \epsilon} \left[ \| x_0 - x_{0\theta}(\sqrt{\alpha_t}x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \|^2 \right] \] (2.29)

Note that minimizing \( L^*_\text{noise} \) or \( L^*_\text{image} \) no longer maximizes the ELBO. Instead, the objectives emphasize different parts of the reverse process. The reweighting of the loss terms is shown in Fig. 2.3. \( L^*_\text{noise} \) emphasizes intermediate timesteps whereas \( L^*_\text{image} \) puts a strong weight on higher \( t \).
After training we can generate samples by sequentially evaluating the learnt reverse process $p_\theta(x_{t-1} \mid x_t)$, starting from $x_T \sim \mathcal{N}(x_T; 0, I)$. In order to sample from the gaussian transition probability distributions, in each step we sample a noise vector $z_T \sim \mathcal{N}(z; 0, I)$. The sampling procedure can then be written for the noise-parameterization as

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$$  \hspace{1cm} (2.30)

and for the image-parameterization as

$$x_{t-1} = \frac{\sqrt{\alpha_t} \beta_t}{\sqrt{\alpha_t(1 - \bar{\alpha}_t)}} x_{0\theta}(x_t, t) + \frac{\sqrt{\alpha_t(1 - \bar{\alpha}_{t-1})}}{1 - \bar{\alpha}_t} x_t + \sigma_t z$$  \hspace{1cm} (2.31)

Alternative to the probabilistic sampling, [28] derive a sampling scheme (further denoted as DDIM-sampling) that deterministically maps $x_T$ onto $x_0$. While it technically corresponds to a different, non-Markovian forward process, it can be readily employed for any pretrained diffusion model.

$$x_{t-1} = \sqrt{\alpha_{t-1}} \left( \frac{x_t - \sqrt{1 - \alpha_t} \epsilon_\theta(x_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}} \epsilon_\theta(x_t, t)$$  \hspace{1cm} (2.32)

An analogous scheme (obtained by making use of Eq. 2.19) can be used for image-parameterized models.
2.2.4 Latent Diffusion Models

DMs have shown impressive results in terms of sample quality as well as mode coverage, i.e., the samples exhibit a great diversity [2] [3]. However, both training and sampling with DMs is computationally very demanding. They typically have \(\sim 1000\) latent variables that are of the same dimensionality as the input and sampling requires sequential evaluation of the Markov chain.

In order to leverage the expressiveness of DMs while reducing the model size, they can be trained in the latent space of an autoencoder [12] [29]. The main idea of the autoencoder is to reduce the dimensionality of the diffusion model while retaining as much semantic information as possible. This was motivated by the observation that DMs (and likelihood-based models in general) use much capacity on modeling imperceptible details of the data [30] [12] [2]. While reweighting the diffusion loss term (see Fig. 2.3) addresses this to some extent, employing the DM on the compressed data drastically reduces its computational demands and explicitly masks out low-level image details for the likelihood-based DM optimization.

The diffusion model learns the distribution of continuous latents \(z \sim q(z)\) which are obtained by encoding the images \(z = \mathcal{E}(x)\). The decoder recovers the image from the latent \(\tilde{x} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(x))\). We experiment with the autoencoders \(\{\mathcal{E}, \mathcal{D}\}\) provided by [12], which are trained with a perceptual and an adversarial loss. The autoencoder is then fixed for the DM training. In the following, the DM trained in the latent space, as well as the whole model setup, will be referred to as LDM (Latent Diffusion Model).

\footnote{Sampling methods such as DDIM-sampling allow for evaluating fewer steps than the model was trained on. While this reduces sampling time, it usually comes at the cost of lower sample quality.}
The training objective for the LDM is the same as that for the DM, i.e., all derivations in the previous sections still hold with $x$ replaced by $z$. During training, $z_t$ is obtained by a single pass of $x$ through the encoder $E$ and then applying Eq. 2.19. For image synthesis, $z$ is sampled from the latent DM and subsequently passed through the decoder $D$.

The core of the LDM is a time-conditional U-Net consisting of convolutional residual blocks and attention layers at certain depth-levels. Early experiments show that choosing an autoencoder with spatial compression by a factor of $f = 4$ and unchanged channel number yields the best results in terms of sampling and reconstruction quality. This can be explained by the strong spatial inductive bias of both the autoencoder and the LDM architecture which can then be leveraged the most. An input image of size $H \times W \times c$ is thus encoded to a latent of size $H/f \times W/f \times c$.

There is no need for the discrete decoder in the ELBO derivation since the latents are continuous. The $L_0$ term thus becomes an additional $L_{t-1}$-summand.
### 2.2.5 Representation-conditional Latent Diffusion

A striking difference between DMs and VAEs is that in the former the latent variables are designed s.t. information is sequentially destroyed, whereas in the latter the latent variable contains a dense representation of the data, allowing for faithful image reconstruction and interpolation (depending on the regularization towards the latent prior). In order to learn a meaningful data representation in the framework of diffusion models, we extend the LDM by a jointly trained representation encoder \( \mathcal{E}_r : z_0 \rightarrow r(z_0) \). The extracted code \( r(z_0) \) is then passed to the denoising U-Net as conditioning information.

![Graphical model representation of a LRDM](image)

Fig. 2.5 shows the graphical model representation for the LRDM (Latent Representation Diffusion Model). The joint distribution becomes

\[
p_\theta(z_{0:T}, r) = p_\theta(x_{0:T} \mid r)p(r)p(z_T) \prod_{t=1}^{T} p_\theta(z_{t-1} \mid z_t, r) \tag{2.33}
\]

and the extended forward process is now

\[
q_\phi(z_{1:T}, r \mid z_0) = q(z_{1:T} \mid z_0)q_\phi(r \mid z_0) \tag{2.34}
\]
with $\phi$ being the parameters of the representation encoder $E_r$. We set the latent prior to a gaussian distribution $p(r) = \mathcal{N}(r; 0, I)$ and parameterize $E_r$ accordingly through $q_\phi(r \mid z_0) = \mathcal{N}(r; \mu_\phi(z_0), \sigma_\phi(z_0)^2 I)$. As training objective, we propose the (representation-conditional) LDM loss combined with a regularization term controlled by a parameter $\lambda$.

$$L_{LRDM} := L_{image} + \lambda L_{prior}$$
$$= \mathbb{E}_{x_0, \epsilon, t} \left[ \| z_0 - z_{0\theta}(z_t, t, r(z_0)) \|^2 \right] + \lambda D_{KL}(q_\phi(r \mid z_0) \parallel p(r)) \tag{2.35}$$

The first term is the reweighted diffusion loss (for the image-parameterization) and the second term pushes the representation distribution towards the gaussian prior. $L_{LRDM}$ corresponds to a reweighted ELBO-derived objective (see Sec. A.4 for a derivation).

Note that maximizing the ELBO is no longer equivalent to approximating the unconditional true reverse transitions $q(z_{t-1} \mid z_t)$ because $p_\theta(z_{t-1} \mid z_t, r(z_0))$ can now receive information through $r(z_0)$ that is destroyed in $z_t$. That, in turn, also provides the incentive for learning a meaningful representation. The information encoded by $E_r$ can bridge the information gap between $z_t$ and $z_0$, which is to be predicted.

In order to increase the expressivity of the representation, one can consider a separate latent variable $r_t$ for each timestep (see Fig. 2.6). This can be realized by making the representation encoder $E_r$ timestep-conditional. We call this model t-LRDM in the following. The corresponding loss is then

$$L_{t-LRDM} := \mathbb{E}_{x_0, \epsilon, t} \left[ \| z_0 - z_{0\theta}(z_t, t, r_t(z_0, t)) \|^2 \right] + \lambda D_{KL}(q_\phi(r_t \mid z_0) \parallel p(r_t)) \tag{2.36}$$
Figure 2.6: Graphical model representation of a t-LRDM. Shaded circles denote latent variables, unshaded circles denote observable variables. Solid lines represent the generative path, dashed lines represent the encoding path.
3 Related Work

With Diffusion Models having received a lot of attention in recent research, several concurrent projects were published over the course of this thesis. In particular, these include (in chronological order) [29, 31, 19, 12, 32, 9]. Those and other related work are addressed in this chapter.

Variational Autoencoders

Our work brings VAEs [7, 25] and diffusion models [1, 2] closely together. VAEs are suitable for representation learning as the VAE objective enforces a meaningful structured latent space [14]. However, they can exhibit “posterior collapse” [33] which DMs do not [19]. Our approach to representation learning with diffusion models (LRDM) combines an adapted VAE objective with the DM objective. Typically, VAEs assume gaussian posteriors and gaussian priors over continuous latents. The VQ-VAE [33] uses discrete latents with vector-quantization as regularization. Hierarchical approaches can strongly increase the sampling quality [34].

Diffusion Models

Diffusion generative models [1, 2] have shown impressive results in multiple domains (especially on images [2, 3] and audio [35]) on both generative tasks [3] and density estimation [11]. Competitive sampling quality could be reached by introducing the reweighted training objective in [2], which also establishes a close connection to score-based models [36, 37]. For the latter, a continuous formulation based on SDEs is presented in [37]. DMs exhibit stable training and allow for likelihood evaluation, as opposed to GANs [8]. However, the computational requirements for DMs are high, and sampling is slow compared to GANs due to the sequential evaluation of the reverse process. Thus, multiple works proposed improvements to the sampling process [28, 39, 37]. Important architecture improvements were presented in [3], further pushing the limits of image synthesis. Other variants employ different noise schedules [26] or a reweighted learning objective [40] to improve sampling quality.
Latent Diffusion Models

Training and sampling with DMs requires a lot of computational resources, because all latent variables are of the same dimensionality as the input and hence model evaluations and gradient computations are costly for high-resolution images. By employing them in a compressed latent space of an autoencoder, they can be trained more efficiently, also on high-resolution images [12]. In [29], a diffusion model is employed in the latent space of a jointly trained NVAE. In [12], the difficulty of weighing reconstruction quality against learning the prior over the latent space is avoided by separating the training of the first-stage autoencoder and the latent DM. As first-stage model, a VQGAN [22] is pretrained.

Representation Learning with Diffusion Models

In DMs, information contained in the latent variables is successively destroyed by the diffusion process. The semantic information is then rather contained in the learnt reverse process as a whole, distributed over the different timesteps. This makes representation learning difficult for DMs, as opposed to other generative models, such as VAEs, which inherently provide a low-dimensional latent representation space.

[31] propose to condition the denoising U-Net on the output of a jointly trained representation encoder which has access to the denoised input image. Their work differs from ours in various respects: i) They employ a DM directly in image space, whereas the LRDM uses the representation-conditional DM to model the distribution of compressed latents. This allows us provide results for images of higher resolution. ii) In [31], the noise-parameterization is used. We find that the image-parameterization is favorable in the LRDM setup and show that it provides a closer connection to the unconditional VAE for representation learning. iii) They use a non-spatial representations that are passed to the denoising U-Net together with the positional timestep-conditioning. We use spatial representations and concatenation, respectively. [31] also introduces the timestep-conditional representation. [41] find that it contains information of different kind at different timesteps, i.e., training a timestep-conditional encoder provides a richer representation.
Diffusion autoencoders \cite{19} use a deterministic semantic encoder on which a jointly trained DDIM is conditioned. They are able to learn a meaningful and decodable non-spatial representation. However, in order to sample from the representation space, a separate DM has to be trained on the encoded latents. Our work focuses on regularizing the representation through a KL-penalty. For a strong enough regularization, we can directly sample from the latent gaussian prior without training an additional model. For Diffuse VAEs \cite{32}, a VAE and DM is trained separately. The DM thereby refines the VAE samples. DALL-E 2 \cite{9} uses CLIP \cite{10} embeddings on which a diffusion model is conditioned.
4 Experiments

In this chapter, the main results of this thesis are presented. In Sec. 4.1, the LDM and the role of the reverse process parameterization are analyzed. Sec. 4.2 shows results for the representation learning setup (LRDM), Sec. 4.2.1 further investigates the role of the regularization term, and Sec. 4.2.2 evaluates the t-LRDM. Finally, Sec. 4.2.3 presents results for experiments with additional conditioning information.

The implementation of the framework used in this thesis is based on [3] and [37], the architecture of the LDM is based on [3] (with the pretrained autoencoder from [12]). The code is available at GitHub.

4.1 Latent Diffusion Model

We begin by giving additional details on the model architecture and training, before discussing quantitative results in Sec. 4.1.1.

Because of the mild spatial compression rate \( f=4 \) of the first-stage model, the latents \( z \) are still image-like, thereby allowing us to exploit the spatial bias of the well-developed denoising U-Net architecture. Experiments confirm that an image-like dimensionality of the latents (with 3 channels) yields the best sampling quality for the used architecture.

We compare the two autoencoders provided by [12] in terms of sampling quality. Both are trained with a perceptual and an adversarial loss for high reconstruction quality, but one is (slightly) regularized by a KL-divergence towards a standard normal (referred to as KL-AE in the following) and the other is regularized by a vector quantization layer [33] within the decoder (referred to as VQ-AE in the following). The latter learns a discrete codebook of fixed size to represent the images as spatial collection of codebook entries [22]. In this case, the DM is trained on the pre-quantizations, i.e., on the continuous latents just before the quantization layer. For the KL-AE, the la-
tents are sampled from the reparameterized posterior \( E(x) = E_{\mu}(x) + E_{\sigma}(x)\epsilon \), with \( \epsilon \sim \mathcal{N}(0, 1) \).

For both autoencoders, we rescale the latents \( z \) to have unit variance. This ensures that the gaussian assumption of the latent prior \( p(z_T) \) is valid by decreasing the signal-to-noise ratio. The rescaling has significant effects for the KL-AE, while it has no noticeable effect for the VQ-AE [12]. In order to rescale the latents \( z \leftarrow z/\hat{\sigma} \), the standard deviation \( \sigma \) is estimated from the first 100 batches via Welford’s online algorithm.

In the denoising U-Net, Adaptive Group Normalization [3] is used to inject the timestep-embedding into the residual blocks. Upsampling and downsampling is done by the residual blocks. Similar to [29], we find that a dropout of around 0.2 for the LDM yields better results than without dropout. An overview of all hyperparameters for the different experiments can be found in Sec. A.1.

### 4.1.1 Reparameterizing the Reverse Process

This section quantitatively evaluates the LDM, in particular comparing \( L_{\text{noise}}^* \) and \( L_{\text{image}}^* \). We investigate differences between the parameterizations and demonstrate that the image-parameterized LDM can achieve competitive FID [42] scores.

As described in Sec. 2.2.3, the reverse process can be parameterized in different ways. Most frequently, the noise- parameterization is chosen – driven by the success of the \( L_{\text{noise}}^* \) objective [2], where the \( \epsilon \)-MSE for each noise scale is weighted equally. Relative to the ELBO-derived objective, the loss terms for low \( t \) are given very little weight and intermediate noise scales are weighted the most. Previous experiments have shown that the bits allocated at low \( t \) correspond to imperceptible distortions [2, 12, 40]. Reweighting the loss terms allows for introducing an inductive bias towards high perceptual sample quality. However, it is not clear which reweighting factors maximize the perceptual quality. In [40], this question is approached by looking at the signal-to-noise ratio which reduces during the diffusion process, depending on the noise schedule. They identify a so-called content-range of intermediate SNR-values \( (10^{-2} - 10^0) \) and show that putting a strong weight on the corresponding noise scales consistently improves the sample quality in terms
of FID [32]. We want to note that the empirically estimated content-range is not universal and might not be the same when training DMs not on the image data directly but on a latent representation. Moreover, our results indicate that the optimal weighting scheme also depends on the underlying learning task. In particular, we obtain the best sample quality for the image-parameterization with a weighting scheme that strongly differs from $L^\star_{\text{noise}}$ and the one in [40].

**Structural differences between the image and noise parameterization**

There is little research on the different reparameterizations. [2] show results for the $\mu_\theta(x_t, t)$-parameterization (with $L_{\text{vib}}$) and report that training is instable when dropping the loss-prefactors. We observe instable training with this parameterization for both weighting schemes. Furthermore, [2] discard the image-parameterization mentioning worse sampling quality. Other than that, the image-parameterization is used in [27], although they formulate a very different, adversarial learning objective.

In the following, we want to discuss fundamental differences in the learning task for the different parameterizations. First, we compare the pixel-wise distribution of the actual NN output. For $\mu_\theta(x_t, t)$, the NN directly performs the small denoising step, which means that its output is close to a standard gaussian for $t \to T$ and ideally matches the input image for $t \to 0$. $\epsilon_\theta(x_t, t)$ and $z_{0\theta}(x_t, t)$, on the other hand, have a fixed target during a sampling process – with the former always predicting gaussian noise and the latter always predicting the input image$^1$. For the noise-parameterization, 2.19 shows that the learnt mapping for $t \to T$ is close to the identity transform. For $t \to 0$, this is less and less so, but also the information gap between $z_t$ and $z_0$ decreases. For the image-parameterization, this is the other way round in the sense that the mapping approaches the identity transform for $t \to 0$, and the prediction task for $t \to T$ is particularly difficult.

$^1$For another perspective, let us briefly consider the whole generative process as a single neural network with $x_{1:T-1}$ being hidden layers and shared parameters across layers. Then, $\epsilon_\theta(x_t, t)$ and $z_{0\theta}(x_t, t)$ can be viewed as ResNet-like NNs (compare Eq. 2.30 and Eq. 2.31), as opposed to $\mu_\theta(x_t, t)$. This might give an intuition for why training the latter is more difficult. Note that this comparison somewhat falls short since the model is not really trained “end-to-end”.
Image-parameterized LDMs achieve competitive sampling quality

Can the same sample quality be achieved with the image-parameterization as with the widely used noise-parameterization? Tab. 4.1 shows results on the sampling quality for various models trained on the LSUN-Churches [43] (256 × 256) dataset. The reported FID and IS values were evaluated with 50k samples (using all 1000 sampling steps) against the training set, using the Python package provided by [44]. The LDM achieves competitive results after a relatively short training time for both $L_{\text{noise}}^*$ and $L_{\text{image}}^*$. The FID scores for the image-parameterization are consistently slightly higher than for the noise-parameterization but they don’t fall behind by far. Unconditional samples are displayed in Fig. 4.2 and Fig. 4.3.

Efficient training in latent space

LDMs can be trained efficiently on high-dimensional image data. After only 100 epochs, which corresponds to 47k gradient updates with a batch-size of 256, an FID better than in [2] is reached. In terms of V100 training days, this corresponds to 4 vs 50 in [2]². For comparison, LDM-8* was trained for around 400 epochs. Note that the FID scores were evaluated after a fixed number of training epochs (instead of reporting the lowest overall FID, as done frequently). The training time was constricted to avoid excessive usage of computational resources while being able to perform ablation studies. However, the FID scores consistently decreased over training time, and we expect further improvement for longer training (The image-parameterized LDM (VQ-AE) reaches an FID of 5.64 and an IS of 2.62 after 400 training epochs). Evaluating different dropout probabilities for the latent DM shows that using dropout is beneficial when training in latent space. We fix the dropout to 0.2 for further LDM experiments.

²Our models were trained on a single NVIDIA A100 GPU and we assume a speedup of ×2.2 for A100 vs V100 [12]. [2] assume a speedup of ×8 for their TPU v3-8 vs V100.
Next, we take a closer look at what the different estimators learn. To that end, we compute the distortion (RMSE) curves both in the image space ($x_0$-distortion) and in the latent space of the first-stage model ($z_0$-distortion), i.e., $\sqrt{\|x_0 - \hat{x}_0\|^2}$ and $\sqrt{\|z_0 - \hat{z}_0\|^2}$, respectively. For the image-parameterization, $\hat{z}_0$ is directly given by the NN output. For the noise-parameterization, it can be obtained from the predicted noise via Eq. 2.19. The distortion curves for the different LDMs are displayed in Fig. 4.1. When comparing the the parameterizations, we see that for a large timestep range, the $z_0$-distortions are very similar (see Fig. 4.1b). Visualizing the ratio (see Fig. 4.1c and Fig. 4.1d) shows that the $z_0$-predictions are more accurate for the image-parameterized model at high $t$ but significantly less accurate at very low $t$. We attribute the former to the strong weighting that $L^*_{image}$ puts on large $t$ and the latter to the difficulty of predicting crisp images. However, we also see that this accuracy gap at low $t$ vanishes in the image space, i.e., after passing the latent through the decoder of the first-stage model. Hence, the first-stage model (VQ-AE and KL-AE) can compensate for the extra noise in $\hat{z}_0$.

**Image-parameterized LDMs harmonize well with VQ-AEs**

Comparing the two different first-stage models, we observe, as in [12], that the sample quality is better for the VQ-AE than for the KL-AE (see Tab. 4.1) even though the reconstruction quality (without latent DM) is slightly better for the latter [12]. While that difference is small for the noise-parameterized model, it is significant for the image-parameterized model, improving the FID by more than 4 points (after 200 epochs). In Fig. 4.1b, we observe a striking difference between the $z_0$-distortion curves for the VQ-AE and the KL-AE, indicating structural differences between the latent spaces of the two first-stage models. Recall that in the decoder of the VQ-AE, the latents are first quantized via a discrete codebook, i.e., each latent vector (one for each pixel in the latent image representation) is mapped to its closest codebook entry. This might induce a certain invariance to $z_0$-distortions that harmonizes well with the image-parameterization, as the $x_0$-distortion is again similar for the VQ-AE and KL-AE, and even slightly lower for the VQ-AE at low $t$ (see Fig. 4.1a). Overall, the distortion curves differ both between image...
and latent space and between the VQ-AE and KL-AE as first-stage model. This indicates that DM training configurations optimized for DMs that are employed in image space (e.g. noise schedule, loss-reweighting) need not be the most favorable for LDMs. For example, using the loss-weighting corresponding to $L_{\text{noise}}^*$ together with the image-parameterized model leads to a significant decrease in sample quality (around 6 FID points after 100 epochs).

| Model                    | param. dropout | FID ↓ | IS ↑ |
|--------------------------|----------------|-------|------|
| DDPM* [2]                | noise          | 7.89  | -    |
| LDM-8* (KL-AE) [12]      | noise          | 4.02  | -    |
| StyleGAN2* [45]          | -              | 3.86  | -    |

| Model                    | param. dropout | FID ↓ | IS ↑ |
|--------------------------|----------------|-------|------|
| LDM (KL-AE [12])         | noise          | 6.08  | 5.23 | 2.56 | 2.60 |
|                          | 0.1            | 6.78  | 5.70 | 2.55 | 2.62 |
|                          | image          | 0.2   | -    | 10.72 | -    | 2.41 |
| LDM (VQ-AE [12])         | noise          | 6.00  | 5.38 | 2.59 | 2.64 |
|                          | image          | 0.2   | 8.27 | 6.50 | 2.44 | 2.54 |

Table 4.1: Evaluation metrics for unconditional image synthesis. Noise-parameterized LDMs were trained with $L_{\text{noise}}^*$, image-parameterized models with $L_{\text{image}}^*$. Our models were evaluated after 100 and 200 epochs. For models marked with *, the scores are presented as given in the respective papers. The image-parameterized LDM (VQ-AE) achieves competitive FID scores after rather short training times.
Figure 4.1: Analyzing the distortion (RMSE) at different timesteps in image space and in the latent space of the respective first-stage model. We observe structural differences (i) between the DMs trained on the VQ-AE latents and on those trained on KL-AE latents, and (ii) between the distortion curves in image space and latent space. See text for further discussion.
Figure 4.2: Unconditional samples from the noise-parameterized LDM (KL-AE) on the LSUN-Churches dataset.
Figure 4.3: Unconditional samples from the image-parameterized LDM (VQ-AE) on the LSUN-Churches dataset.
4.2 Representation Learning

We introduce LRDMs by extending LDMs by an additional encoding path which is regularized by a KL-penalty towards a gaussian prior, similar to a VAE. Experiments show that the LRDM is able to learn a meaningful structured representation that allows for faithful reconstructions and semantic interpolations. Before presentation quantitative results in Sec. 4.2.1 and further extensions in Sec. 4.2.2 and Sec. 4.2.3, we give an overview of architectural choices and investigate the properties of the representation learning qualitatively.

Figure 4.4: Overview of the LRDM architecture. The LDM is extended by a representation encoder $E_r$, which extracts a spatial representation directly from the denoised latent $z_0$. The denoising U-Net is then conditioned on the representation via concatenation on multiple spatial scales. To that end, the representation is scaled up through $D_r$. Adapted from [12].
Fig. 4.4 depicts the architecture of the LRDM. The representation encoder \( \mathcal{E}_r \) shares the same base architecture as the contracting path of the diffusion U-Net. The encoded representation \( \mathbf{r} \) is then injected into the contracting path of the diffusion U-Net by concatenation. For better guidance, a conditioning signal is provided on all depth-levels. For that, an intermediate module \( \mathcal{D}_r \) with the inverse encoder base architecture, computes a conditioning signal for all spatial resolutions of the diffusion U-Net. The concatenation is done only before the first residual block of each depth-level. We use representations with spatial dimensions to introduce some spatial bias. In particular, we compare representations of shape \( 64 \times 8 \times 8 \) (\( C \times W \times W \)) and \( 16 \times 16 \times 16 \).

**A connection between VAEs and LRDMs**

The LRDM setup can be interpreted in two different ways: i) As conditional LDM: The diffusion U-Net is now conditioned on the representation signal \( \mathbf{r}(z_0) \) which is directly informed by the denoised image. ii) As stack of conditional VAEs: With \( \mathcal{E}_r \) as encoder and \( z_{0\theta} \) as decoder (absorbing \( \mathcal{D}_r \) in the decoder), we have a stack of \( T \) VAEs with shared parameters that are conditioned on different signals \( z_t \). This becomes clear from Eq. 2.35 which matches a conditional VAE objective with a MSE reconstruction loss. At different timesteps, the conditioning \( z_t \) contains a different amount of information about \( z_0 \). For \( t = T \) (and large enough \( T \)), \( z_T \) is independent of \( z_0 \) and thus the objective becomes that of an unconditional VAE. Therefore, the learning of a comprehensive representation is particularly enforced at \( t \to T \).

In [31], the same argument is given, but for the noise-parameterized DM. However, we argue that the last statement only holds for image-parameterized diffusion models. For the noise-parameterized model at \( t = T \), the conditioning \( z_t \) is independent of \( z_0 \), but it also matches the prediction target, since \( z_T \approx \epsilon \). Thus, the unconditional VAE objective is not truly reached. We hypothesize that this may affect representation learning in the way that capturing the global semantics (for high \( t \)) is not so much enforced, compared to the image-parameterization. In early experiments, we observe better results in terms of image reconstruction and interpolation for the latter. In the following, only image-parameterized LRDMs are used.
Faithful reconstructions and semantic interpolations

Next, we want to qualitatively assess the properties of representation learning with LRDMs. We train LRDMs with a representation of size $64 \times 8 \times 8$ on LSUN-Churches [43] and CelebA-HQ [46] data. We then reconstruct input images from the encoded representation. We choose to take the mode of the approximate posterior $r = \mu_\phi(z_0)$ and use the deterministic DDIM sampling scheme. Fig. 4.5 and Fig. 4.6 show reconstructions for re-sampled $z_T$. The representation $r$ captures high-level semantics and global features, changing the latent code $z_T$ affects local details like windows, mouth, eyes, or local texture and color.

The aim of regularizing the learnt $q_\phi(r \mid z_0)$ towards a gaussian prior is to obtain a semantically structured representation space. We evaluate this property qualitatively by interpolating in representation space. For that we encode two input images into $r$ and $z_T$ via $r = \mu_\phi(z_0)$ and Eq. 2.19 respectively. Both are then interpolated via Slerp (Spherical linear interpolation) [28]. Finally, we decode the interpolated $(r, z_T)$ pairs using DDIM sampling. Interpolation results are shown in Fig. 4.7 and Fig. 4.8. The LRDM produces smooth interpolations, gradually transforming semantic features from start to end point. In Fig. 4.7 a comparison to interpolations directly in the latent space of the first-stage model is shown. There, the latents are not semantically structured and the interpolations are merely a blending of the two input images. Note that the first-stage model is explicitly designed to not perform semantic compression, we include this comparison rather as additional motivation for why a learnt semantic representation is desirable. We further compare to DDIM interpolations with a LDM (VQ-AE) trained for 400 epochs. To that end, we evaluate the (deterministic) forward DDIM process (inverse process to Eq. 2.32) to obtain $z_T$s. We then evaluate the DDIM sampling process for the interpolated $z_T$s. The DDIM produces less smooth transitions than the LRDM. Comparing the CelebA-HQ interpolations (Fig. 4.8) to those in [19], the LRDM also achieves smooth changes of head pose and facial attributes.

3In Fig. A.3 and Fig. A.2 we show reconstructions where $r$ is re-sampled from the approximate posterior and the probabilistic sampling scheme is used.

4See Fig. A.4 and Fig. A.5 for interpolations with probabilistic sampling and random $z_T$ for each sample.
Sampling progression of LDMs and LRDMs

How is the representation used for image synthesis? To address this question, we display exemplary \( \hat{x}_t \) and \( \hat{x}_0 \) progressions for the sampling process in Fig. 4.9. Note that the latter can be interpreted as lossy decoding for \( t > 0 \). For the LDM, the \( \hat{x}_0 \)-estimate gradually changes from close to the dataset mean towards the final sample. For the LRDM, the \( \hat{x}_0 \)-estimate is close to the final sample in terms of the overall shape and course features, from the first sampling step on. \( \hat{x}_0 \) then stays almost the same for a certain timestep range, before local features and details emerge eventually. The results suggest that for a certain timestep range the \( \hat{x}_0 \)-prediction is only based on the representation, and for lower \( t \) the sampling process is more similar to that of the LDM.

The above observations indicate that the noise or sampling schedule could be adapted and the number of steps reduced while preserving sample quality. In [19], the authors can confirm this for their model setup. With the additional information from the denoised image, \( p(x_{t-1} \mid x_t, r) \) may be gaussian when \( p(x_{t-1} \mid x_t) \) was not (e.g. due to an increased step size). However, we leave investigations in this direction for future work.
Figure 4.5: LRDM reconstructions from the encoded representation (mode) for randomly sampled \( z_T \), using DDIM sampling. The representation is of shape \( 64 \times 8 \times 8 \), \( \lambda = 10^{-4} \), image-parameterized LRDM (VQ-AE) trained on LSUN-Churches. The representation \( r \) captures high-level semantics and global features, changing the latent code \( z_T \) affects local details such as the shape of windows or local texture.
Figure 4.6: LRDM reconstructions from the encoded representation (mode) for randomly sampled $z_T$, using DDIM sampling. The representation is of shape $64 \times 8 \times 8$, $\lambda = 10^{-4}$, image-parameterized LRDM (VQ-AE) trained on CelebA-HQ. The representation $r$ captures high-level semantics and global features, changing the latent code $z_T$ affects local details such as the eyes or mouth shape.
Figure 4.7: Slerp interpolation results. Top rows: Interpolating between two input images in the (pre-quantization) latent space of the VQ-AE. Middle rows: Interpolating $z_T$ using the DDIM forward and reverse process (LDM with VQ-AE, trained for 400 epochs). Bottom rows: Interpolating between two input images in the representation space of a LRDM (trained for 100 epochs, representation shape: $64 \times 8 \times 8$, VQ-AE, $\lambda = 10^{-4}$, DDIM sampling, LSUN-Churches). The LRDM achieves smooth semantic interpolations.
Figure 4.8: Slerp interpolations between two input images in the representation space of a LRDM (representation shape: $64 \times 8 \times 8$, VQ-AE, $\lambda = 10^{-4}$, DDIM sampling, CelebA-HQ). The LRDM achieves smooth transformations of head pose and facial attributes.
Figure 4.9: Sampling progressions, $\hat{x}_t$ (above) and $\hat{x}_0$ (below) over time from left ($t = T$) to right ($t = 0$). Top rows: LRDM sampling. Bottom rows: LDM sampling. Center row: Mean of the LSUN-Churches training dataset. For the LDM, $\hat{x}_0$ gradually changes from close to the dataset mean towards the final sample. The progression for the LRDM exhibits an almost constant $\hat{x}_0$ over a certain initial timestep range, which already contains the overall shape and course features. This matches with the plateau observed in Fig. 4.10.
4.2.1 Combining Latent VAE and Latent Diffusion

This section quantitatively analyzes the properties of the learnt representation space and the performance in the task of image synthesis, dependent on the regularization strength $\lambda$. Increasing $\lambda$ in Eq. 2.35 reduces the information throughput in the representation bottleneck as the approximate posterior is pushed towards the gaussian prior. Thus, for high $\lambda$ the model should approach a LDM, leaving the representation encoder unused. Decreasing $\lambda$ imposes a stronger relative weight on $L^*_{\text{image}}$ encouraging the model to learn a more powerful representation of the input data.

First, we look at the distortion of $z_0$ over time, see Fig. 4.10. For weaker regularization, the distortion curve develops a plateau above a certain timestep, where the distortion is reduced to an almost constant value compared to the LDM. The plateau matches the observations made in Fig. 4.9. For $\lambda = 0.1$, the distortion curve closely matches that of the LDM.

![Figure 4.10: $z_0$-distortion (RMSE) curve for the LRDM for varying regularization strength. The distortion plateau at high $t$ stretches for weaker regularization. The plateauing distortion matches the observations made in Fig. 4.9.](image)

Note that the LDM $z_0$-distortion curve in Fig. 4.10 does not match that in Fig. 4.1b because the LDM was trained twice as long for the latter.
Quantitative evaluation of LRDMs

For a quantitative evaluation, LRDMs (VQ-AE) are trained for different regularization strengths $\lambda$. Tab. 4.2 shows results for the sampling and reconstruction quality after 100 epochs. The statistics in Tab. 4.2 are visualized in Fig. 4.11. For the unconditional sampling FID ($50k$), we sample $r$ and $z_T$ from the gaussian priors and then evaluate the 1000 reverse diffusion steps via probabilistic sampling (Eq. 2.31). All reconstruction metrics were evaluated on the validation set. For the reconstruction FID, we obtain the representation by encoding the input image and sampling from the posterior distribution $q_\phi(r \mid z_0)$. Here, too, we use the probabilistic sampling scheme. For the MSE $\langle \|x_0 - ˆx_0\|^2 \rangle$, RMSE $\sqrt{\langle \|x_0 - ˆx_0\|^2 \rangle}$, and the variance, $r$ is obtained by taking the mode of $q_\phi(r \mid z_0)$, and we use DDIM sampling for less stochasticity in the generative process. The metrics are then evaluated in image space, i.e., after passing the sampled latent through the VQ-AE decoder.

As baselines for the reconstruction quality, the scores for the pretrained VQ-AE as well as for a latent VAE (LVAE) are reported. For the latter, we trained a model with the LRDM architecture only on the last timestep $t = T$ such that the loss reduces to that of an unconditional VAE. Sampling is then done by a single pass through the model. In the image of DMs as stacked VAEs (see Sec. 4.2), this corresponds to training and evaluating only the top-most VAE of the stack.

The FID for unconditional sampling indeed reaches that of the LDM for stronger regularizations. Below a $\lambda$ of around $5e-4$, the approximate posterior is no longer close the gaussian prior and hence the FID grows rapidly. In order to sample from the representation space in this regime, one could train a separate model (e.g. diffusion model) there – as done in [19]. The recFID is improves that of the LVAE for lower $\lambda$. There, we see the benefits of both VAE and DM coming together, with the VAE-like reconstruction objective allowing for faithful reconstructions and the (reverse) diffusion process improving the perceptual image quality. Regarding the reconstruction RMSE and variance, we see that changing the regularization factor "interpolates between" the LDM and LVAE.

---

6For a fair comparison, we compare to the LDM (VQ-AE) evaluation after 100 epochs, see Tab. 4.1.
Furthermore, we compare different dimensionalities of the representation space. The results indicate a small improvement for the spatially more compressed representation, i.e., when there is less inductive bias towards spatial features. However, for the spatially more compressed representation, the diffusion U-Net also has an additional depth-level which might also be the relevant factor responsible for the improvement. Further architectural ablations could help investigating this question, see Sec. 5.

VAEs inherently entail a trade-off between reconstruction and sampling quality. We observe the same for the LRDM. For \( \lambda > 5 \times 10^{-4} \), where the sampling quality approaches that of the LDM, the reconstruction fidelity already starts to decrease rapidly. Samples and reconstructions for different regularization factors are shown in the Appendix in Fig. A.6 and Fig. A.7 respectively. In order to learn a comprehensive representation more efficiently, we experiment with a relaxed timestep-masking during training, i.e., we sample \( t \) uniformly from a timestep-range that successively increases from \([T - \epsilon, T]\) to \([0, T]\) over a certain initial training phase. However, we observe no improvements in terms of sampling and reconstruction quality.
| model      | $\lambda$ | FID  | recFID | MSE  | RMSE | Var  | Var* |
|------------|-----------|------|--------|------|------|------|------|
| LRDM (16 × 16 × 16) | $10^{-6}$ | 457.70 | 6.03   | 0.04 | 0.19 | 0.01 | 0.01 |
|            | $10^{-4}$ | 58.08 | 8.63   | 0.07 | 0.27 | 0.03 | 0.05 |
|            | $5 \cdot 10^{-4}$ | 10.54 | 13.30  | 0.17 | 0.41 | 0.08 | 0.11 |
|            | $10^{-3}$ | 10.45 | 14.62  | 0.23 | 0.48 | 0.11 | 0.14 |
|            | $10^{-1}$ | 8.39  | 12.90  | 0.23 | 0.48 | 0.11 | 0.15 |
| VQ-AE      | -         | -     | 3.75   | 0.01 | 0.12 | 0.00 | -    |
| LVAE       | $10^{-6}$ | 315.54 | 8.97   | 0.02 | 0.16 | 1e-6 | 3e-4 |
| LRDM (64 × 8 × 8) | $10^{-4}$ | 46.84 | 9.31   | 0.08 | 0.29 | 0.03 | 0.05 |
|            | $5 \cdot 10^{-4}$ | 8.89  | 11.75  | 0.18 | 0.42 | 0.08 | 0.11 |
|            | $10^{-3}$ | 8.04  | 12.29  | 0.23 | 0.48 | 0.11 | 0.13 |
|            | $10^{-1}$ | 7.65  | 12.30  | 0.54 | 0.73 | 0.28 | 0.29 |
| t-LRDM (16 × 16 × 16) | $10^{-4}$ | 106.90 | 8.84   | 0.10 | 0.32 | 0.05 | -    |
|            | $10^{-3}$ | 12.34 | 13.24  | 0.26 | 0.51 | 0.14 | -    |

Table 4.2: Sampling and reconstruction quality for various representation regularizations $\lambda$ for LSUN-Churches. All models but the VQ-AE are trained for 100 epochs. † denotes models trained with relaxed timestep-masking. Var* is calculated using probabilistic sampling. For a visual depiction, see Fig. 4.11.
(a) FID scores for unconditional sampling. For reference, the FID for the LDM is shown.

(b) Reconstruction FID scores evaluated on the validation dataset.

(c) Pixel-wise variance of the image reconstructions. For reference, the reconstruction variance of the LVAE and the sampling variance of the LDM are shown.

(d) Reconstruction RMSE (pixel-wise).

Figure 4.11: LRDM sampling and reconstruction quality over representation regularization $\lambda$ for different representation dimensionalities, on LSUN-Churches (values from Tab. 4.2). Changing $\lambda$ interpolates, to some extent, between the LDM (when narrowing down the representation bottleneck) and a latent VAE (towards low $\lambda$).
4.2.2 Timestep-conditional Representation

This section evaluates the t-LRDM, which extends the LRDM by introducing a latent variable $r_t$ for each timestep. This increases the dimensionality of the representation space by a factor of $T$ and the model should thus be able to learn a richer representation [31]. In particular, the encoded $r_t$ may contain the specific information that is needed to predict $z_0$ from $z_t$. E.g., for $t \to 0$, almost no extra is needed, and useful, to predict the denoised image. For the LRDM, this causes information to be pushed out of the representation space due to the KL-divergence loss, while for the t-LRDM other $r_t$ may be unaffected. The timestep-conditioning of the representation model is realized in the same way as in the denoising U-Net, i.e., the positional timestep-embedding is integrated into the residual blocks.

We train t-LRDMs with representation shape $16 \times 16 \times 16$ for two different regularization factors for 100 epochs. Note that while the training time is the same as for the LRDM, the reconstruction time increases since at each reverse process step, $r_t$ has to be computed anew. Results for the sampling and reconstruction quality can be found in Tab. 4.2. Fig. 4.12 shows the distortion curve (evaluated in latent space) and the KL-Divergence between the encoded distribution of $r_t(z_t, t)$ and the gaussian prior. Interestingly, the latter exhibits a sharp transition at a certain timestep below which the KL-Divergence is close to zero, and thus the corresponding latent variables $r_t$ basically unused, and above which the KL-Divergence is constant. The weaker the regularization the more $r_t$ encode information about $z_0$. The distortion curve has a similar form as that of the LRDM, but with the same sharp transition. Due to these observations, we argue that this transition separates the model in two stages: i) For high $t$, the denoised image is reconstructed from the encoded representation $r_t$, in a (conditional) VAE-like manner. ii) For lower $t$, the task becomes an actual image denoising task (from $z_t$) as for the LDM.
Compared to the LRDM, the FID scores and reconstruction metrics are slightly worse, though the timestep-conditional representation should be – in theory – at least as good as the non-$t$-conditional representation [31]. Possibly, this might be due to the relatively short training times of 100 epochs. Future investigations may also include ablations of the representation dimensionality. Moreover, the representation learning depends on the parameterization, the reweighting of the diffusion loss terms and the noise schedule – which are all partly interdependent, and worth exploring.

Figure 4.12: Results for the t-LRDM with representation shape $16 \times 16 \times 16$. Left: $z_0$-distortion (RMSE) over $t$. Right: KL-Divergence of the encoded representation distribution of $r_t$ and the gaussian prior. The KL-Divergence exhibits a sharp transition depending on $\lambda$. For low $t$, the representation is almost unused, for high $t$, the distortion plateau (see Fig. 4.10) sharpens.
4.2.3 Conditional Representation Learning

We perform additional experiments for explicit conditioning information. First, we present results for class-conditional representation learning on MNIST data. Then, preliminary results for an approach to shape-style disentanglement are shown.

Class-conditioning

For the class-conditional model, we additionally condition the denoising U-Net and the representation encoder on the class-label \( p_\theta(z_{t-1} \mid z_t, c, r(z_0, c)) \). The forward process is unchanged. We realize the class-conditioning by adding the class-label-embedding to the positional timestep-embedding \([3]\). Due to the small size of the MNIST images, the diffusion model is employed directly in the image space. Through the explicit conditioning we ideally remove the class-information from the learnt representation space, such that only the appearance of the digits is encoded. We visualize the learnt representation space by using PCA to identify the two dimensions of highest variance and then sampling from linearly spaced grid-points on a plane that is spanned by the first two principal components. I.e., first the grid-points in representation space are determined, then we sample from the model for each of them (with fixed class-label). As displayed in Fig. 4.13, appearance features such as linewidth and tilt gradually change along the grid axes. In such a representation space we can also smoothly interpolate between different appearances. Having separated the class and appearance information, we can change one while keeping the other fixed. In Fig. 4.13b, we obtain the \( r \) from the input images and then reconstruct from \( r \) conditioned on different class-labels. The representation used here has a significant spatial extent \((24 \times 7 \times 7)\) and thus provides strong spatial guidance during sampling. This could be modified by including less spatial bias, e.g. by using no spatial extent for the representation.
(a) Class-conditional samples on a linearly spaced grid lying on a plane that is spanned by the first two principal components of the representation space ($\lambda = 10^{-3}$).

(b) Class-appearance switching ($\lambda = 10^{-2}$). In rows: The representation (containing appearance information) extracted from the input image is decoded with different class-conditionings.

(c) Interpolations ($\lambda = 10^{-3}$)

Figure 4.13: Results for class-conditional representation learning with diffusion models on MNIST. Due to the explicit class-conditioning, the representation encodes the appearance, such as stroke width and tilt.
In another experiment, we seek to disentangle the style and the shape of images. For that, the representation encoder $E_r$ is replaced by two encoders $E_r^{\text{shape}}$ and $E_r^{\text{style}}$ (compare Fig. 4.4). The former is analogous to $E_r$ and directly receives the latent $z_0$, the latter receives $z_0$ with the shape information removed. That way, $E_r^{\text{style}}$ can only encode style information and $E_r^{\text{shape}}$ encodes the remaining information, i.e., the shape. How to remove the shape information from an image while preserving the style? Here, we use image warping via Thin Plate Spine (TPS) transformations where the image is deformed given some (random) control points – $E_r^{\text{style}}$ then receives $E(\text{warp}(x_0))$ (with $E$ being the pretrained first-stage model). We decide to impose a KL-penalty towards a gaussian prior only for the shape representation to ease the learning task and add skip-connections between $E_r^{\text{style}}$ and $D_r$ (the shape representation $r$ is concatenated at the middle block between the two modules). Note that unconditional sampling from the model is no longer possible in this setup.

Qualitative results for LSUN-Churches with a representation of shape $4 \times 32 \times 32$ are shown in Fig. 4.14. For the style-shape-switching, two input images are encoded – one by $E_r^{\text{style}}$ and the other by $E_r^{\text{shape}}$. The warp transformation is not applied for sampling, but only during training. If both images are the same, the samples are faithful reconstructions of the input. For the mixed samples, we observe that the shape is recovered very accurately, and the style is transferred mainly in terms of the color scheme. Local style attributes like wall texture is not transferred but encoded in the shape representation. We find that increasing the regularization and compressing the shape representation more spatially rapidly leads to a worse shape reconstruction. On the other hand, since we use a symmetric base architecture for $E_r^{\text{style}}$ and $E_r^{\text{shape}}$, $E_r^{\text{style}}$ also only performs spatial compression up to $32 \times 32$ – whereas less spatial bias might be favorable there. We see room for improvement in choosing the right inductive biases and fine-tuning the architecture as well as the regularization.
Figure 4.14: Style-Shape-switching on LSUN-Churches (representation shape $4 \times 32 \times 32$, $\lambda = 5 \cdot 10^{-6}$). The style-encoder receives warped input during training, thereby enforcing the shape information to be encoded in the KL-regularized bottleneck of the shape-encoder. For the style-shape-switches, each encoder receives a different input image. Then we sample from the diffusion model conditioned on the combined representation.
5 Conclusion & Limitations

Diffusion models have been able to push the limits of image synthesis through a structured generative process composed of sequential transitions between latent random variables. However, they lack a semantically meaningful and decodable representation as the encoding diffusion process is designed to gradually destroy information in the latent variables.

This work explores the topic of representation learning with diffusion models. To this end, we revisited the image-parameterization for diffusion models. We investigated differences to the commonly used noise-parameterization and demonstrated that competitive FID scores can be achieved when employing the diffusion model in the latent space of a pretrained autoencoder with VQ-regularization (LDM). While the sample quality slightly falls short of that of the corresponding noise-parameterized models, we think that this gap may be closed by further fine-tuning of the training setup – which has already been done specifically for the noise-parameterization. By employing the diffusion models in a compressed latent space of a pretrained autoencoder, the computational load compared to employing them directly in image space can be reduced. However, sampling from the model still remains rather slow compared to other generative models like GANs, since the reverse process has to be evaluated sequentially.

We then presented the LRDM for representation learning in the framework of diffusion models, finding that the image-parameterization is favorable for this setup and recovering the unconditional VAE objective as edge case \( (t = T) \) of the LRDM objective. By changing the regularization parameter, we can, to some extent, interpolate between a latent VAE and the LDM. We evaluated the timestep-conditional t-LRDM as extension of the LRDM. We believe that further fine-tuning of the representation learning can help in extracting a comprehensive and timestep-specific representation, in particular encoding image features of different size at the corresponding timesteps. Future investigations might include a timestep-specific regularization \( \lambda(t) \) or finding a stable setup for the \( \mu_\theta \)-parameterization which is biased in that
respect. Future work may include evaluating the learnt representations in terms of their capabilities in downstream tasks such as classification. We presented additional results for class-appearance and style-shape separation through conditional representation learning. While for the latter the spatial nature of the representation proved to be crucial, this bias might be limiting for learning global semantics. Thus, exploring other architectural choices and using non-spatial representations would be interesting to investigate in future work.

Overall, we introduced a framework for learning semantically meaningful representations with diffusion models and demonstrated that LRDMs are capable of efficiently learning such representations, which allow for faithful reconstructions and smooth interpolations.
Bibliography

[1] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, November 2015. Number: arXiv:1503.03585 arXiv:1503.03585 [cond-mat, q-bio, stat]. 2 3 4 12 21 63

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. arXiv:2006.11239 [cs, stat], December 2020. arXiv: 2006.11239. 2 3 4 10 11 12 13 14 16 21 25 26 27 29 63

[3] Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. arXiv:2105.05233 [cs, stat], May 2021. arXiv: 2105.05233. 2 3 13 16 17 21 24 25 49

[4] Dagmar Zeithamova, Michael L. Mack, Kurt Braunlich, Tyler Davis, Carol A. Seger, Marlieke T.R. van Kesteren, and Andreas Wutz. Brain Mechanisms of Concept Learning. The Journal of Neuroscience, 39(42):8259–8266, October 2019. 2

[5] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. Behavioral and Brain Sciences, 40:e253, 2017. 2

[6] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, May 2015. 2

[7] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, May 2014. Number: arXiv:1312.6114 arXiv:1312.6114 [cs, stat]. 3 4 6 21

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks, June 2014. arXiv:1406.2661 [cs, stat]. 3 4 21
[9] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents, April 2022. Number: arXiv:2204.06125 arXiv:2204.06125 [cs]. 3, 21, 23

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, February 2021. arXiv:2103.00020 [cs]. 3, 4, 23

[11] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational Diffusion Models. arXiv:2107.00630 [cs, stat], July 2021. arXiv: 2107.00630. 3, 11, 21

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs], December 2021. arXiv: 2112.10752. 3, 4, 16, 17, 21, 22, 24, 25, 27, 28, 29, 33

[13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives, April 2014. arXiv:1206.5538 [cs]. 4

[14] Xiao Liu, Pedro Sanchez, Spyridon Thermos, Alison Q. O’Neil, and Sotirios A. Tsaftaris. Learning Disentangled Representations in the Imaging Domain, April 2022. Number: arXiv:2108.12043 arXiv:2108.12043 [cs]. 4, 9, 21

[15] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. arXiv:1811.12359 [cs, stat], June 2019. arXiv: 1811.12359. 4, 9

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, May 2017. 4
[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations, June 2020. arXiv:2002.05709 [cs, stat].

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning, March 2021. arXiv:2004.11362 [cs, stat].

[19] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion Autoencoders: Toward a Meaningful and Decodable Representation. arXiv:2111.15640 [cs], December 2021. arXiv: 2111.15640.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, December 2015. Number: arXiv:1512.03385 arXiv:1512.03385 [cs].

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. arXiv:1706.03762 [cs], December 2017. arXiv: 1706.03762.

[22] Patrick Esser, Robin Rombach, and Björn Ommer. Taming Transformers for High-Resolution Image Synthesis. arXiv:2012.09841 [cs], December 2020. arXiv: 2012.09841.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, May 2015. Number: arXiv:1505.04597 arXiv:1505.04597 [cs].

[24] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. page 22, 2017.

[25] Diederik P. Kingma and Max Welling. An Introduction to Variational Autoencoders. Foundations and Trends® in Machine Learning, 12(4):307–392, 2019. arXiv:1906.02691 [cs, stat].
[26] Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. *arXiv:2102.09672 [cs, stat]*, February 2021. arXiv: 2102.09672.

[27] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the Generative Learning Trilemma with Denoising Diffusion GANs. *arXiv:2112.07804 [cs, stat]*, December 2021. arXiv: 2112.07804.

[28] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. *arXiv:2010.02502 [cs]*, October 2020. arXiv: 2010.02502.

[29] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based Generative Modeling in Latent Space. *arXiv:2106.05931 [cs, stat]*, June 2021. arXiv: 2106.05931.

[30] Sander Dieleman. Musings on typicality, 2020.

[31] Korbinian Abstreiter, Stefan Bauer, Bernhard Schölkopf, and Arash Mehrjou. Diffusion-Based Representation Learning. *arXiv:2105.14257 [cs]*, September 2021. arXiv: 2105.14257.

[32] Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. DiffuseVAE: Efficient, Controllable and High-Fidelity Generation from Low-Dimensional Latents. *arXiv:2201.00308 [cs]*, January 2022. arXiv: 2201.00308.

[33] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. *arXiv:1711.00937 [cs]*, May 2018. arXiv: 1711.00937.

[34] Arash Vahdat and Jan Kautz. NVAE: A Deep Hierarchical Variational Autoencoder, January 2021. arXiv:2007.03898 [cs, stat].

[35] Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic Music Generation with Diffusion Models. *arXiv:2103.16091 [cs, eess, stat]*, March 2021. arXiv: 2103.16091.

[36] Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. page 13, 2019.
[37] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. arXiv:2011.13456 [cs, stat], November 2020. arXiv: 2011.13456. 21, 24

[38] Pascal Vincent. A Connection Between Score Matching and Denoising Autoencoders. page 13, 2011. 21

[39] Alexia Jolicoeur-Martineau, Rémi Piché-Taillefer, Rémi Tachet des Combes, and Ioannis Mitliagkas. Adversarial score matching and improved sampling for image generation. arXiv:2009.05475 [cs, stat], October 2020. arXiv: 2009.05475. 21

[40] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception Prioritized Training of Diffusion Models. page 14, 2022. 21, 25, 26

[41] Sarthak Mittal, Guillaume Lajoie, Stefan Bauer, and Arash Mehrjou. From Points to Functions: Infinite-dimensional Representations in Diffusion Models. page 7, 2022. 22

[42] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, January 2018. Number: arXiv:1706.08500 arXiv:1706.08500 [cs, stat]. 25, 26

[43] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop, June 2016. Number: arXiv:1506.03365 arXiv:1506.03365 [cs]. 27, 35

[44] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. Version: 0.3.0, DOI: 10.5281/zenodo.4957738. 27

[45] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8107–8116, Seattle, WA, USA, June 2020. IEEE. 29
[46] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation, February 2018. arXiv:1710.10196 [cs, stat]. 35

[47] Dan C. Cireşan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, and Jürgen Schmidhuber. High-Performance Neural Networks for Visual Object Classification, February 2011. arXiv:1102.0183 [cs]. 19
A

A.1 Hyperparameters

For all diffusion model parameters, we use an EMA with a decay rate of 0.9999. No EMA is used for the representation encoder. Optimization is done using the Adam optimizer. An overview of the various model configurations used for the experiments is shown in Tab. A.1.

A.2 Linear Noise-Schedule

Figure A.1: Relevant factors for a linear β-schedule (see Sec. 2.2.1).
|                        | LDM | LRDM (16 × 16 × 16) | LRDM (64 × 8 × 8) | LRDM (16 × 16 × 16) | t-LRDM (16 × 16 × 16) | LRDM (MNIST) | Warp-LRDM (4 × 32 × 32) |
|------------------------|-----|---------------------|------------------|---------------------|-----------------------|---------------|-------------------------|
| **Diffusion Model Parameters** |     |                     |                  |                     |                       |               |                         |
| 1st-stage model        | VQ-AE-f4 | VQ-AE-f4 | VQ-AE-f4 | VQ-AE-f4 | - | VQ-AE-f4 |
| Diffusion steps        | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| Noise schedule         | linear | linear | linear | linear | linear | linear |
| Channels               | 128 | 128 | 128 | 128 | 32 | 128 |
| Channel Multiplier     | [1, 2, 4] | [1, 2, 4] | [1, 2, 2, 2] | [1, 2, 4] | [1, 2, 2] | [1, 4] |
| Residual Blocks per Layer | 2 | 2 | 3 | 2 | 2 | 2 |
| Attention resolutions  | [32, 16] | [32, 16, 8] | [32, 16] | [32, 16] | [14, 7] | [32] |
| Head Channels          | 64 | 64 | 64 | 64 | all | 64 |
| Dropout                | 0.2 | 0.2 | 0.2 | 0.2 | 0.1 | 0.2 |
| Class-conditional      | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **Training Parameters** |     |                     |                  |                     |                       |               |                         |
| Batch Size             | 256 | 256 | 256 | 256 | 256 | 256 |
| Learning Rate          | 1e-4 | 1e-4 | 1e-4 | 1e-4 | 1e-4 | 1e-4 |
| $N_{\text{params, train}}$ | 71.8 M | 110 M | 110 M | 110 M | 3.7 M | 91 M |
| $N_{\text{params, total}}$ | 144 M | 182 M | 182 M | 182 M | 3.7 M | 163 M |
| **Representation Model Parameters** |     |                     |                  |                     |                       |               |                         |
| Channels               | - | 128 | 128 | 128 | 32 | 128 |
| Channel Multiplier     | - | [1, 2, 2] | [1, 2, 2, 2] | [1, 2, 2] | [1, 2, 2] | [1, 2] |
| Residual Blocks per Layer | - | 4 | 4 | 4 | 2 | 4 |
| Attention resolutions (1 head) | - | [32, 16] | [32, 16, 8] | [32, 16] | [14, 7] | [32] |
| Dropout                | - | 0.1 | 0.1 | 0.1 | - | 0.1 |
| Class-conditional      | - | ✓ | ✓ | ✓ | ✓ | ✓ |
| Timestep-conditional   | - | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1: Hyperparameters for all models.
A.3 DM Objective

For completeness, we include the derivation of the DM objective in Eq. 2.23 as done in [1].

\[
L = \mathbb{E}_q \left[ -\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)} \right] 
= \mathbb{E}_q \left[ -\log p(x_T) - \sum_{t=1}^T \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} \right] 
= \mathbb{E}_q \left[ -\log p(x_T) - \sum_{t>1}^T \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} - \log \frac{p_\theta(x_0 \mid x_1)}{q(x_1 \mid x_0)} \right] 
= \mathbb{E}_q \left[ -\log p(x_T) - \sum_{t>1}^T \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1}, x_0)} \frac{q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)} - \log \frac{p_\theta(x_0 \mid x_1)}{q(x_1 \mid x_0)} \right] 
= \mathbb{E}_q \left[ D_{KL}(q(x_T \mid x_0) \parallel p(x_T)) + \log p_\theta(x_0 \mid x_1) 
+ \sum_{t>1}^T D_{KL}(q(x_{t-1} \mid x_t, x_0) \parallel p_\theta(x_{t-1} \mid x_t)) \right] \quad (A.1)

An alternative derivation of the DM objective is presented in [2], which is not tractable to estimate but demonstrates progressive coding properties of classical DMs.
\[
L = \mathbb{E}_q \left[ -\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)} \right] \\
= \mathbb{E}_q \left[ -\log p(x_T) - \sum_{t=1}^T \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} \right] \\
= \mathbb{E}_q \left[ -\log p(x_T) - \sum_{t=1}^T \log \frac{p_\theta(x_{t-1} \mid x_t) q(x_{t-1})}{q(x_{t-1} \mid x_t) q(x_t)} \right] \\
= \mathbb{E}_q \left[ -\log p(x_T) - \sum_{t=1}^T \log \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t)} - \log \frac{q(x_0)q(x_1) \ldots q(x_{T-1})}{q(x_1) \ldots q(x_{T-1}) q(x_T)} \right] \\
= D_{KL}(q(x_T) \parallel p(x_T)) + \mathbb{E}_q \left[ \sum_{t=1}^T D_{KL}(q(x_{t-1} \mid x_t) \parallel p_\theta(x_{t-1} \mid x_t)) \right] + H(x_0)
\]

(A.2)

### A.4 Representation-conditional Objective

Below, we derive the ELBO for the representation-conditional DM (Eq. [2.35]) as used in the LRDM.

For the forward process
\[
q_\phi(x_{1:T}, r \mid x_0) = q(x_{1:T} \mid x_0)q_\phi(r \mid x_0) \tag{A.3}
\]

and the joint probability
\[
p_\theta(x_{0:T}, r) = p_\theta(x_{0:T} \mid r)p(r) = p(r)p(x_T)\prod_{t=1}^T p_\theta(x_{t-1} \mid x_t, r) \tag{A.4}
\]

the (negative) ELBO becomes
The negative ELBO can be derived analogously to the LRDM.

\[
\mathbb{E}[- \log p_\theta(x_0)] \leq \mathbb{E}_q \left[ - \log \frac{p_\theta(x_{0:T}, r)}{q(x_{1:T} | x_0)q(r | x_0)} \right] \\
= \mathbb{E}_q \left[ - \log p(x_T) - \log \frac{p(r)}{q(r | x_0)} - \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1} | x_t, r)}{q(x_t | x_{t-1})} \right] \\
= \mathbb{E}_q \left[ - \log p(x_T) - \log \frac{p(r)}{q(r | x_0)} - \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1} | x_t, r) q(x_{t-1} | x_0)}{q(x_{t-1} | x_t, x_0) q(x_t | x_0)} \right] \\
= \mathbb{E}_q \left[ \sum_{t=1}^{T} D_{KL}(q(x_{t-1} | x_t, x_0) \parallel p_\theta(x_{t-1} | x_t, r)) \right] \\
+ D_{KL}(q(r | x_0) \parallel p(r)) \\
+ D_{KL}(q(x_T | x_0) \parallel p(x_T)) \\
\]  

(A.5)

where the first two terms can be identified as the DM loss and the third term regularizes the representation encoder towards the latent prior.

For the t-LRDM, we have a set of representations \( \{r_t\}_{t=1}^{T} \) with the following model setup:

\[
q_\phi(x_{1:T}, r_{1:T} | x_0) = q(x_{1:T} | x_0)q_\phi(r_{1:T} | x_0) = \prod_{t=1}^{T} q(x_t | x_{t-1})q_\phi(r_t | x_0) \\
p_\theta(x_{0:T}, r_{1:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} | x_t, r_t)p(r_t) \\
\]  

(A.6)

The negative ELBO can be derived analogously to the LRDM.

\[
\mathbb{E}[- \log p_\theta(x_0)] \leq \mathbb{E}_q \left[ - \log \frac{p_\theta(x_{0:T}, r_{1:T})}{q_\phi(x_{1:T}, r_{1:T} | x_0)} \right] \\
= \mathbb{E}_q \left[ \sum_{t=1}^{T} D_{KL}(q(x_{t-1} | x_t, x_0) \parallel p_\theta(x_{t-1} | x_t, r_t)) \right] \\
+ \sum_{t=1}^{T} D_{KL}(q(r_t | x_0) \parallel p(r_t)) \\
+ D_{KL}(q(x_T | x_0) \parallel p(x_T)) \\
\]  

(A.7)

65
A.5 Additional Samples

Figure A.2: LRDM (VQ-AE) reconstructions with representation of shape $64 \times 8 \times 8$, using probabilistic sampling and re-sampling $z_T$ for each reconstruction, $\lambda = 10^{-4}$, CelebA-HQ. $r$ is sampled from the approximate posterior.
Figure A.3: LRDM (VQ-AE) reconstructions for different representation dimensionalities, using probabilistic sampling and re-sampling $z_T$ for each reconstruction ($\lambda = 10^{-4}$, LSUN-Churches). $r$ is sampled from the approximate posterior.
Figure A.4: LRDM (VQ-AE) interpolations (slerp) on LSUN-Churches for representation shape $64 \times 8 \times 8$, $\lambda = 10^{-4}$, using probabilistic sampling and re-sampling $z_T$ for each sample.
Figure A.5: LRDM (VQ-AE) interpolations (slerp) on CelebA-HQ for representation shape $64 \times 8 \times 8$, $\lambda = 10^{-4}$, using probabilistic sampling and re-sampling $z_T$ for each sample. The additional stochasticity (compared to the deterministic sampling in Fig. 4.8) through the probabilistic sampling manifests in variations of small-scale features.
Figure A.6: Unconditional samples from the LRDM (VQ-AE) for different regularization factors. Increasing $\lambda$ leads to a representation distribution closer to the gaussian prior.
Figure A.7: Image reconstructions from the LRDM (VQ-AE) for different regularization factors. $z_T$ is re-sampled for each reconstruction. For low $\lambda$, the representation allows for faithful reconstructions, varying $z_T$ incudes variations in local image features. For intermediate $\lambda$, only the overall shape and color scheme is encoded. For high $\lambda$, the LRDM approaches the LDM.