Deconstructing Denoising Diffusion Models for Self-Supervised Learning

Xinlei Chen[1] Zhuang Liu[1] Saining Xie[2] Kaiming He[1]
[1]FAIR, Meta [2]New York University

https://arxiv.org/abs/2401.14404

Introduction

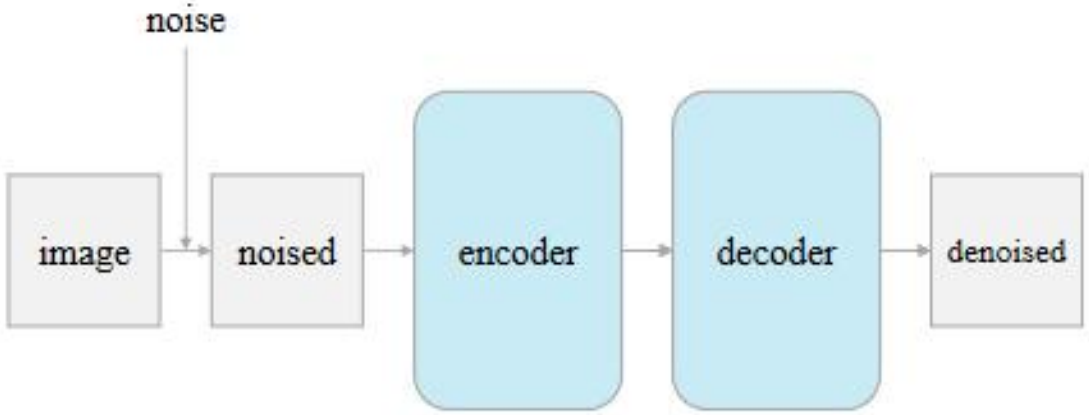
Popularly known as *Denoising Diffusion Models* (DDM) today, these methods achieve impressive image generation quality—in fact, these generation models are so good that they appear to have strong recognition representations for understanding the visual content.

At the core of our philosophy is to *deconstruct* a DDM, changing it stepby-step into a classical DAE.

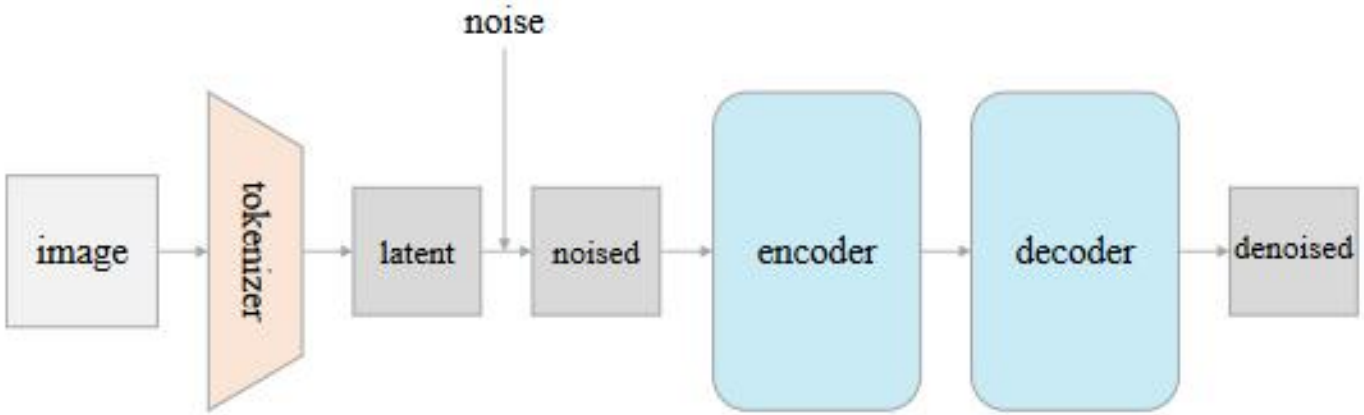
This research process gains us new understandings on what are the critical components for a DAE to learn good representations.

Background

Denoising Autoencoder (DAE)： An autoencoder that receives a corrupted data point as input and is trained to predict the original, uncorrupted data point as its output.



(a) a classical **Denoising Autoencoders** (DAE)



(b) a modern **Denoising Diffusion Model** (DDM) on a latent space

Background

Self-supervised learning is a paradigm in machine learning where a model is trained on a task using the data itself to generate supervisory signals, rather than relying on external labels provided by humans.

linear probing: We take a model that was trained on some task, such as a language model. We generate representations using the model, and train another classifier that takes the representations and predicts some property. If the classifier performs well, we say that the model has learned information relevant for the property.

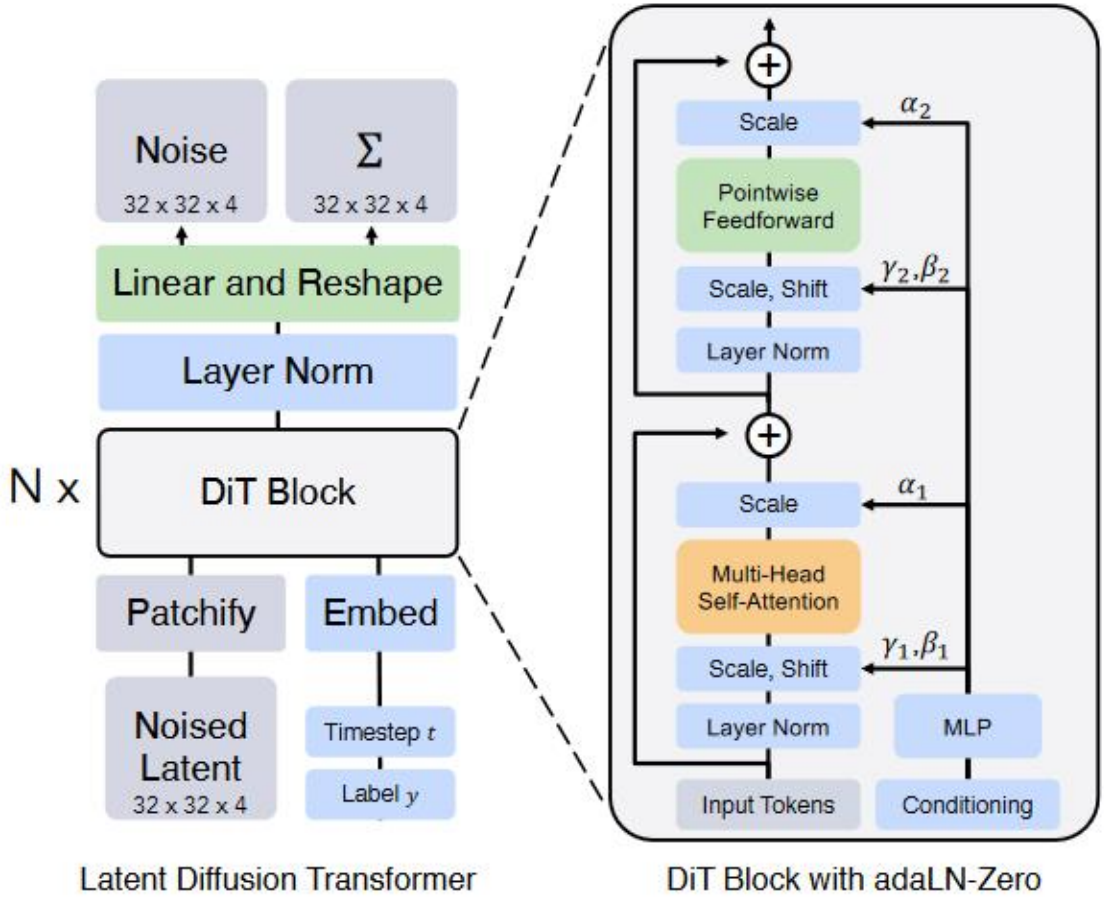
Background Diffusion Transformer (DiT)

We choose this Transformer-based DDM for several reasons:

(i) Transformer-based architectures can provide fairer comparisons with other selfsupervised learning baselines driven by Transformers;

(ii) DiT has a clearer distinction between the encoder and decoder, while a UNet's encoder and decoder are connected by skip connections;

(iii) DiT trains much faster than other UNet-based DDMs while achieving better generation quality.



Background

We use the DiT-Large (**DiT-L**) variant as our DDM baseline (24 blocks) .

We evaluate the representation quality (linear probe accuracy) of the encoder, which has 12 blocks, referred to as "$\frac{1}{2}$ L" (half large).

By default, we train the models for 400 epochs on ImageNet with a resolution of 256×256 pixels.

With DiT-L, we report a linear probe accuracy of **57.5%** using its encoder. The generation quality (FID-50K) of this DiT-L model is **11.6**. This is the starting point of our destructive trajectory.

Deconstructing Denoising Diffusion Models

We first adapt the generation-focused settings in DiT to be more oriented toward self-supervised learning

Next, we deconstruct and simplify the tokenizer step by step

Finally, we attempt to reverse as many DDM-motivated designs as possible, pushing the models towards a classical DAE

Remove class-conditioning

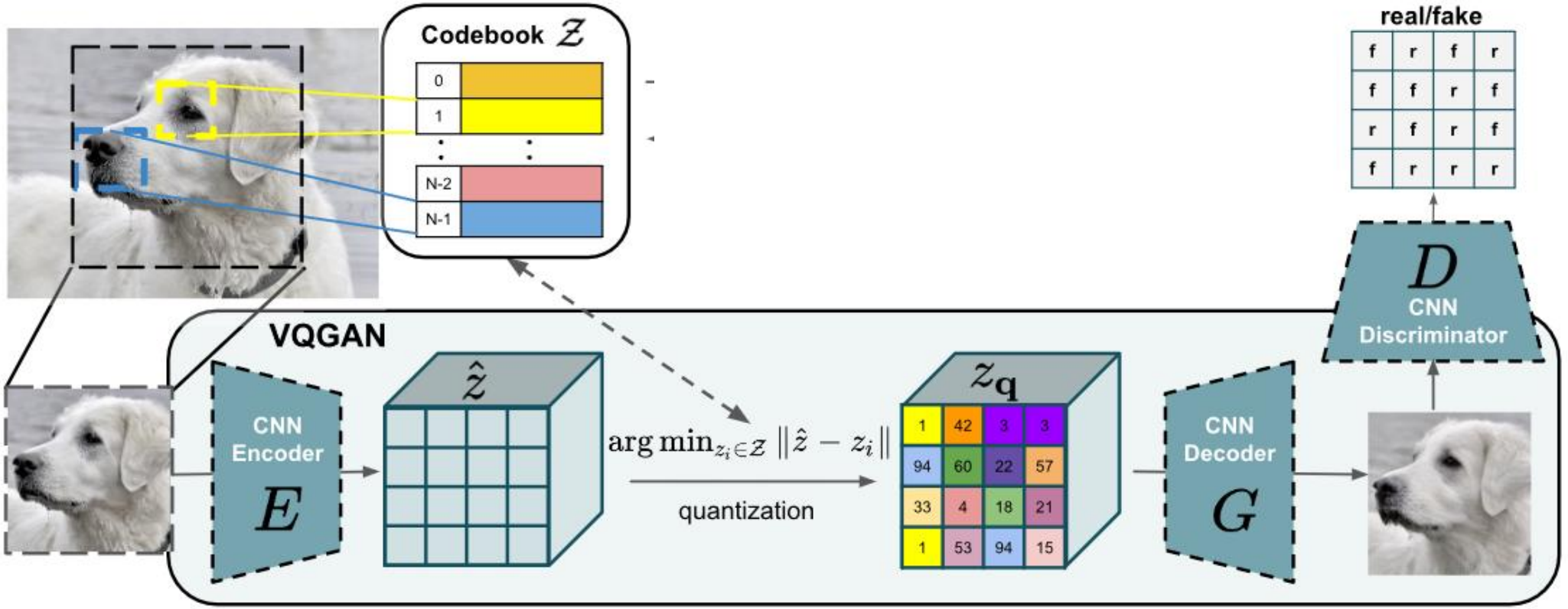
	acc. (↑)	FID (↓)
DiT baseline	57.5	11.6
+ remove class-conditioning	62.5	30.9
+ remove VQGAN perceptual loss	58.4	54.3
+ remove VQGAN adversarial loss	59.0	75.6
+ replace noise schedule	63.4	93.2

The usage of class labels is simply not legitimate in the context of our self-supervised learning study

Directly conditioning the model on class labels may reduce the model's demands on encoding the information related to class labels.

Removing the class-conditioning can force the model to learn more semantics.

Deconstruct VQGAN



$$\mathcal{L}_{\text{VQ}}(E, G, \mathcal{Z}) = \|x - \hat{x}\|^2 + \|\text{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 + \|\text{sg}[z_{\mathbf{q}}] - E(x)\|_2^2.$$

$$\mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D) = [\log D(x) + \log(1 - D(\hat{x}))]$$

Deconstruct VQGAN

VQGAN? VAE?

"We use an off-the-shelf pre-trained variational autoencoder (VAE) model from Stable Diffusion"

In our baseline, the VQGAN tokenizer, presented by LDM and inherited by DiT, is trained with multiple loss terms:

autoencoding reconstruction loss $\|x - \hat{x}\|^2$

KL-divergence regularization loss $\mathbb{KL}\left[f(x)|\mathcal{N}\right]$

adversarial loss $\mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D) = [\log D(x) + \log(1 - D(\hat{x}))]$

perceptual loss

Deconstruct VQGAN

	acc. (↑)	FID (↓)
DiT baseline	57.5	11.6
+ remove class-conditioning	62.5	30.9
+ remove VQGAN perceptual loss	58.4	54.3
+ remove VQGAN adversarial loss	59.0	75.6
+ replace noise schedule	63.4	93.2

Perceptual loss based on a supervised VGG net trained for ImageNet classification.

As the perceptual loss involves a supervised pretrained network, using the VQGAN trained with this loss is not legitimate.

This comparison reveals that *a tokenizer trained with the perceptual loss (with class labels) in itself provides semantic representations*.

Deconstruct VQGAN

	acc. (↑)	FID (↓)
DiT baseline	57.5	11.6
+ remove class-conditioning	62.5	30.9
+ remove VQGAN perceptual loss	58.4	54.3
+ remove VQGAN adversarial loss	59.0	75.6
+ replace noise schedule	63.4	93.2

We train the next VQGAN tokenizer that further removes the adversarial loss.

With this, **our tokenizer at this point is essentially a VAE**, which we move on to deconstruct in the next subsection.

Replace noise schedule

	acc. (↑)	FID (↓)
DiT baseline	57.5	11.6
+ remove class-conditioning	62.5	30.9
+ remove VQGAN perceptual loss	58.4	54.3
+ remove VQGAN adversarial loss	59.0	75.6
+ replace noise schedule	63.4	93.2

$$z_t = \gamma_t z_0 + \sigma_t \epsilon \qquad x_t = \sqrt{\overline{\alpha}_t} x_0 + \sqrt{1 - \overline{\alpha}_t}\epsilon$$

This allows the model to spend more capacity on cleaner images

the original schedule focuses too much on noisier regimes.

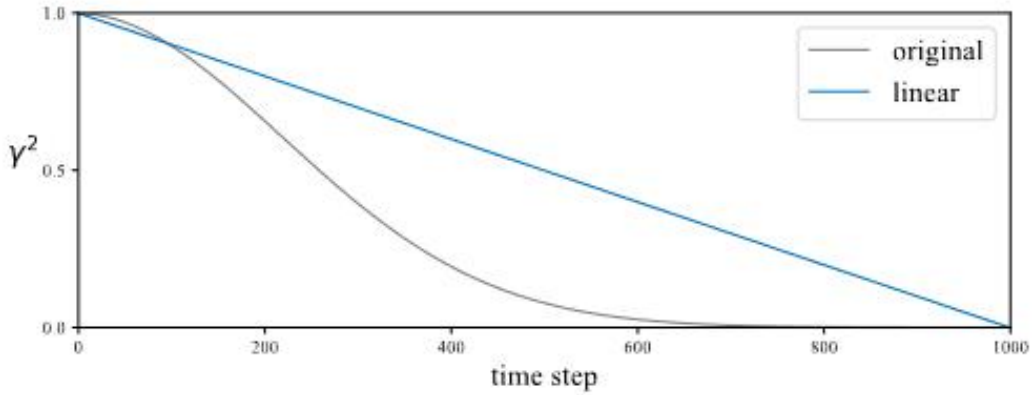


Figure 3. **Noise schedules**. The original schedule [23, 32], which sets $\gamma_t^2 = \Pi_{s=1}^{t}(1-\beta_s)$ with a linear schedule of β, spends many time steps on very noisy images (small γ). Instead, we use a simple schedule that is linear on γ^2, which provides less noisy images.

Reorienting DDM for Self-supervised Learning

	acc. (↑)	FID (↓)
DiT baseline	57.5	11.6
+ remove class-conditioning	62.5	30.9
+ remove VQGAN perceptual loss	58.4	54.3
+ remove VQGAN adversarial loss	59.0	75.6
+ replace noise schedule	63.4	93.2

The results reveal that ***self-supervised learning performance is not correlated to generation quality***. The representation capability of a DDM is not necessarily the outcome of its generation capability.

Deconstructing the Tokenizer

Our deconstruction thus far leads us to a VAE tokenizer. We further deconstruct the VAE tokenizer by making substantial simplifications

Convolutional VAE $\|x - g(f(x))\|^2 + \mathbb{KL}\left[f(x)|\mathcal{N}\right].$

Patch-wise VAE

the VAE encoder and decoder are both linear projections $\|x - U^T V x\|^2 + \mathbb{KL}\left[Vx|\mathcal{N}\right].$

Patch-wise AE $\|x - U^T V x\|^2.$

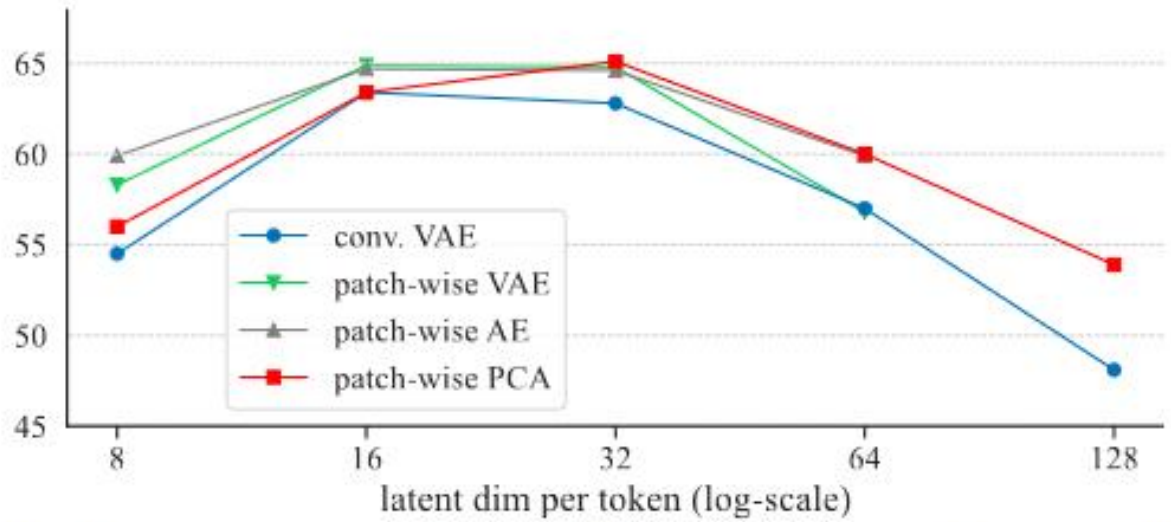
Patch-wise PCA $\|x - V^T V x\|^2.$ in which V satisfies $VV^T = I$

Deconstructing the Tokenizer

all four variants of tokenizers exhibit similar trends

Interestingly, the optimal dimension is relatively low (d is 16 or 32), even though the full dimension per patch is much higher (16×16×3=768).

Surprisingly, the convolutional VAE tokenizer is neither necessary nor favorable. In addition, the KL regularization term is unnecessary, as both the AE and PCA variants work well.



latent dim. d	8	16	32	64	128
conv. VAE (baseline)	54.5	**63.4**	62.8	57.0	48.1
patch-wise VAE	58.3	**64.9**	64.8	56.8	-
patch-wise AE	59.9	**64.7**	64.6	59.9	-
patch-wise PCA	56.0	63.4	**65.1**	60.0	53.9

Deconstructing the Tokenizer

High-resolution, pixel-based DDMs are inferior for selfsupervised learning.

we consider a “naïve tokenizer” that performs identity mapping on patches extracted from resized images.

Interestingly, this pixel-based tokenizer exhibits a similar trend with other tokenizers we have studied, although the optimal dimension is shifted.

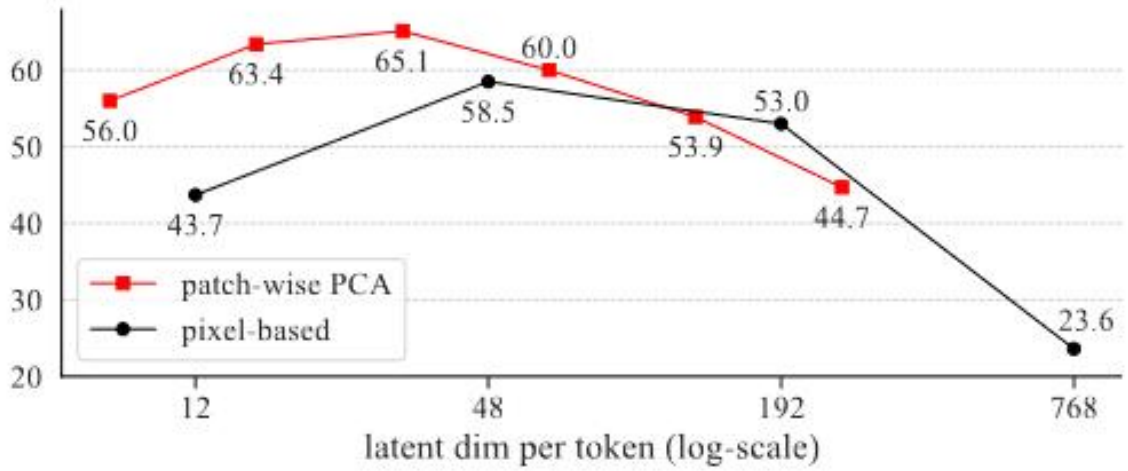


Figure 5. Linear probe results of the **pixel-based tokenizer**, operated on an image size of 256, 128, 64, and 32, respectively with a patch size of 16, 8, 4, 2. The “latent” dimensions of these tokenized spaces are 768, 192, 48, and 12 per token. Similar to other tokenizers we study, this pixel-based tokenizer exhibits a similar trend: a relatively small dimension of the latent space is optimal.

Deconstructing the Tokenizer

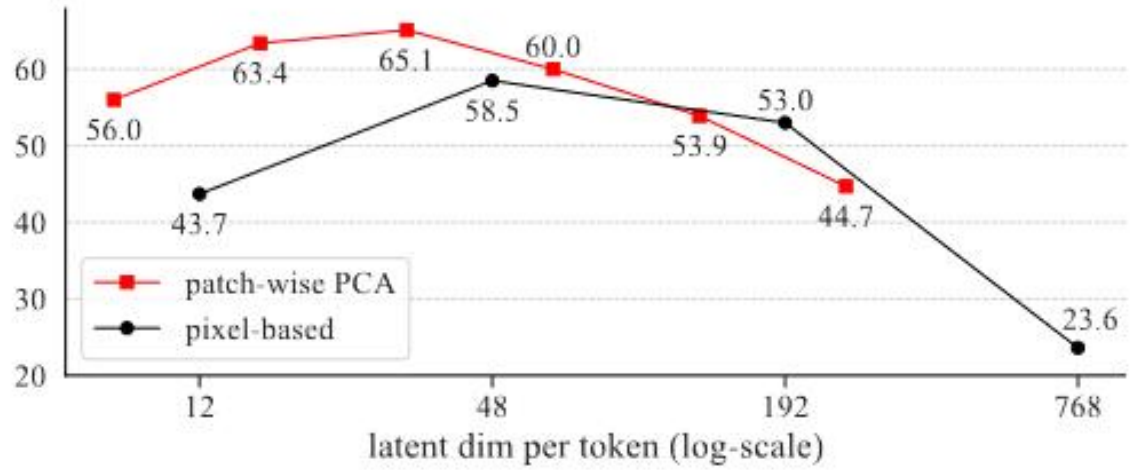


Figure 5. Linear probe results of the **pixel-based tokenizer**, operated on an image size of 256, 128, 64, and 32, respectively with a patch size of 16, 8, 4, 2. The "latent" dimensions of these tokenized spaces are 768, 192, 48, and 12 per token. Similar to other tokenizers we study, this pixel-based tokenizer exhibits a similar trend: a relatively small dimension of the latent space is optimal.

These comparisons show that the tokenizer and the resulting latent space are crucial for DDM/DAE to work competitively in the self-supervised learning scenario.

Predict clean data (rather than noise)

$$\lambda_t \| z_0 - \mathtt{net}(z_t) \|^2$$

	acc.
patch-wise PCA baseline	65.1
+ predict clean data (rather than noise)	62.4
+ remove input scaling (fix $\gamma_t \equiv 1$)	63.6
+ operate on image input with inv. PCA	63.6
+ operate on image output with inv. PCA	63.9
+ predict original image	64.5

We find that setting $\lambda_t = \gamma_t^2$ works better in our scenario. Intuitively, it simply puts more weight to the loss terms of the cleaner data (larger γ_t).

Even though we suffer from a degradation in this step, we will stick to this modification from now on, as our goal is to move towards a classical DAE

Remove input scaling

$$z_t = \gamma_t z_0 + \sigma_t \epsilon$$

$$\lambda_t \| z_0 - \mathtt{net}(z_t) \|^2$$

	acc.
patch-wise PCA baseline	65.1
+ predict clean data (rather than noise)	62.4
+ remove input scaling (fix $\gamma_t \equiv 1$)	63.6
+ operate on image input with inv. PCA	63.6
+ operate on image output with inv. PCA	63.9
+ predict original image	64.5

In modern DDMs, the input is scaled by a factor of γ_t. This is not common practice in a classical DAE.

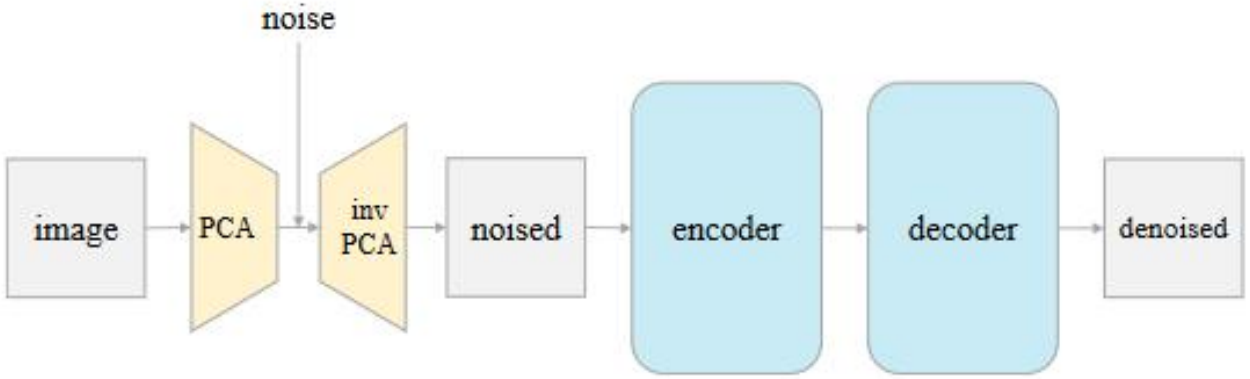
we set $\gamma_t = 1$. As γ_t is fixed, we need to define a noise schedule directly on σ_t. We simply set σ_t as a linear schedule from 0 to $\sqrt{2}$. Moreover, we empirically set the weight as $\lambda_t = 1/(1+\sigma_t^2)$, which again puts more emphasis on cleaner data (smaller σ_t).

scaling the data by γ_t is not necessary in our scenario.

Operate on the *image* space with inverse PCA

	acc.
patch-wise PCA baseline	65.1
+ predict clean data (rather than noise)	62.4
+ remove input scaling (fix $\gamma_t \equiv 1$)	63.6
+ operate on image input with inv. PCA	63.6
+ operate on image output with inv. PCA	63.9
+ predict original image	64.5

Ideally, we hope our DAE can work directly on the image space while still having good accuracy.



Operate on the *image* space with inverse PCA

	acc.
patch-wise PCA baseline	65.1
+ predict clean data (rather than noise)	62.4
+ remove input scaling (fix $\gamma_t \equiv 1$)	63.6
+ operate on image input with inv. PCA	63.6
+ operate on image output with inv. PCA	63.9
+ predict original image	64.5

Applying this modification on the input side (while still predicting the output on the latent space) has 63.6% accuracy. Further applying it to the output side (i.e., predicting the output on the image space with inverse PCA) has 63.9% accuracy.

Both results show that operating on the image space with inverse PCA can achieve similar results as operating on the latent space.

Predict original image

While inverse PCA can produce a prediction target in the image space, this target is not the original image.

	acc.
patch-wise PCA baseline	65.1
+ predict clean data (rather than noise)	62.4
+ remove input scaling (fix $\gamma_t \equiv 1$)	63.6
+ operate on image input with inv. PCA	63.6
+ operate on image output with inv. PCA	63.9
+ predict original image	64.5

When we let the network predict the original image, the "noise" introduced includes two parts:
(i) the additive Gaussian noise, whose intrinsic dimension is d, and
(ii) the PCA reconstruction error, whose intrinsic dimension is D − d (D is 768).

Predict original image

While inverse PCA can produce a prediction target in the image space, this target is not the original image.

	acc.
patch-wise PCA baseline	65.1
+ predict clean data (rather than noise)	62.4
+ remove input scaling (fix $\gamma_t \equiv 1$)	63.6
+ operate on image input with inv. PCA	63.6
+ operate on image output with inv. PCA	63.9
+ predict original image	64.5

we can compute the residue r projected onto the full PCA space:

$$r \triangleq V(x_0 - net(x_t))$$

Here V is the D-by-D matrix representing the full PCA bases.

Then we minimize the following loss function: $\lambda_t \sum_{i=1}^{D} w_i r_i^2$.

weight w_i is 1 for i ≤ d, and 0.1 for d < i ≤ D. Intuitively, wi down-weights the loss of the PCA reconstruction error.

Single noise level

We note that multi-level noise, given by noise scheduling, is a property motived by the diffusion process in DDMs; it is conceptually unnecessary in a classical DAE.

We fix the noise level as a constant. Using this single-level noise achieves decent accuracy of 61.5%, a 3% degradation vs. the multi-level noise counterpart (64.5%).

Using multiple levels of noise is analogous to **a form of data augmentation** in DAE: it is beneficial, but not an enabling factor. This also implies that the representation capability of DDM is **mainly gained by the denoising-driven process, not a diffusion-driven process.**

Summary

We use the entry at the end of Tab as our final DAE instantiation We refer to this method as “latent Denoising Autoencoder”, or in short, **l-DAE**.

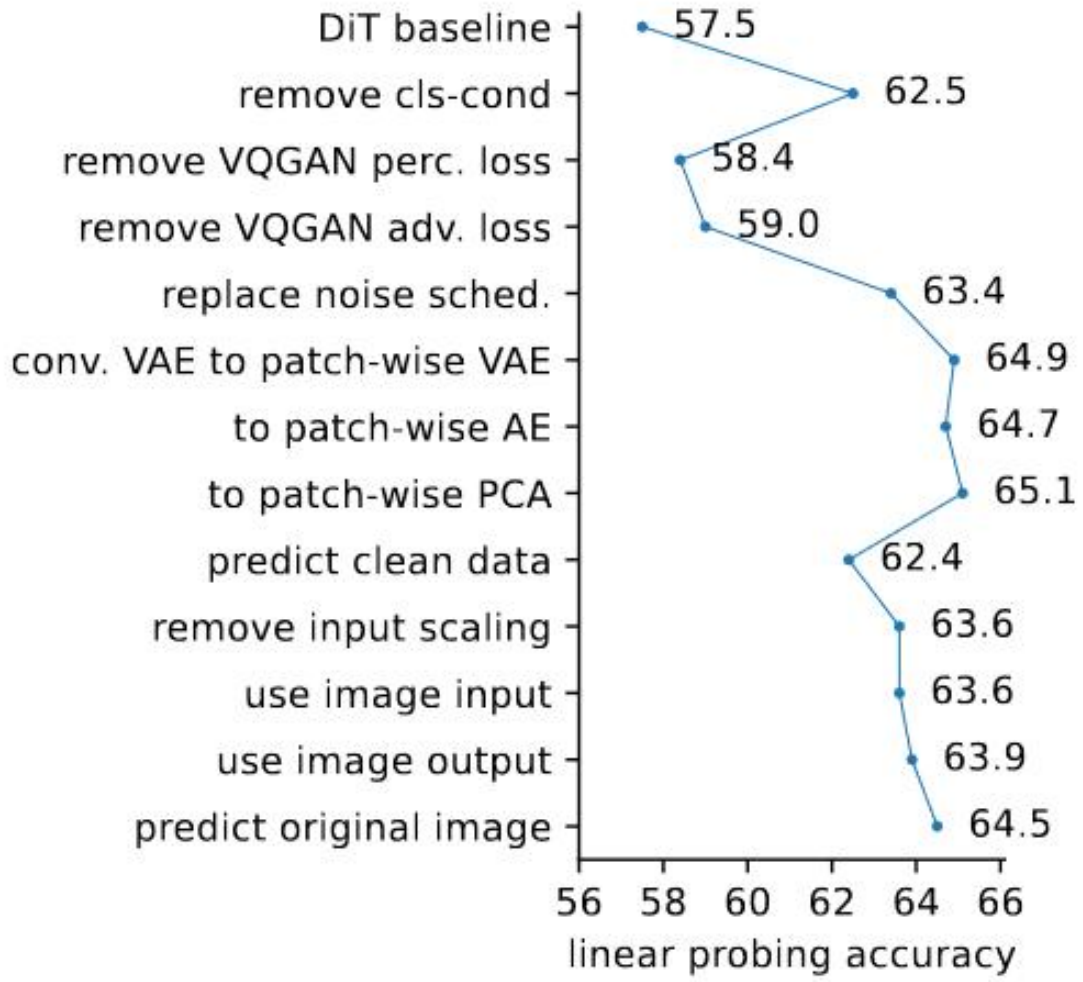


Figure 6. **The overall deconstructive trajectory** from a modern DDM to *l*-DAE, summarizing Tab. 1, Tab. 2, and Tab. 3. Each line is based on a modification of the immediately preceding line.

Analysis and Comparisons

Denoising results

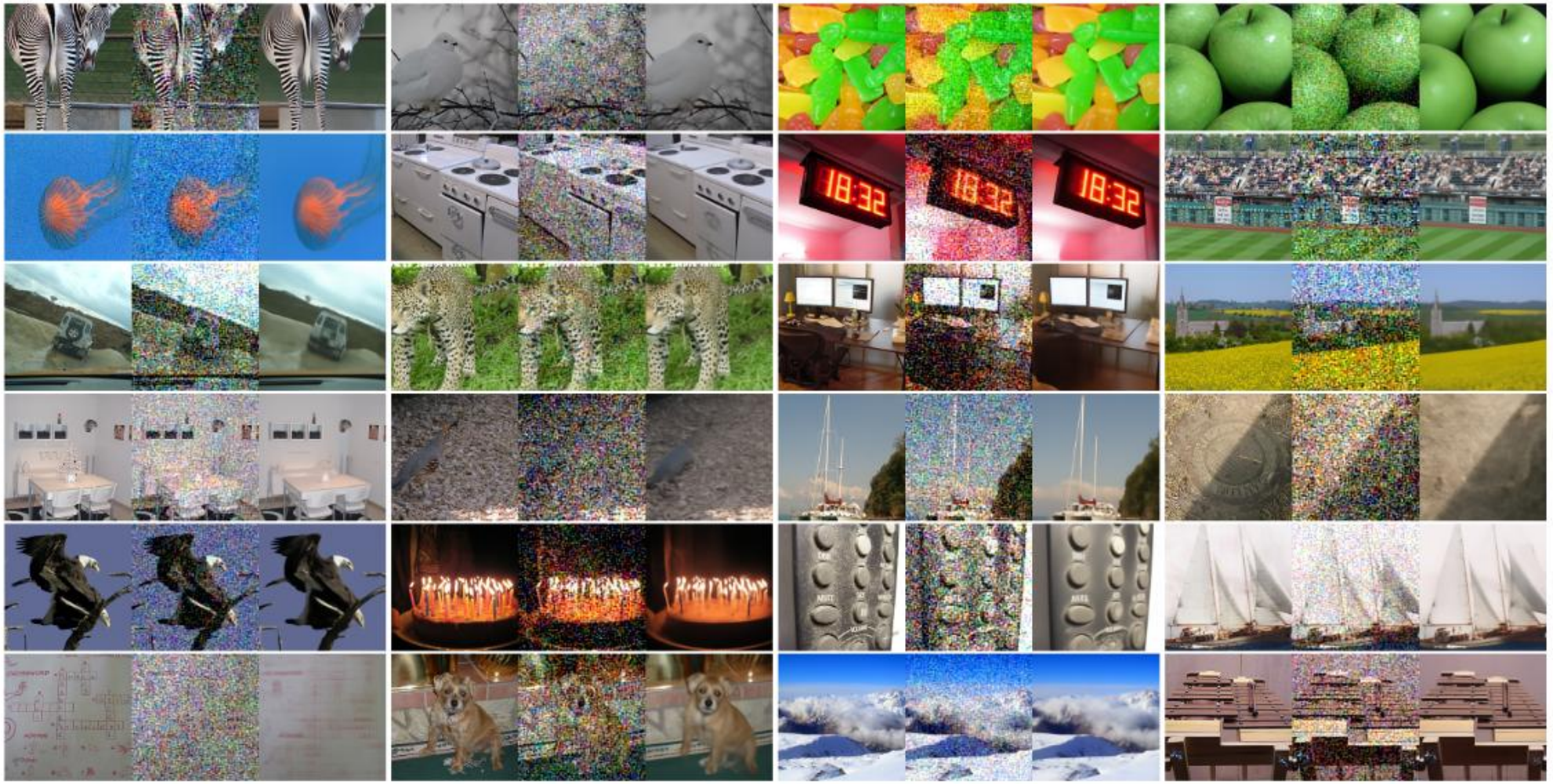


Figure 8. **Denoising results** of *l*-DAE, evaluated on ImageNet validation images. *This denoising problem, serving as a pretext task, encourages the network to learn meaningful representations in a self-supervised manner.* For each case, we show: (**left**) clean image; (**middle**) noisy image that is the input to the network, where the noise is added to the latent space; (**right**) denoised output.

Analysis and Comparisons

epochs	400	800	1600
acc.	65.0	67.5	69.6

Training epochs

All our experiments thus far are based on 400-epoch training. Following MAE , we also study training for 800 and 1600 epochs

encoder	ViT-B	ViT-$\frac{1}{2}$L	ViT-L
acc.	60.3	65.0	70.9

Model size

We observe a good scaling behavior w.r.t. model size: scaling from ViT-B to ViT-L has a large gain of 10.6%.

Analysis and Comparisons

Comparison with previous baselines.

method	ViT-B (86M)	ViT-L (304M)
MoCo v3	76.7	77.6
MAE	68.0	75.8
l-DAE	66.6	75.0

Interestingly, l-DAE performs decently in comparison with MAE, showing a degradation of 1.4% (ViT-B) or 0.8% (ViT-L).
we have largely closed the accuracy gap between MAE and a DAE-driven method.

Last, we observe that autoencoder-based methods (MAE and l-DAE) still fall short in comparison with contrastive learning methods under this protocol, especially when the model is small.