Denoising Autoregressive Representation Learning

Yazhe Li [1] **Jorg Bornschein** [1] **Ting Chen** [1 2]

Abstract

In this paper, we explore a new generative approach for learning visual representations. Our method, DARL, employs a decoder-only Transformer to predict image patches autoregressively. We find that training with Mean Squared Error (MSE) alone leads to strong representations. To enhance the image generation ability, we replace the MSE loss with the diffusion objective by using a denoising patch decoder. We show that the learned representation can be improved by using tailored noise schedules and longer training in larger models. Notably, the optimal schedule differs significantly from the typical ones used in standard image diffusion models. Overall, despite its simple architecture, DARL delivers performance remarkably close to state-of-the-art masked prediction models under the fine-tuning protocol. This marks an important step towards a unified model capable of both visual perception and generation, effectively combining the strengths of autoregressive and denoising diffusion models.

1. Introduction

With the rise of Large Language Models (LLMs), generative pre-training has become increasingly popular. Representations learned via next token prediction improves the performance when transferred to a diverse set of downstream tasks (Radford et al., 2018; 2021; Brown et al., 2020; Raffel et al., 2023). Beyond learning representations, the model directly generates language, acting as a user interface and allowing for interactive adjustments (Liu et al., 2021). The likelihood-based pre-training objective enables us to investigate scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022). These scaling laws predict how a model's pre-training loss relates to its capacity and the amount of training data. Generally, we anticipate that achieving a lower pre-training loss leads to superior performance on downstream tasks.

In vision, however, representation learning and image generation often use separate techniques. For learning representations, methods such as contrastive learning (van den Oord et al., 2019; Chen et al., 2020b; He et al., 2019), distillation-based self-supervised learning (Grill et al., 2020; Caron et al., 2021) and masked image modelling (MIM) (He et al., 2021; Bao et al., 2022) are widely used. Despite their strengths in learning robust visual and cross-modal (Radford et al., 2021) representations, as well as their efficient use of model capacity, these methods lack generation capabilities. Furthermore, the pre-training loss, influenced by the difficulty of the pre-training task, does not serve as a reliable indicator of performance on downstream tasks.

In this paper, we investigate the potential of a unified model capable of both visual perception and generation by combining autoregressive and denoising diffusion models. We use a straightforward architecture - a decoder-only Transformer - which predicts the next image patch based on a sequence of previously observed patches. Instead of absolute or learnable positional encoding, we implement relative positional encodings through the utilization of decomposed rotary position embedding (2D RoPE). We show that 2D RoPE improves the performance, in particular for causal Transformers. When trained with MSE loss, the fine-tuning performance of the model is not far away from the state-of-the-art representation methods. To enhance the image generation ability, we introduce a denosing patch decoder and substitute the MSE loss with the diffusion objective. Our results demonstrate that model performance depends on the noise schedule employed during training. When the noise schedule is more focused on high noise levels, training with diffusion objective leads to an improvement which becomes more pronounced with extended pre-training epochs. Due to the concentration on high noise level, the optimal noise schedule differs significantly from those suitable for generation purpose (Chen et al., 2020a; Nichol & Dhariwal, 2021; Rombach et al., 2022; Chen, 2023; Hoogeboom et al., 2023). This deviation from image generation models can be interpreted as the competition for model capacity between higher-level abstraction and lower-level details. Overall, our method significantly advances representation learning with generative pre-training. Under fair comparison conditions,

[1]Google DeepMind, London, UK. [2]xAI, San Francisco, US. Work done while at Google DeepMind. Correspondence to: Yazhe Li <yazhe@google.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

our best model achieves performance remarkably close to state-of-the-art masked prediction models like Masked Autoencoders (MAE), with a minor performance gap of 1%. This demonstrates the potential of generative pre-training for vision data.

Our contributions and findings:

- **Denoising autoregressive representation learning:** We propose DARL, a generative approach for learning visual representations that demonstrates performance comparable to leading masked prediction models.
- **Decomposed RoPE:** We show that causal Transformers significantly benefit from employing 2D RoPE, an implementation of relative positional encodings.
- **MSE and diffusion objectives:** We observe that training on MSE loss alone yields strong performance. Incorporating a denoising patch decoder further enhances representation quality and generative ability, especially in larger models with extended training and optimized noise schedules. Denoising is also beneficial when using large patch sizes.
- **Patch ordering:** Extensive analysis reveals that raster scan order is near-optimal for fixed patch ordering. Random ordering does not offer any performance advantages.

2. Related Works

Self-supervised Representation Learning learns through solving pretext tasks which are constructed without external labels. Meticulously designed pretext tasks, such as relative location prediction (Doersch et al., 2015), colorization (Zhang et al., 2016), jigsaw puzzle solving (Noroozi & Favaro, 2017) and rotation prediction (Gidaris et al., 2018), are empirically demonstrated to learn meaningful visual representations. Contrastive learning (Bachman et al., 2019; Chen et al., 2020b) involves constructing distinct views of the same image through data augmentation. Given one view, the model is trained to distinguish data originating from the same source image from others. InfoNCE loss (Belghazi et al., 2018; van den Oord et al., 2019) is often used and could be seen as minimizing the distance between positive pairs while maximizing those between negative pairs. Alternative metrics for measuring distances between positive and negative pairs, such as L2 (Grill et al., 2020) or kernel distance (Li et al., 2021), can also be employed. Performance of contrastive learning can be improved by using a momentum encoder (He et al., 2019), which can also be leveraged to remove the necessity of negative examples (Grill et al., 2020; Caron et al., 2021). This can be seen as an instance of self-distillation. Masked prediction task predicts the missing content from a partial input. It has demonstrated strong performance and gained popularity through methods like BERT (Devlin et al., 2019) in Natural Language Processing (NLP), Masked Autoencoders (MAE) (He et al., 2021) and BeiT (Bao et al., 2022) in vision.

Generative Pre-Training. In the vision domain, earlier attempts of using generative models for representation learning includes Variational Autoencoders (VAE) (Kingma & Welling, 2022; Rezende et al., 2014; Higgins et al., 2017; van den Oord et al., 2018) and GANs (Donahue & Simonyan, 2019). With the success of GPT (Radford et al., 2018; 2021; Brown et al., 2020) in NLP, generative pre-training attracts renewed attention. Image-GPT (Chen et al., 2020a) adapts the GPT model for pre-training on images.

Diffusion Models are a class of latent variable models inspired by statistical physics and non-equilibrium thermodynamics (Sohl-Dickstein et al., 2015). It has been demonstrated that diffusion models excel at generating high-quality images (Ho et al., 2020; Dhariwal & Nichol, 2021; Rombach et al., 2022). In addition, they offer flexibility to generate images guided by labels (Dhariwal & Nichol, 2021; Ho & Salimans, 2022) or textual descriptions (Nichol et al., 2022; Saharia et al., 2022). There is also a growing interest in utilizing diffusion models for representation learning (Hudson et al., 2023; Wei et al., 2023).

Autoregressive Models (AR) models have a rich history in language and speech. Developments of innovative architectures, such as recurrent neural networks (Rumelhart et al., 1986), long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Transformers (Vaswani et al., 2023), keep improving their capabilities. In the image domain, AR models are adopted in NADE (Uria et al., 2016), MADE (Germain et al., 2015), PixelRNN (van den Oord et al., 2016a), PixelCNN (van den Oord et al., 2016b), Image Transformer (Parmar et al., 2018) and Image-GPT (Chen et al., 2020a). AR models are tightly related to generative models as they are often trained with a likelihood-based objective functions. Concurrent work (El-Nouby et al., 2024) shows that patch-based image Transformer trained with L2 loss possesses similar scaling property as their NLP counterpart. However, their model cannot be regarded as a full-fledged generative model. It's perhaps worth noting that diffusion models can also be seen as AR model, but in the frequency space (Dieleman, 2023).

3. Denoising Autoregressive Representation Learning (DARL)

3.1. Architecture

The architecture used for our study is straighforward (see Figure 1): a Vision Transformer (ViT) (Dosovitskiy et al., 2021) backbone with causal attention masking. Adopting this backbone allows us to make a direct comparison with prior representation learning methods.

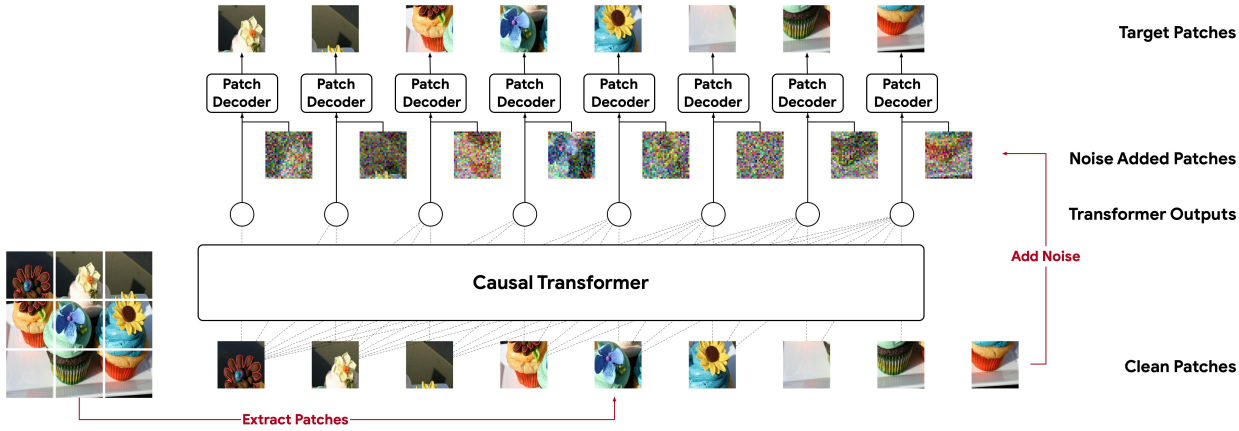


Figure 1. **DARL architecture.** Images are segmented into non-overlapping patches to form an input sequence. Causal attention masking is applied to the Vision Transformer. Random noises, parameterized by a noise schedule, are independently sampled to corrupt the patches. The output of the Transformer, along with the corrupted patch, are taken as input to the patch decoder to reconstruct the clean patch.

Following the ViT approach, images are segmented into non-overlapping patches and linearly projected into an embedding space. The resulting embeddings are arranged in raster scan order, and a start-of-sequence token is prepended. This forms the inputs to the Transformer. The combination of causal attention masking and the one-position offset by start-of-sequence token ensures that the patch generated at current position only receives information from the previous patches.

We use relative positional encodings in the form of decomposed RoPE (detailed in Section 3.4). We find that relative positional encodings outperform absolute and learnable ones, in particular for AR models. Extending RoPE from 1D to 2D allows better generalization for image data (see Section 4.1).

A patch decoder is responsible for mapping the Transformer output into pixel space. In case of training with MSE loss, we simply use a linear layer. In the case of diffusion objective, we use a denoising patch decoder consisting of a single layer of Transformer block which processes the output of the backbone and the embedding of the corrupted patch (treating each as an input token).

3.2. Training Objective

The training uses the standard AR objective function:

$$\mathcal{L}(\theta;\mathcal{D}) = -\sum_{\mathcal{D}}\sum_{t=1}^{T}\log p_\theta(x_t|x_{<t}) \tag{1}$$

Mean Squared Error (MSE) is the simplest loss we can adopt for patch prediction. With MSE, Equation (1) becomes

$$\mathcal{L}_{MSE}(\theta;\mathcal{D}) \propto \sum_{\mathcal{D}}\sum_{t=1}^{T}\|f(x_{<t}) - x_t\|^2$$

$f(x_{<t})$ is output of the Transformer. This can be interpreted as modelling x_t with a Gaussian distribution centered at $f(x_{<t})$ with a constant variance. Despite the probabilistic interpretation of MSE loss, it is rarely used in state-of-the-art generative models, because the unimodal assumption of the Gaussian distribution leads to blurry images.

Diffusion Objective allows the model to express a multimodal believe over the patch content, yielding better quality of the generation. The training is performed by optimizing the Variational Lower Bound (ELBO) of the image patch distribution:

$$\begin{aligned}
&\mathcal{L}_{\text{DIFF}}(\theta;\mathcal{D}) = -\sum_{\mathcal{D}}\sum_{t=1}^{T}\mathcal{L}_{\text{ELBO}}(x_t;x_{<t},\theta)\\
&\mathcal{L}_{\text{ELBO}}(x;\theta) = \mathbb{E}_{q(x^{1:S}|x^0)}\left[\log p_\theta(x^0|x^1)\right]\\
&-\sum_{s=2}^{S}\mathbb{E}_{q(x^s|x^0)}\left[KL\left[q(x^{s-1}|x^s,x^0)||p_\theta(x^{s-1}|x^s)\right]\right]\\
&+H[q(x^S|x^0)] - H[p(x_S)]
\end{aligned}$$

We use subscript t for the token at the t-th step of the sequence, and reserve the superscript s for the timestep of the diffusion process.

Since the denoising patch decoder takes the corrupted target patches as input and predicts the clean ones, it can be regarded as performing a denoising task, which was a popular early representation learning technique (Bengio et al., 2006). In addition, the decoder is conditioned on previous uncorrupted patches through the autoregression process and can thus be considered a conditional diffusion model.

In practical terms, changing the loss function from MSE to the diffusion objective doesn't require any changes to the backbone network - only the replacement of the patch decoder with a denoising patch decoder.

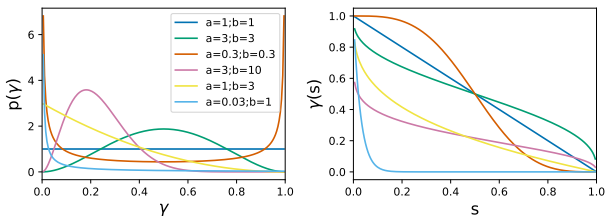


Figure 2. **Noise schedule.** γ is sampled directly from a Beta distribution parameterized by a and b. **Left**: Beta distributions with varying values for a and b. **Right**: the corresponding transformation function if γ is computed from a transformation from s sampled from a uniform distribution.

3.3. Patch Diffusion

In this section, we describe in details how the diffusion is implemented in our study. We mainly follow the DDPM formulation (Ho et al., 2020). Instead of applying the same corruption to the whole image, the image patches are corrupted independently.

Forward Process For each patch x (we drop the patch index t for clarity), the forward process $q(x^s|x^{s-1}) = \mathcal{N}(\sqrt{\alpha(s)}x^{s-1}, (1-\alpha(s))I)$ gradually adds noise to the image patches. With $\gamma(s) = \prod_{s' \leq s} \alpha(s')$, we can analytically write the result of the forward process given an original image patch x^0:

$$q(x^s|x^0) \sim \mathcal{N}(\sqrt{\gamma(s)}x^0, (1-\gamma(s))I)$$
$$i.e. \quad x^s = \sqrt{\gamma(s)}x^0 + \sqrt{1-\gamma(s)}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

$\gamma(s)$ is the noise schedule employed for the forward process. In DDPM, s is randomly sampled from a uniform distribution $\mathcal{U}[0,1]$ and a function $\gamma(.)$ maps s to $\gamma \in [0,1]$. This is equivalent to sampling γ directly from a distribution P of which the cumulative probability $Pr(X \leq x) = T^{-1}(x)$ and $T^{-1}(x)$ is the inverse function of $T(s) = 1 - \gamma(s)$.

In the experiments, we sample γ directly from a Beta distribution $\mathrm{B}(\gamma; a, b)$ which is parameterized by two hyperparameters - a and b. By setting different values to a and b, we recover a rich set of transformations that are close to the commonly used cosine and sigmoid transformations (see Figure 2). For example, when $a = 1$ and $b = 1$, γ is sampled uniformly between $[0, 1]$; when $a, b > 1$, the mode is concentrated on $(a-1)/(a+b-2)$ and the bigger the total counts $a + b$ the smaller the variance; when $a < 1$ or $b < 1$, it's a bi-modal distribution that concentrates on 0 and 1. This formulation offers dual benefits: it reduces the number of hyperparameters and offers more interpretability to the model's preference on noise schedules.

Reverse Process The reverse process relies on a denoising model $p_\theta(x^{s-1}|x^s)$ to remove the noise added in the forward process. $p_\theta(x^{s-1}|x^s)$ is parameterized as a Gaussian distribution centered at μ_θ with a fixed variance. With the simplified objective proposed in Ho et al. (2020), the variance only affects the likelihood computation and can be ignored for training. The mean μ_θ can be formulated by either using the noise ϵ or the target x^0:

$$\mu_\theta = \frac{1}{\sqrt{\alpha_s}}x^s - \frac{1-\alpha_s}{\sqrt{\alpha_s(1-\gamma_s)}}\epsilon \tag{2}$$
$$= \frac{\sqrt{\alpha_s}(1-\gamma_{s-1})}{1-\gamma_s}x^s + \frac{\sqrt{\gamma_{s-1}}(1-\alpha_s)}{1-\gamma_t}x^0 \tag{3}$$

In Equation (2), the model learns to predict the noise $\hat{\epsilon}$; while in Equation (3), the model learns to predict the original image patch $\hat{x}^0$. We use the latter formulation and empirically show that predicting target works better.

Denote $g(x_t^s, z_t)$ the denoising patch decoder, where the conditioning $z_t = f(x_{<t})$ is the output of the backbone Transformer and x_t^s is the corrupted version of patch x_t, the simplified diffusion objective is:

$$\mathcal{L}_{\text{SIMP}}(x_t; x_{<t}, f, g) = \mathbb{E}_{\gamma \sim \mathcal{B}, \epsilon}\left[\|x_t^0 - g(x_t^{s(\gamma)}, f(x_{<t}))\|^2\right]$$

3.4. Rotary Positional Embedding for Images

While rotary positional embedding (RoPE) (Su et al., 2023) is widely used in NLP, its application in vision has been limited due to perceived accuracy/compute trade-offs and slower training compared to absolute or learnable positional encoding (Dosovitskiy et al., 2021). We extend RoPE to higher-dimensional positions by applying it separately to each dimension, addressing these limitations.

Let $m = (m_x, m_y)$ and $n = (n_x, n_y)$ represent the coordinates of two patches. In the self-attention layer, let $z_m^k = W_k x_m$ and $z_n^q = W_q x_n$ be the key and query projections for patches at locations m and n, respectively. The rotation matrix R is a block diagonal matrix with different frequencies for each coordinate (x or y). For a simplified case with 4 channels in z^k and z^q, the rotation matrix with one frequency per coordinate is:

$$R_\theta(m) = \begin{bmatrix} \cos 2\pi m_x\theta_x & -\sin 2\pi m_x\theta_x & 0 & 0 \\ \sin 2\pi m_x\theta_x & \cos 2\pi m_x\theta_x & 0 & 0 \\ 0 & 0 & \cos 2\pi m_y\theta_y & -\sin 2\pi m_y\theta_y \\ 0 & 0 & \sin 2\pi m_y\theta_y & \cos 2\pi m_y\theta_y \end{bmatrix}$$
$$k_m = R_\theta(m)z_m^k, \quad q_n = R_\theta(n)z_n^q$$
$$\langle k_m, q_n \rangle = (z_m^k)^T R_\theta(m)^T R_\theta(n) z_n^q$$

where θ_x and θ_y are fixed frequency components for x- and y-axis respectively. With the same principle, rotation matrix can be constructed for larger channel dimension.

The decomposed RoPE is easy to implement, requires minimal changes relative to 1D RoPE, and readily extends to higher-dimensional coordinates. However, splitting features

by coordinate dimensions reduces the frequency coverage per dimension. We did not encounter this issue with 2D data, but if it arises, consider projecting activations into a higher-dimensional space before applying the rotation matrix. The extension of RoPE described here has been independently proposed and implemented by the pytorch library rotary-embedding-torch.

4. Experiments

We evaluate the proposed approach (DARL) using both MSE and diffusion objectives. Our experiments explore the basic properties of the model and present ablations on training schedule and model scaling. We compare our results with other representation learning methods and show that DARL achieves performance close to state-of-the-art. Finally, we present results on transfer learning using the Visual Task Adaption Benchmark (VTAB, Zhai et al. (2020)).

Implementation Details We employ ViT backbone architecture with varying sizes and apply causal attention masking to preserve temporal dependencies. The ablations use ViT-L with patch size 16 pre-trained on the ImageNet dataset (Deng et al., 2009). The scaling experiment, in addition, uses ViT-B16 and ViT-H14. Relative positional encodings are applied through the decomposed 2D RoPE as described in Section 3.4. Unless otherwise stated, the patch decoder used for MSE loss is a linear layer; the denoisinng patch decoder is a single Transformer block with the output of the backbone and the embedding of corrupted patch as the inputs.

The input image has a resolution of 224×224. Spatial augmentation (random cropping, resizing and horizontal flipping) is applied during pre-training. The initialization scheme and optimization follow the MAE recipe (He et al., 2021). Pre-training uses AdamW (Loshchilov & Hutter, 2019) for optimization with learning rate using cosine decay and 40 epochs warm-up. The full list of hyper-parameters can be found in Table 8 in the Appendix.

Evaluation Protocol We use fine-tuning for evaluation and performance comparison. Instead of using special tokens or global mean pooling, we directly utilize the last patch's output as the global descriptor for downstream tasks. Unless stated otherwise, we fine-tune the model without causal attention masking. An ablation study of fine-tuning with or without causal attention masking is provided in Section 4.1.

For fine-tuning, we keep the image resolution of 224×224 pixels. For ImageNet, we apply random augmentation (Cubuk et al., 2019) which is also applied for supervised training baselines. For most ablation studies, we fine-tune the network for 50 epochs. To achieve a better performance in the training schedule and model scaling studies, we use an extended fine-tuning schedule of 90 epochs.

Due to the lack of bottleneck, the network learns a representation that is more distributed across the network. As a result, fine-tuning is a better evaluation protocol. However, in Appendix B.1, we provide a more in-depth study of linear evaluation results. We find that, although no explicit bottleneck is imposed, the best performing layer is situated roughly in the middle of the Transformer stacks.

4.1. Main Properties

Relative Positional Embedding We compare the decomposed RoPE (2D RoPE) with NoPE (Kazemnejad et al., 2023), absolute positional encodings and learnable positional embedding. NoPE doesn't apply any positional encodings. Absolute positional encodings uses the sine and cosine functions of various frequencies to encode the 2D position (Vaswani et al., 2023; He et al., 2021). Learnable positional embedding is randomly initialized and adapted with the training. In addition, we compare 2D RoPE which uses the inductive bias of spatial coordinates with the vanilla RoPE (1D RoPE). To evaluate or fine-tune models on a higher resolution, we interpolate the positions when using RoPE.

Table 1. **ImageNet top-1 accuracy of models with various positional encodings**

Posistion Encodings	Supervised	Supervised + Causal	Unsupervised + Causal
NoPE	80.3	80.2	81.9
Absolute	82.5	81.0	83.1
Learnable	81.9	80.7	83.2
1D RoPE	**82.8**	81.7	83.5
2D RoPE	82.7	**81.8**	**84.5**
Evaluate with resolution 384			
1D RoPE	52.0	36.4	84.2
2D RoPE	81.7	80.4	85.1

Comparison between no positional encoding (NoPE), 2D absolute positional encoding (Absolute), learnable positional embedding (Learnable), RoPE without the inductive bias for 2D coordinates (1D RoPE) and decomposed RoPE (2D RoPE) with supervised and unsupervised training. Models with supervised data is trained for 200 epochs and evaluated without exponential moving averages (EMA). Models with generative pre-training are trained with MSE loss for 400 epochs and fine-tuned for 50 epochs.

For supervised learning, the first two columns of Table 1 shows that RoPE, both 1D and 2D versions, outperforms other types of positional encodings with or without causal attention masking. This suggests that relative positional encodings are more effective for image data. Implementation details of the supervised baselines are in Appendix A.2. The improvement of performance is more prominent for models with causal attention masking than those without:

+0.8% (supervised) and +0.9% (unsupervised) for causal v.s. +0.2% for non-causal. This suggests that the positional encodings play a more important role in AR models.

2D RoPE has a much better inductive bias for spatial coordinates compared with the 1D counterpart. This reflects in the evaluation results when the resolution of the image changes from 224 to 384 (last two rows in Table 1). There is a significant drop in performance when scaling up the resolution for 1D RoPE in the supervised case, while the decrease in performance is much less for 2D RoPE. In addition, 2D RoPE performs much better (+1% compared to 1D RoPE) in the case of fine-tuning after unsupervised pre-training. This gap doesn't diminish when fine-tuned on a larger resolution.

Patch Decoder We compare between a linear patch decoder, a residual MLP and a Transformer patch decoder with varying depth. The Transformer patch decoder only applies for the denoising patch decoder: the patch decoding sequence consists of *two* tokens - the output of the backbone and the embedding of the noise corrupted patch, both at the same autoregressive step t. No causal attention masking is applied for the Transformer patch decoder. In addition to the architectural choices, we experiment with whether or not to condition on the diffusion step γ. When conditioned on γ, it's added as a channel to the noise corrupted image patch and linearly projected to the embedding space.

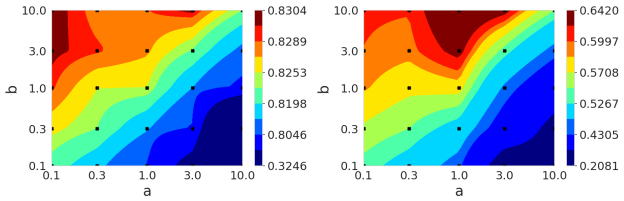
Table 2. **Top-1 accuracy of models pre-trained with different patch decoders and fine-tuned on ImageNet.**

		γ	Number of Layers				
Objective	Decoder	Cond	0	1	2	4	8
MSE	MLP	-	82.7	82.6	**82.8**	82.6	82.3
Diffusion	MLP	Yes	-	**82.7**	82.7	82.7	82.3
		No	-	82.6	**82.8**	82.7	82.4
	Transformer	Yes	82.9	82.8	83.0	82.8	**83.1**
		No	82.7	83.0	**83.1**	82.9	83.0

Models are pre-trained for 100 epochs and fine-tuned for 50 epochs. When number of layers is 0, the patch decoder is a single linear layers.

Table 2 suggests that, for MSE loss, a deeper patch decoder doesn't improve much of the fine-tuning performance. Therefore, we use the simple linear patch decoder when pre-training with MSE.

For the denoising patch decoder, deeper MLPs with or without conditioning on γ do not improve the performance. The Transformer decoder generally leads to better fine-tuning performance. When conditioning on γ, more transformer layers are beneficial; when not conditioning on γ, 1 or 2 Transformer blocks are sufficient to achieve the best performance. For rest of the experiments, we use the single-block Transformer as the denoising patch decoder.



(a) Patch Size 16 (b) Patch Size 56

Figure 3. **ImageNet top-1 accuracy of models pre-trained with different noise schedules.** Figure 3a and Figure 3b are trained with patch size 16 and 56 respectively. Models are pre-trained for 100 epochs and fine-tuned for 50 epochs. The colormap corresponds to threshold values of every 10 percentile, i.e. 10th, 20th, ..., 90th percentile. The x-axis and y-axis are hyperparameters a and b of the Beta distribution from which γ is sampled. The optimal noise schedule of ViT-L16 is biased toward extremely high noise levels, while ViT-L56 prefers a more balanced one.

Read-out Mechanism DARL learns a more distributed representation due to generative pre-training, and one question is, how these can be best used for downstream classification tasks? We investigate fine-tuning with or without the causal attention mask, as well as using mean pooling or last token output as the global image descriptor. When using the last token output as the descriptor, the input token is the last image patch which the model doesn't need during pre-training.

Table 3. **ImageNet top-1 accuracy with various read-outs.**

Masking	Mean Pool	Last Token
Causal	81.5	81.7
Non Causal	82.7	82.7

The base model is a ViT-L16 pre-trained with MSE loss for 100 epochs. Then the base model is fine-tuned for 50 epochs with different options for attention masking and global descriptor.

Table 3 suggests that model fine-tuned without the causal attention masking has a better classification performance. When causal attention masking is applied, it's preferable to use the last token as the global descriptor compared to mean pooling.

The 1% performance gap between causal and non-causal Transformers occurs both for supervised training from scratch (see Table 1) and fine-tuning from pre-trained models. We hypothesis that Transformer without causal attention masking has a better inductive bias for image data which, unlike text data, doesn't have an inherent ordering. Therefore, non-causal Transformer is able to make more efficient use of the model capacity. However, architectural improvements, such as relative position encodings, can mitigate this disadvantage to a certain degree.

Noise Schedule In diffusion models, the noise schedule determines which spatial frequencies are corrupted by noise.

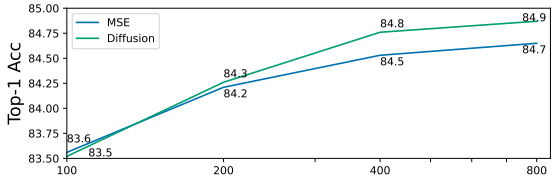


Figure 4. **ImageNet top-1 accuracy with varying training length.** Model trained with diffusion objective outperforms MSE with longer training schedules. Diffusion noise schedule is $a = 0.03$ and $b = 1$.

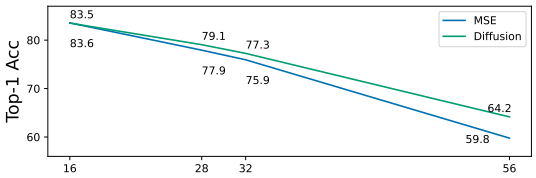


Figure 5. **ImageNet top-1 accuracy of model pre-trained with varying patch sizes.** Model trained with diffusion objective degrades more gracefully compared to MSE loss.

This means the noise schedule has a significant influence on the representation encoded in the model when pre-trained with the diffusion objective.

As described in Section 3.3, in our implementation, γ values are directly sampled from a Beta distribution. We study the effect of the noise schedule by varying the parameters a and b on a logarithmically spaced grid between 0.1 and 10. Figure 3a shows how the ImageNet top-1 accuracy varies with a and b. To achieve a better performance on ImageNet, the noise schedule would have to sample γ close to 0, i.e. heavily corrupted image patches. This explains why the MSE loss works well and diffusion yields only a marginal improvement. When more samples are drawn from the less noise-added region, the fine-tuning performance degrades rapidly. However, there is a set of parameters that work equally well with different Beta distribution profiles. For example, $a = 0.1, b = 10$ samples heavily from extremely noisy patches; $a = 3, b = 10$ peaks around $\gamma \approx 0.23$ with a relatively large variance.

Concurrent work (Chen et al., 2024) prefers a different noise schedule than ours, likely due to their focus on denoising as the pre-text task. Our framework combines autoregressive prediction with denoising, suggesting that autoregressive prediction is the primary driver of representation learning, with denoising providing additional benefits under specific conditions. Appendix B.5 provides more details.

Training Objective As shown in Equation (2) and Equation (3), diffusion models can be trained to predict either the additive noise ϵ or the original image x^0. We find that predicting the original image is better than predicting noise: for models pre-trained for 100 epochs and fine-tuned for 50 epochs, the ImageNet top-1 accuracy is 83.6% if the denoising patch decoder predicts the original image and 82.9% if it predicts the noise.

Table 5 suggests that the diffusion objective benefits from larger model capacity, likely because it learns features across all spatial frequencies. Figure 4 demonstrates that while models pre-trained for 100 epochs with MSE and diffusion objective perform similarly, extending the pre-training to 800 epochs reveals the diffusion objective's superiority.

Patch size We further investigate the influence of patch size for both objective functions by varying the patch size between 16, 28, 32 and 56. For each patch size, we sweep the noise schedule as described in Section 4.1 when training on diffusion objective.

While accuracy decreases for both objectives as patch size increases, Figure 5 reveals a gentler performance decline for the diffusion-based model compared to the MSE objective. Unlike the model pre-trained with patch size 16 (Figure 3a), when training with patch size 56, a more balanced noise schedule is preferred (see Figure 3b). In this schedule, the γ values are less concentrated on higher noise levels. Since the denoising patch decoder currently uses a single Transformer block, we speculate that adding more layers might further enhance the diffusion objective's performance. This is an area to explore for future works.

4.2. Comparison to Previous Results

We compare DARL to prior representation learning methods trained on ImageNet and using standard ViT architecture. We categorize these methods as: contrastive learning (with self-distillation), masked prediction, and generative pre-training. Due to limited results in generative pre-training, we include Image-GPT, despite its differing architecture.

Table 5. **ImageNet top-1 Accuracy Comparison**

		Backbone			
Category	Method	ViT-B	ViT-L	ViT-H	Others
Contrastive	DINO [1]	82.8	-	-	-
	MoCo v3 [2]	83.2	84.1	-	-
Masked Pred.	BeiT [3†]	83.2	85.2	-	-
	MAE [4]	83.6	85.9	86.9	-
Generative	Image-GPT [5]	-	-	-	72.6
	DARL (MSE)	82.7	84.7	85.5	-
	DARL (DIFF)	81.9	84.9	85.9	-

[1] Caron et al. (2021) [2] He et al. (2019) [3] Bao et al. (2022)
[†] Tokenizer is trained on a much larger custom dataset (Ramesh et al., 2021). [4] He et al. (2021) [5] Chen et al. (2020a)
Our method is pre-trained for 800 epochs and fine-tuned for 90 epochs. Noise schedule used for diffusion objective is $a = 0.03$ and $b = 1$.

Table 4. **Top-1 Accuracy on VTAB Benchmark**

	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Mean	Camelyon	EuroSAT	Resisc45	Retinopathy	Mean	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Elev	Mean
Supervised [1]	86.2	91.0	69.5	91.4	**93.0**	94.9	75.3	85.9	81.0	98.7	93.8	81.6	88.8	**94.3**	88.3	63.9	**81.3**	**98.5**	**85.1**	25.3	51.2	**73.5**
DARL (MSE)	87.6	85.2	**79.1**	97.8	92.4	97.5	79.2	88.4	**89.9**	**98.8**	97.3	73.6	89.9	93.7	90.2	76.3	45.5	37.2	48.7	30.3	95.7	64.7
DARL (DIFF)	**87.9**	**87.7**	77.7	**98.5**	91.3	**97.7**	**79.9**	**88.7**	89.2	**98.8**	**97.4**	**83.5**	**92.2**	93.9	**90.5**	**78.5**	39.2	39.7	48.8	**31.4**	**96.0**	64.7

[1] Steiner et al. (2022). ViT-L16 backbone is used for all experiments. Models are pre-trained on ImageNet dataset.

As shown in Table 5, DARL achieves significantly better results in generative pre-training than previous approaches, surpassing i-GPT which uses a much larger model by a large margin. We also perform better than contrastive methods when it comes to fine-tuning performance. Despite a small performance gap compared to BeiT and MAE, DARL achieve results that are highly comparable to the current state-of-the-art masked prediction models. Based on previous comparison between causal and non-causal Transformers, we hypothesis that this gap could also due to the different inductive bias resulting from applying causal attention masking rather than the prediction task itself.

4.3. Transfer Experiments

Classification. To investigate the transfer performance, we fine-tune our models on diverse downstream tasks. We use the VTAB benchmark (Zhai et al., 2020) which consists 19 classification tasks. The tasks are divided into 3 groups - *Natural*, *Spcialized* and *Structured* - representing different distribution shifts from the pre-training dataset.

Table 4 repports the top-1 accuracy by dataset and category. Details of the hyperparameter selection and evaluation procedure are described in Appendix A.4. In the *Natural* and *Specialized* categories, DARL demonstrates decent performance gains over supervised pre-training, improving results on 10 out of 11 datasets. While the *Structured* category presents challenges, particularly with Kitti-Distance and dSprites, this highlights areas for future research and development.

Object detection and segmentation. To assess the transfer capability of DARL in tasks other than classification, we fine-tune ViTDet (Li et al., 2022) with Cascade Mask R-CNN (Cai & Vasconcelos, 2017) as detector heads. Other implementations details are in Appendix A.3.

Table 6. **COCO Object Detection and Segmentation**

	SUP	MAE †	DARL (DIFF)
AP	54.1	57.2	56.4
mAP	45.8	48.6	48.0

ViT-L16 backbone fine-tuned with ViTDet architecture and Cascade Mask R-CNN detectors. †MAE results are reproduced from our own codebase.

Table 6 confirms that, similar to ImageNet classification, DARL outperforms the supervised pre-training baseline and performs closely to the strong MAE baseline. ViTDet architecture uses a simple feature pyramid design which takes only the last layer activation of the Transformer. This design was developed with supervisely trained models and encoder-only representations (bottleneck at the last layer of the Transformer stack). It could be sub-optimal for generative pre-trained models such as ours. We leave this for future research endeavors.

4.4. Studies on Patch Ordering

While autoregressive modelling works seamlessly for sequences like language or audio, the optimal token ordering for images remains unclear. It's also tempting to assume that random ordering leads to better model performance. Our research explores two key questions: 1. Does one fixed image token ordering outperform others? 2. Can models trained on randomly ordered patches achieve superior results?

Fixed Ordering We develop two fixed ordering strategies and compare their performance to standard raster order. For both strategies, first, we divide the image into fixed-size and non-overlapping blocks. Next,

1. Nested Raster Order: Order the blocks in raster order; Divide the blocks into patches and, again, order the patches within each block in raster order.
2. Round-Robin Order: Divide the blocks into raster ordered patches; Starting from the first block, one patch is selected from each block. The process cycles through the blocks repeatedly.

For both strategies, we experiment with 2×2, 4×4 and 8×8 patches per block. Visualization of orderings are in Appendix C.

Table 7. **Comparison of different fixed order strategies**

Raster	Nested Raster			Round-Robin		
	2×2	4×4	8×8	2×2	4×4	8×8
83.3	83.3	83.4	83.3	83.0	82.7	82.6

All models are trained with MSE loss for 100 epochs and fine-tuned for 50 epochs. Image resolution is 256×256. Block sizes are 2×2, 4×4 and 8×8 patches per block.

Table 7 suggests that raster ordering is close to optimal. This observation is consistent with concurrent work from El-Nouby et al. (2024). The round-robin pattern yields lower performance compared to raster order, whereas nested raster order achieves similar or even better results. We provide more visualization of the results in Appendix C.

Random Ordering Ablating random ordering requires architectural changes, as the model relies on query tokens to guide its patch prediction. We use the XLNet architecture proposed in Yang et al. (2020). Two-stream architecture which consists of a query stream and a content stream. The query stream can integrate information from previous query and content tokens, as well as the current query. However, it cannot see the current content token. This is achieved by a special attention masking scheme (see details in Appendix A.5 and Yang et al. (2020)). The content stream uses the causal attention masking and operates exactly the same way as a decoder-only Transformer. In the experiment, we use learnable query tokens and 2D RoPE for positional encodings.

We experiment raster scan order and randomly permuted patch order on the two-stream architecture. Figure 6 reveals two insights: 1. Random ordering requires longer training to match the performance of fixed ordering; 2. Contrary to common belief, random ordering does **not** ultimately offer any performance advantage.

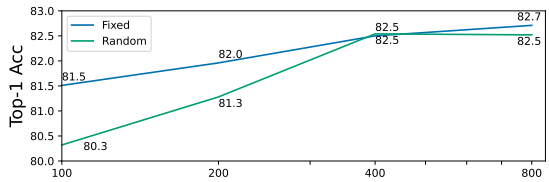


Figure 6. **ImageNet top-1 accuracy of raster order and random order.** XLNet style two-stream Transformer with ViT-L16 backbone is used for both fixed and random ordering. MSE loss is used for pre-training. Models are fine-tuned for 50 epochs with fixed ordering.

5. Discussion and Limitations

We development, DARL, a model unifying visual representation learning and image generation, leveraging the power of autoregressive and denoising diffusion models. Our approach demonstrates that generative pre-training in the vision domain achieves similar performance to state-of-the-art masked prediction models, with a minor 1% difference. This observation is consistent with studies from the language domain, where encoder-decoder models achieve a slightly better result than decoder-only models (Tay et al., 2023; Fu et al., 2023).

DARL offers two advantages: generative capability and a likelihood-based training objective. We equip our model with a denoising patch decoder to generate multi-modal patch distributions, enhancing its generative potential (see Figure 7 for samples from DARL) and aligning it with likelihood-based training. However, our study also suggests that image generation and learning higher-level abstractions may be conflicting. Competition for the model capacity forces prioritization of different aspects. The distinct preference for noise schedules is a manifestation of this competition. Future works could investigate whether scaling to larger models helps alleviate the capacity constraints and improve performance on both tasks. Overall, we believe there is significant room for improvements to fully realize the benefits of this generative pre-training approach.

Figure 7. **Samples from DARL conditioned on the top half of the image.** Resolution is 64×64. The first column (in the red rectangle) is the original image in ImageNet validation set. The rest of the columns are samples generated conditioned on the top half of the image. Details of the model architecture and training are described in Appendix F. Samples presented here are cherry-picked. For more samples, please see Figure 15 in the Appendix.

Acknowledgements

We are grateful to David Fleet, Lala Li, Saurabh Saxena, Priyank Jaini, Olivier Henaff and Joao Carreira for their helpful discussions. Special thanks go to Jovana Mitrovic for her careful review of the paper and to Aaron van den Oord for his unwavering support.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which are specific to contributions of our paper. We explore the synergy of image generation and representation learning. Image generation, however, raises important ethical concerns about the potential for creating misleading or harmful content. Its use could amplify the spread of fake images and exacerbate issues of disinformation. Another potential concern is fairness of the algorithm which is heavily influenced by dataset bias. Being a pre-training method, the presence of bias in the dataset could be carried over to downstream tasks resulting in a wider propagation of the negative effects.

References

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views, 2019.

Bao, H., Dong, L., Piao, S., and Wei, F. Beit: Bert pre-training of image transformers, 2022.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020.

Cai, Z. and Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection, 2017.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers, 2021.

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

Chen, T. On the importance of noise scheduling for diffusion models, 2023.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020b.

Chen, X., Liu, Z., Xie, S., and He, K. Deconstructing denoising diffusion models for self-supervised learning, 2024.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

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

Dieleman, S. Perspectives on diffusion, 2023. URL https://sander.ai/2023/07/20/perspectives.html.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015. URL http://arxiv.org/abs/1505.05192.

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

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

El-Nouby, A., Klein, M., Zhai, S., Bautista, M. A., Toshev, A., Shankar, V., Susskind, J. M., and Joulin, A. Scalable pre-training of large autoregressive image models, 2024.

Fu, Z., Lam, W., Yu, Q., So, A. M.-C., Hu, S., Liu, Z., and Collier, N. Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder, 2023.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. Made: Masked autoencoder for distribution estimation, 2015.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations, 2018.

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

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners, 2021.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Ho, J. and Salimans, T. Classifier-free diffusion guidance, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models, 2022.

Hoogeboom, E., Heek, J., and Salimans, T. simple diffusion: End-to-end diffusion for high resolution images. *arXiv preprint arXiv:2301.11093*, 2023.

Hudson, D. A., Zoran, D., Malinowski, M., Lampinen, A. K., Jaegle, A., McClelland, J. L., Matthey, L., Hill, F., and Lerchner, A. Soda: Bottleneck diffusion models for representation learning, 2023.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models, 2020.

Kazemnejad, A., Padhi, I., Ramamurthy, K. N., Das, P., and Reddy, S. The impact of positional encoding on length generalization in transformers, 2023.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes, 2022.

Li, Y., Pogodin, R., Sutherland, D. J., and Gretton, A. Self-supervised learning with kernel dependence maximization, 2021.

Li, Y., Mao, H., Girshick, R., and He, K. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pp. 280–296. Springer, 2022.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization, 2019.

Nichol, A. and Dhariwal, P. Improved denoising diffusion probabilistic models, 2021.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models, 2022.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles, 2017.

Parmar, N., Vaswani, A., Uszkoreit, J., Łukasz Kaiser, Shazeer, N., Ku, A., and Tran, D. Image transformer, 2018.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation, 2021.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models, 2014.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2022.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

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

Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. How to train your vit? data, augmentation, and regularization in vision transformers, 2022.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding, 2023.

Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., Chung, H. W., Shakeri, S., Bahri, D., Schuster, T., Zheng, H. S., Zhou, D., Houlsby, N., and Metzler, D. Ul2: Unifying language learning paradigms, 2023.

Uria, B., Côté, M.-A., Gregor, K., Murray, I., and Larochelle, H. Neural autoregressive distribution estimation, 2016.

van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks, 2016a.

van den Oord, A., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K. Conditional image generation with pixelcnn decoders, 2016b.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning, 2018.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023.

Wei, C., Mangalam, K., Huang, P.-Y., Li, Y., Fan, H., Xu, H., Wang, H., Xie, C., Yuille, A., and Feichtenhofer, C. Diffusion models as masked autoencoders, 2023.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.

Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A. S., Neumann, M., Dosovitskiy, A., Beyer, L., Bachem, O., Tschannen, M., Michalski, M., Bousquet, O., Gelly, S., and Houlsby, N. A large-scale study of representation learning with the visual task adaptation benchmark, 2020.

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization, 2016.

A. Implementation Details

A.1. Hyperparameters

Table 8. ImageNet Pre-training Hyperparams.

Name	Value
Initialization	Xavier Uniform
Drop path	0.0
Positional encoding	2D RoPE
Augmentation	Spatial
Optimizer	AdamW
Optimizer momentum	β_1=0.9, β_2=0.95
Base learning rate	1.5e-4
Learning rate scaling †	True
Weight decay	0.05
Batch size	4096
Learning rate schedule	Cosine
Warm-up epochs	40

† $\text{total_lr} = \text{base_lr} \times \text{batch_size}/256$

Table 9. ImageNet Fine-tuning Hyperparameters

Name	Value			
Positional encoding	2D RoPE			
Augmentation	RandAug (9, 0.5)			
Mixup	0.8			
Cutmix	1.0			
Label smoothing	0.1			
Optimizer	AdamW			
Optimizer momentum	β_1=0.9, β_2=0.95			
Layer-wise lr decay	0.75			
Weight decay	0.05			
Batch size	4096			
Learning rate schedule	Cosine			
Warm-up epochs	5			
	B	L		H
Training epochs	90	50	90	90
Learning rate	1e-3	1e-3	1e-3	3e-3
Drop path	0.0	0.0	0.1	0.2

A.2. Supervised Baseline

The supervised baseline is similar to the ViT (Dosovitskiy et al., 2021) implementation with the following differences:

1. Initialization and optimization scheme: the initialization and optimization scheme porposed in He et al. (2021) are used.
2. Positional encodings: positional encodings varied by the ablations.
3. Attention masking: causal and non-causal attention masking applied as required by the ablations.

4. Classfication head: global mean pooling is as global image descriptor and classification head is a single linear layer.

Table 10. ImageNet Supervised Training Hyperparameters

Name	Value
Initialization	Xavier Uniform
Drop path	0.2
Augmentation	RandAug (9, 0.5)
Mixup	0.8
Cutmix	1.0
Label smoothing	0.1
Optimizer	AdamW
Optimizer momentum	β_1=0.9, β_2=0.95
Base learning rate	1e-4
Learning rate scaling	True
Weight decay	0.3
Batch size	4096
Learning rate schedule	Cosine
Warm-up epochs	20
Training epochs	200

A.3. COCO Object Detection and Segmentation

We simply initialize the weight of the ViT-L16 backbone with weight after supervised, MAE and DARL pre-training. Our implementation follows that of ViTDet in Li et al. (2022) and Cascade Mask R-CNN in Cai & Vasconcelos (2017). Hyperparameters used for DARL pre-trained with diffusion loss are listed in Table 11.

Table 11. Hyperparameters for COCO Experiments

Name	Value
Resolution	1024 x 1024
Drop path	0.4
Augmentation	panoptic deeplab policy †
Random scaling	[0.1, 2.5]
Learning rate schedule	Cosine
Warm-up steps	256
Training steps	73920
Batch size	64
Optimizer	AdamW
Optimizer momentum	β_1=0.9, β_2=0.999
Learning rate	7.5e-4 (DARL); 5e-5 (MAE/SUP)
Weight decay	0.1
Layer-wise lr decay	0.85 (DARL); 0.8 (MAE/SUP)
Exponential moving average decay	0.9998

†See details at https://github.com/tensorflow/models/blob/master/official/vision/ops/augment.py#L2301.

A.4. VTAB

Following the procedure in Zhai et al. (2020), we first conduct a hyperparameter sweep (see Table 12) using train and validation sets. We then select the best-performing hyperparameters based on validation results, retrain the model on the combined train+validation set, and report the final test set performance.

Table 12. Hyperparameters for VTAB Experiments

Name	Value
Label smoothing	0.1
Optimizer	AdamW
Optimizer momentum	β_1=0.9, β_2=0.95
Batch size	256
Learning rate schedule	Cosine
Warm-up epochs	3000
Training steps	20000
Attention Masking	Causal
Hyperparameter Sweep	
Drop path	{0.0, 0.1, 0.2}
Augmentation	{ Spatial, RandAug (9, 0.5) + Mixup 0.8 + Cutmix 1.0 }
Weight decay	{ 0.1, 0.3 }
Learning rate	{ 1e-3, 1e-2, 1e-1 }
Layer-wise lr decay	{ 0.7, 0.8, 0.9 }

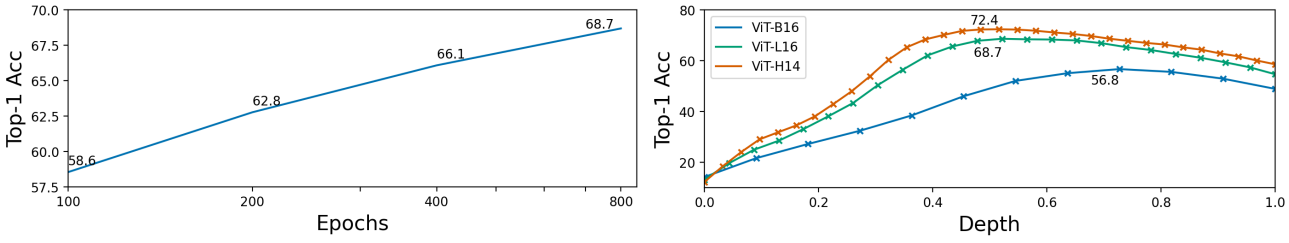
A.5. Two-Stream Architecture

The architecture is similar to the two-stream architecture in XLNet. We provide a brief description here, for more details see Yang et al. (2020). The network has two sets of inputs: one sequence for the query stream and the other for the content stream. The content stream operates exactly the same as a regular AR Transformer. The query stream inputs can be considered as the query associated to a particular token. For example, in our case, we use the location of the next patch in the sequence as the input of the query stream. When the patch ordering is randomly permuted during training, this gives indication to the model of which patch should be predicted. The attention layers compute an additional query head which use the input of the query stream. They also carry out another attention operation by using the additional query and the content stream keys and values. This attention operation employs a different attention masking, where the diagonals in mask matrix are zeroed out. This ensures that the query stream doesn't attend to the current patch content.

In XLNet, random permutation of the token ordering is achieved by sampling attention masking. In this case, the ordering of the input tokens are kept the same. Our implementation directly feeds the permuted token sequence as input and keeps the attention masking the same for all the permutation. We share the weights of the feed-forward layers for the two streams. 2D RoPE is applied for the positional encodings.

B. Further Results and Discussions

B.1. Linear Evaluation



(a) Training Epochs

(b) Network Depth

Figure 8. ImageNet top-1 accuracy with linear evaluation protocol. a) Linear performance increases with longer training schedule. Models are ViT-L16 pre-trained with diffusion objective. b) Linear performance varies with the layer depth. Layer depth is normalized to [0, 1]. Models are pre-trained with diffusion objective.

Hyperparameters for linear evaluation are listed in Table 13. Causal attention masking is applied for linear evaluation. Global mean pooling is used to get the global image descriptor. We apply batch norm on the features extracted from the backbone, similar to He et al. (2021).

Due to the lack of bottleneck, linear evaluation performance in decoder-only Transformer is, in general, lower than contrastive. Longer training schedule and larger models can both improve the linear performance.Figure 8b suggests that the linear interpretability of the features increases with layer depth, peaking roughly in the middle of the network. For model with limited capacity (e.g. ViT-B16), the network prioritizes encoding, pushing the best performing feature deeper into the stack.

Table 13. ImageNet Linear Evaluation Hyperparameters.

Name	Value
Drop path	0.0
Positional encoding	2D RoPE
Augmentation	Spatial
Label smoothing	0.1
Optimizer	LARS
Optimizer momentum	0.9
Learning rate	0.1
Weight decay	0.0
Batch size	16384
Learning rate schedule	Cosine
Warm-up epochs	10
Training epochs	90

B.2. Normalized Pixels

He et al. (2021) introduced normalized pixels as target of the reconstruction. More precisely, instead of using the raw or standardized pixels, they use pixel values normalized by the mean and standard deviation of the patch statistics as the target. He et al. (2021) and Wei et al. (2023) find that using the patch normalized pixel value improves the quality of the representation. We experiment normalized pixels with DARL + MSE loss. However, Table 14 suggests using normalized pixels doesn't improve the performance in our case.

Table 14. ImageNet Top-1 Accuracy using Normalized Pixels as Target

	Architecture		
Target	ViT-B16	ViT-L16	ViT-H14
Pixels	82.7	84.7	85.5
Norm. Pixels	82.4	84.5	85.5

All the models are pre-trained for 800 epochs and fine-tuned for 90 epochs.

B.3. Scaling Behavior

While we don't have a systematic study of models with different sizes beyond the standard ViT family, for the standard model sizes we trained, we observe a correlation between validation loss during pre-training and downstream performance for both MSE and diffusion loss of the same noise schedule (see Table 15).

While there is not enough datapoints to make a statistically significant analysis for the moment, we believe this is a viable direction for future research enabled by our framework.

Table 15. Validation Loss v.s. ImageNet Accuracy

MSE		DIFF †	
Loss	Top-1 Acc (%)	Loss	Top-1 Acc (%)
0.194	83.6	0.179	83.5
0.189	84.2	0.174	84.3
0.185	84.5	0.171	84.8
0.182	84.7	0.168	84.9
0.158	85.5	0.146	85.9

†Diffusion loss with beta distribution noise schedule of a=0.03, b=1.

B.4. Tokenization

One area to further explore is the use of a tokenizer. We provide some preliminary results using a discrete VAE encoder (same tokenizer as in Dall-E and BeiT). The tokenizer serves two functions: encoding the image patch to form the inputs of the Transformer; serving as a target for the outputs of the Transformer.

Table 16. ImageNet Top-1 Accuracy with dVAE Tokenzier

Supervised		Unsupervised			
Default	dVAE inputs	MSE	Denoising	dVAE input+MSE	dVAE target
82.7	75.9	83.6	83.5	62.9	82.9

Supervised uses *non-causal* Transformer. The default model uses linear patch embedding.
Unsupervised uses DARL pre-trained for 100 epochs. The default model uses linear patch embedding and denosing patch decoder.

Table 16 shows the top-1 accuracy on ImageNet. Supervised model is ViT-L16 without causal attention masking and uses 2D RoPE. Unsupervised model is DARL with ViT-L16 backbone pre-trained 100 epochs. Both MSE and denoising objectives perform similarly. We can see that using dVAE to encode the image patches has a significant negative impact on model performance, both in supervised and unsupervised. However, dVAE as the target is only slightly worse than MSE or denoising. It worth further investigating for longer training schedule and evaluate on other downstream tasks.

B.5. Discussion on Noise Schedule

Figure 9 compares our schedule with both the original DDPM schedule and the linear decay schedule proposed by Chen et al. (2024). We could see that DDPM schedule samples 80% from values [0.1, 0.6] and with a slight peak around 0.3. Linear decay schedule samples mostly from very small noise regions. Optimal schedule for DARL samples mainly from high noise regions. As we discussed in the main text, the difference in the preferred noise schedule between l-DAE and our work can be understood as due to the different tasks imposed by the framework. The representation learning task of l-DAE

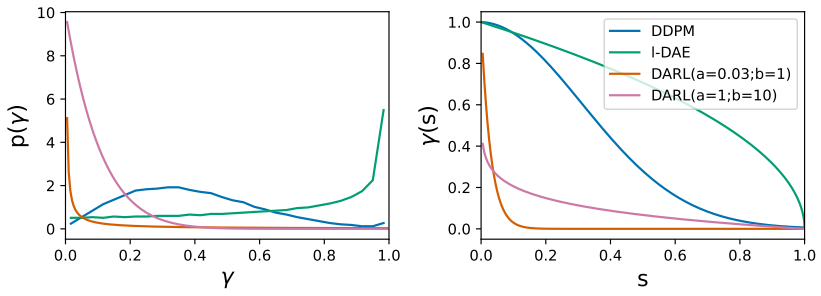


Figure 9. **Comparison of noise schedules between DDPM, l-DAE and DARL.**

is denoising. Therefore, totally destroying the image is problematic. In our case, we combine the autoregressive prediction with denoising.

C. Visualizations for Patch Ordering

Figure 10 compares the patch reconstruction error at each postion in the sequence between different fixed order strategies. Figure 11 and Figure 12 show the visualization and the average per-patch reconstruction errors for nested raster order. Figure 13 and Figure 14 show the visualization and the average per-patch reconstruction errors for round-robin order. See Section 4.4 for details.

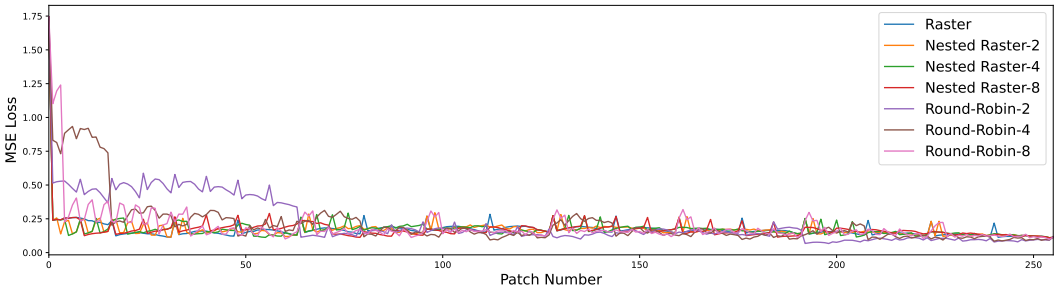
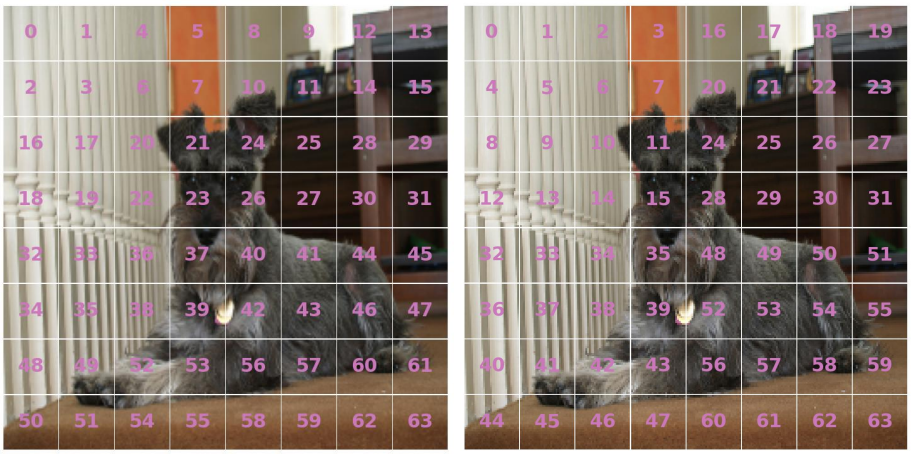


Figure 10. **Reconstruction error by patch number**



(a) Block Size 2 (b) Block Size 4

Figure 11. **Visualization of nested raster order**

D. Rotary Position Embedding

For a 1D sequence, denote the previous layer activation x_m and x_n at position m and n respectively. $\mathbf{X} = [x_1, x_2, \ldots, x_T]$ is a matrix of $T \times d$. Let $\mathbf{Z}^k = \mathbf{X}W^k$ and $\mathbf{Z}^q = \mathbf{X}W^q$ be the projections to key and query. For each location m, we can use two channel dimensions of $\mathbf{Z}_m^{k,q}$ to represent a complex number $z_{m,\theta_i}^{k,q} = [\mathbf{Z}_{m,2i}^{k,q}, \mathbf{Z}_{m,2i+1}^{k,q}] = \mathbf{Z}_{m,2i}^{k,q} + i\mathbf{Z}_{m,2i+1}^{k,q}$ and associate it with a frequency component θ_i. Denote the rotary matrix $R_\theta(m) = \begin{bmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{bmatrix}$. For each frequency

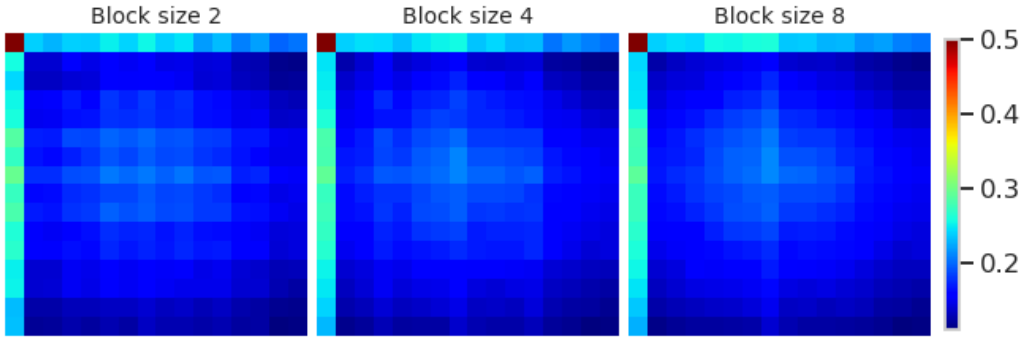
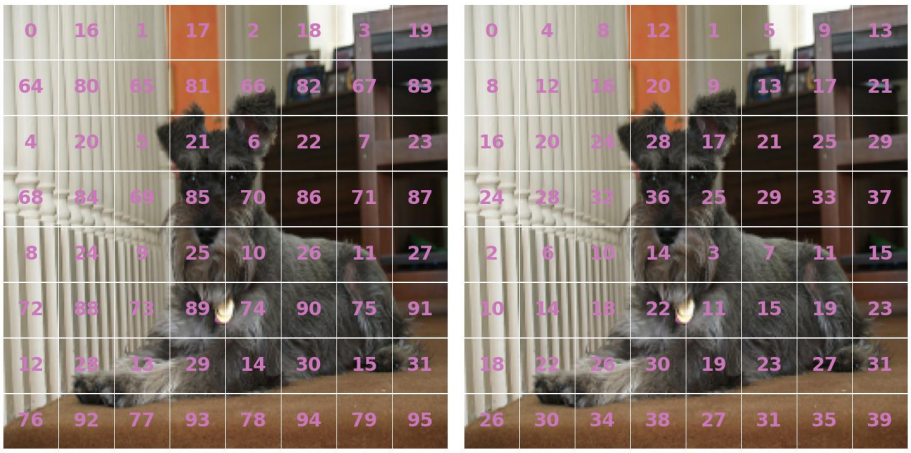


Figure 12. **Reconstruction error for nested raster order**



(a) Block Size 2 (b) Block Size 4

Figure 13. **Visualization of round-robin order**

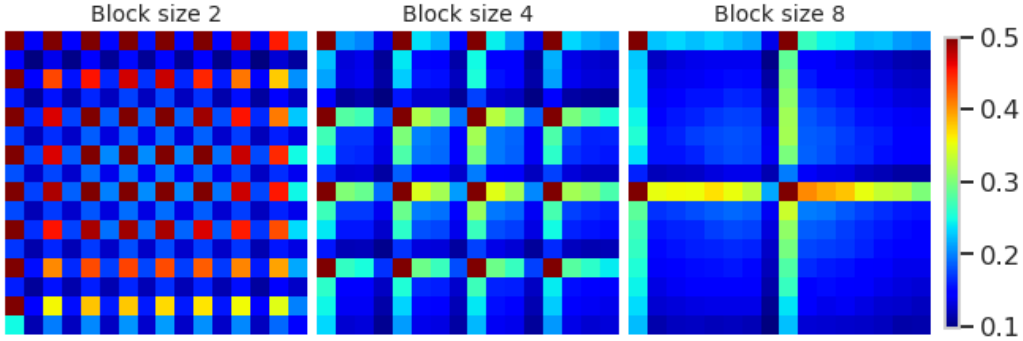


Figure 14. **Reconstruction error for round-robin order**

component θ_i, rotary position embedding is formulated as follows:

$$
\begin{aligned}
k_{m,\theta_i} &= z^k_{m,\theta_i} e^{im\theta_i} = R_{\theta_i}(m) z^k_{m,\theta_i} \\
q_{n,\theta_i} &= z^q_{n,\theta_i} e^{in\theta_i} = R_{\theta_i}(n) z^q_{n,\theta_i} \\
a_{nm,\theta_i} &= \langle k_{m,\theta_i}, q_{n,\theta_i} \rangle = \mathrm{Re}[z^k_{m,\theta_i} (z^q_{n,\theta_i})^* e^{i(n-m)\theta_i}] \\
&= (z^k_{m,\theta_i})^T R_{\theta_i}(m)^T R_{\theta_i}(n) z^q_{n,\theta_i}
\end{aligned}
$$

$a_{nm} = \sum_{\theta_i} a_{nm,\theta_i}$ is the pre-attention before applying the softmax function. The number of frequency components θ_i is half of the number of channels of the attention head, ranging between $[1, \text{max_wavelength}]$ i.e. $\theta_i = \text{max_wavelength}^{2i/d}$.

D.1. ND Rotary Position Embedding

A straightforward way to extend the 1D RoPE to ND is to randomly sample frequencies for the N dimensions. Denote $m = [m_x, m_y]$ a 2D vector with the location of x-axis and y-axis on each dimension. The rotary matrix $R_\theta(m) = \begin{bmatrix} \cos m^T\theta & -\sin m^T\theta \\ \sin m^T\theta & \cos m^T\theta \end{bmatrix}$, with $\theta = [\theta_x, \theta_y]$ frequency components for x-axis and y-axis respectively. The rotary position embedding can be formulated in a similar manner to the 1D case just with $m = [m_x, m_y]$, $n = [n_x, n_y]$, $\theta = [\theta_x, \theta_y]$ being 2D vectors.

The problem with this formulation is that the frequency components is combinatorial to the number of dimensions of m. Because the number of frequencies is associated to the number of channels of the attention head, this significantly reduces the range of frequencies for each dimension. By decomposing along the axes as in Section 3.4, we can use less frequency components to cover the N dimensions.

E. Diffusion

We use the mathematical framework established by DDPM (Ho et al., 2020). Many of the analysis can be found in DDPM paper. We present here for the completeness of our analysis. To simplify notation, we use x as the image or individual patch and subscript t for the diffusion process.

First, the variational bound can be expressed as follows:

$$\begin{aligned}
\log p(x) &= \log \int p(x_{0:T}) \mathrm{d}x_{1:T} \\
&= \log \mathbb{E}_{q(x_{1:T}|x_0)} \left[p(x_{0:T}) / q(x_{1:T}|x_0) \right] \\
&\geq \mathbb{E}_{q(x_{1:T}|x_0)} \left[\log \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} \right] \\
&= \mathbb{E}_{q(x_{1:T}|x_0)} \left[\sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} + \log p(x_T) \right]
\end{aligned}$$

Notice that the forward process is Markov, that means $q(x_t|x_{t-1}) = q(x_t|x_{t-1}, x_0)$. Using the Bayes rule, we have

$$q(x_t|x_{t-1}, x_0) = \frac{q(x_t, x_{t-1}|x_0)}{q(x_{t-1}|x_0)} = \frac{q(x_{t-1}|x_t, x_0) q(x_t|x_0)}{q(x_{t-1}|x_0)}$$

This is also a Gaussian distribution:

$$q(x_{t-1}|x_t, x_0) \sim \mathcal{N}(\mu_q, \Sigma_q), \quad \mu_q = \frac{\sqrt{\alpha_t}(1-\gamma_{t-1})}{1-\gamma_t} x_t + \frac{\sqrt{\gamma_{t-1}}(1-\alpha_t)}{1-\gamma_t} x_0, \quad \Sigma_q = \frac{(1-\alpha_t)(1-\gamma_{t-1})}{1-\gamma_t} I \tag{4}$$

The variational objective can, then, be written as

$$
\begin{aligned}
\mathcal{L} =& \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1}, x_0)}\right] - H\left[p(x_T)\right] \\
=& \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}\right] \\
&+ \sum_{t=2}^{T} \mathbb{E}_{q(x_{t-1}|x_0)}\left[\log q(x_{t-1}|x_0)\right] - \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}\left[\log q(x_t|x_0)\right] - H[p(x_T)] \\
=& \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right] + \sum_{t=2}^{T} \int q(x_t|x_0) q(x_{t-1}|x_t, x_0) \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} \mathrm{d}x_t \mathrm{d}x_{t-1} \\
&- H[q(x_1|x_0)] + H[q(x_T|x_0)] - H[p(x_T)] \\
=& \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log p_\theta(x_0|x_1)\right] - \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}\left[KL[q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)] + H[q(x_T|x_0)] - H[p(x_T)]\right.
\end{aligned}
$$

We can reparameterize $p_\theta(x_{t-1}|x_t)$ by modelling different parts. Note that from the forward process, we have $x_t = \sqrt{\gamma_t} x_0 + \sqrt{1-\gamma_t}\epsilon$, $x_0 = \frac{1}{\sqrt{\gamma_t}}(x_t - \sqrt{1-\gamma_t}\epsilon)$. From Equation (4), we want to model mu_q and it can be written as:

$$
\begin{aligned}
\mu_q &= \frac{\sqrt{\alpha_t}(1-\gamma_{t-1})}{1-\gamma_t} x_t + \frac{\sqrt{\gamma_{t-1}}(1-\alpha_t)}{1-\gamma_t} x_\theta \qquad (5) \\
&= \frac{\sqrt{\alpha_t}(1-\gamma_{t-1})}{1-\gamma_t} x_t + \frac{\sqrt{\gamma_{t-1}}(1-\alpha_t)}{1-\gamma_t} \frac{1}{\sqrt{\gamma_t}}(x_t - \sqrt{1-\gamma_t}\epsilon_\theta) \\
&= \frac{1}{\sqrt{\alpha_t}} x_t - \frac{1-\alpha_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}} \epsilon_\theta \qquad (6)
\end{aligned}
$$

Equation (6) is the DDPM formulation for predicting noise. Our model uses the formulation Equation (5) to build a predictor for the original image.

F. Samples from DARL

To generate samples from DARL, we train it on ImageNet for 800 epochs. We use a resolution 64 and patch size 8, so our backbone is a ViT-L8. The denoising patch decoder is a 8-layer Transformer with conditioning on gamma. During training, we employ noise schedule $\gamma \sim \text{Beta}(1, 1)$, that is uniformly sampled between $[0, 1]$. The prediction target is still the original image patches. During sampling, we use step size 0.001, that is $T = 1000$. Figure 15 shows samples from the model conditioned on the top half of the original images.

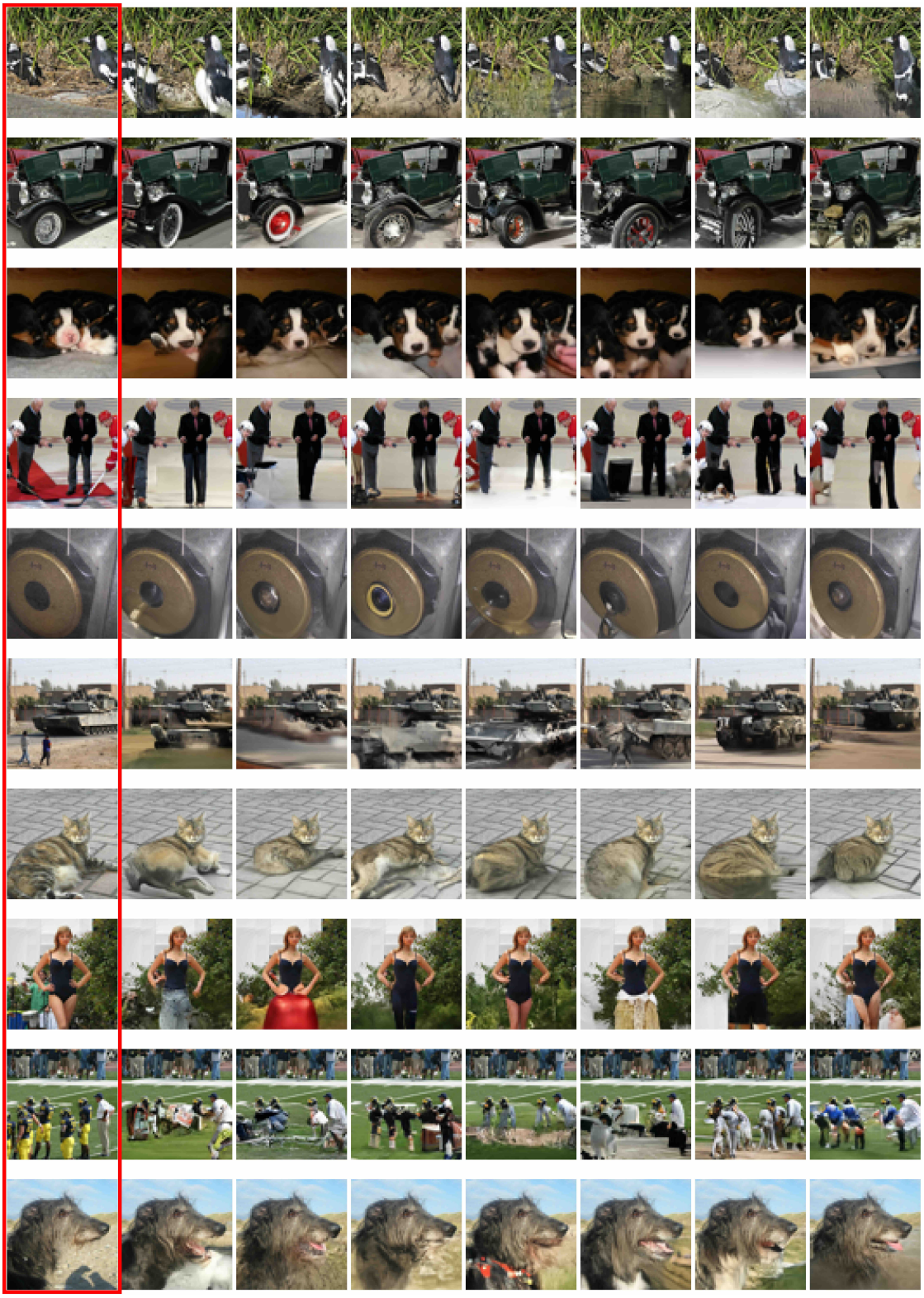


Figure 15. **Samples from DARL.** The first image in each row (in the red rectangle) is the original image in ImageNet validation set. The rest of the images are samples generated conditioned on the top half of the image.