EMBEDDING PREDICTION HELPS IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In diffusion transformers, a class label or a text prompt is embedded once, and the same condition is reused at every denoising step. We ask whether predicted embeddings can serve as this condition instead. Next-Embedding Predictive Autoregression (NEPA) trains a Transformer to predict the next continuous embedding in a sequence. In generation, the clean image follows the noisy image, so its embeddings are the next embeddings after the condition and the noisy image. We train a NEPA model to predict them all at once with **Multi-Embedding Prediction**, and in **Embedding Conditioned Generation**, a DiT generator is conditioned on these predictions, recomputed at every denoising step, so the conditioning signal adapts to the current noisy state. Experiments on class-conditional ImageNet 256×256 study the condition of the generator, the design of Multi-Embedding Prediction, and the scaling of both models. The NEPA model adds a second network to every sampling step; with it, and combined with REPA, our final model, **NEPA-DiT-XL**, reaches an FID of **1.32** using about a third of the training compute of REPA.

1 INTRODUCTION

Diffusion transformers have become the standard backbone for image generation (Peebles & Xie, 2023; Ma et al., 2024; Esser et al., 2024). Across these models, the condition enters through a fixed interface: a class label or a text prompt is embedded once, and the same condition is reused at every denoising step. It is left to the generator to work out what the condition implies for the image at hand as its state moves from noise to a clean sample. Recent work brings pretrained visual representations into this pipeline, as alignment targets for the generator's features (Yu et al., 2025), as the latent space in which diffusion operates (Zheng et al., 2026), or as intermediate generation targets (Li et al., 2024a).

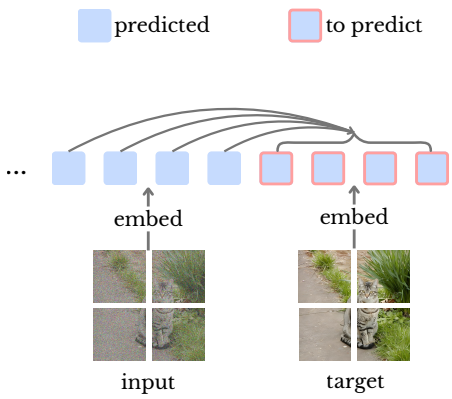


Figure 1: Multi-embedding prediction. From the condition and the noisy image, the model predicts the next embeddings in the sequence, those of the clean image. Supervision is applied in embedding space; the predicted embeddings then condition a diffusion generator.

In all of these, the condition itself stays fixed while the image evolves. We ask: *can the condition itself be predicted from the current noisy image, at every denoising step?*

NEPA trains a Transformer to predict the next embedding in a sequence, rather than to reconstruct pixels or discrete tokens (Xu et al., 2025). Unlike a feature extractor, such a model is predictive by construction: its outputs are estimates of embeddings it has not yet seen. In generation, the sequence runs from the condition through the noisy image to the clean image, so the clean image is what comes next, and its predicted embeddings are a natural candidate for what a generator should condition on.

We answer this question with **Embedding Conditioned Generation** (ECG): a DiT *generator* is conditioned on the embeddings predicted by a frozen NEPA model instead of on the raw condition, and the NEPA model is queried at every denoising step, so the conditioning signal adapts to the current noisy state. The NEPA model is trained with **Multi-Embedding Prediction** (MEP), which extends

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

NEPA to predict the next several embeddings at once, as multi-token prediction does for tokens in language modeling (Gloeckle et al., 2024); it reads the condition and the noisy image and predicts the embeddings of the clean image (Figure 1).

On class-conditional ImageNet 256×256, we compare NEPA predictions with networks of the same size trained in other ways, and study the design of Multi-Embedding Prediction and the scaling of both models. Combined with REPA, our final model, NEPA-DiT-XL, reaches an FID of **1.32**.

We highlight the main contributions of this paper below:

- We propose Embedding Conditioned Generation, which conditions a DiT generator on the clean embeddings predicted by a NEPA model from the condition and the noisy image, recomputed at every denoising step.
- We train the NEPA model with Multi-Embedding Prediction, which extends NEPA to predict the embeddings of the whole clean image at once.
- On class-conditional ImageNet 256×256, we study the condition of the generator, the design of Multi-Embedding Prediction, and the scaling of both models; combined with REPA, NEPA-DiT-XL reaches an FID of 1.32.

2 Related Work

2.1 Diffusion Models and Representations for Generation

Denoising diffusion models have become a dominant framework for image generation (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021b; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021), supported by faster samplers (Song et al., 2021a; Lu et al., 2022; Song et al., 2023) and by flow-based formulations (Lipman et al., 2023; Liu et al., 2023; Ma et al., 2024; Huang et al., 2026). Latent diffusion models operate in the latent space of a pretrained autoencoder to reduce the computational cost (Rombach et al., 2022; Esser et al., 2021; Chen et al., 2025b; Yao et al., 2025), while diffusion transformers replace U-Net denoisers with more scalable architectures (Peebles & Xie, 2023; Bao et al., 2023). Text-to-image models further improve these systems through stronger architectures, flow-based objectives, and text conditioning (Nichol et al., 2022; Ramesh et al., 2022; Saharia et al., 2022; Balaji et al., 2023; Podell et al., 2024; Chen et al., 2024b;a; Esser et al., 2024; Li et al., 2024c; Peng et al., 2025; Xie et al., 2025; 2026). Beyond synthesis, diffusion models have been adapted to text-guided image manipulation and editing (Meng et al., 2022; Hertz et al., 2023; Mokady et al., 2023; Brooks et al., 2023; Xu et al., 2023; 2024), to spatially controlled generation (Zhang et al., 2023), and to temporally consistent video generation (Ho et al., 2022; Guo et al., 2024; Xia et al., 2024). Pretrained visual representations have been brought into this pipeline as generation targets (Li et al., 2024a; Wu et al., 2025; Kouzelis et al., 2025b), alignment signals for diffusion features (Yu et al., 2025; Wang et al., 2026b; Jiang et al., 2026; Tian et al., 2026; Leng et al., 2025), and the generative latent space (Zheng et al., 2026; Tong et al., 2026; Singh et al., 2026). In these models, the class or text condition is injected by adaLN (Peebles & Xie, 2023) or joint attention (Esser et al., 2024) and stays fixed across denoising steps. Unified multimodal models condition a diffusion decoder on embeddings produced by a language model (Sun et al., 2024b; Ge et al., 2024; Pan et al., 2025; Chen et al., 2025a), but these embeddings are likewise computed once before sampling. Self-conditioning (Chen et al., 2023) feeds the estimate of the clean sample from the previous sampling step back into the denoiser, and RIN (Jabri et al., 2023) carries latent states across steps; both reuse the estimates of the denoiser itself. We condition the generator on the predictions of a separate NEPA model, trained with its own embedding prediction objective and recomputed from the current noisy input at every step.

2.2 Predictive Representation Learning

Self-supervised visual learning has progressed from handcrafted pretext tasks (Doersch et al., 2015; Noroozi & Favaro, 2016; Zhang et al., 2016; Gidaris et al., 2018) to contrastive and self-distillation methods (van den Oord et al., 2019; He et al., 2020; Chen et al., 2020; Grill et al., 2020; Caron et al., 2020; 2021; Zhou et al., 2022) and to masked reconstruction approaches that predict pixels, tokens, or visual features (Bao et al., 2022; He et al., 2022; Xie et al., 2022; Wei et al., 2022; Baevski et al., 2022). Generative image models instead learn by predicting discrete or continuous visual tokens,

either autoregressively or through masked and progressively refined prediction (van den Oord et al., 2017; Esser et al., 2021; Yu et al., 2022; Lee et al., 2022; Chang et al., 2022; Li et al., 2023; Yu et al., 2024; Tian et al., 2024; Sun et al., 2024a; Li et al., 2024b). However, pixel and discrete-token prediction can emphasize low-level details or tokenizer-specific structure. Predictive representation learning avoids direct reconstruction by predicting targets in a learned embedding space: JEPA and I-JEPA predict masked target representations from visible context, and V-JEPA extends this principle to spatiotemporal prediction (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025). NEPA further combines representation prediction with causal modeling by replacing the next-token prediction of language models (Radford et al., 2019; Brown et al., 2020) with next-embedding prediction (Xu et al., 2025). Our Multi-Embedding Prediction extends NEPA to predict several embeddings at once, analogous to multi-token prediction in language modeling (Gloeckle et al., 2024).

3 Method

Our method has two stages. First, we train a NEPA model with **Multi-Embedding Prediction** to predict clean image embeddings from condition tokens and noisy image embeddings. Second, we train a DiT *generator* with **Embedding Conditioned Generation**, where the frozen NEPA model produces predictive embeddings at each denoising step and the DiT uses them as its condition.

3.1 Preliminaries

Next-embedding predictive autoregression. NEPA replaces next-token prediction with next-embedding prediction (Xu et al., 2025). An input $\mathbf{x}$ is split into N non-overlapping patches by a convolutional patch embedding layer f, giving a sequence of continuous embeddings $\mathbf{z} = f(\mathbf{x})$. An autoregressive model h_θ reads the sequence and predicts, at each position, the embedding that follows it, and is trained with a distance $\mathcal{D}$ in embedding space between the predictions $\mathbf{p}$ and the targets $\mathbf{z}_{\text{target}}$:

$$\mathbf{p} = h_\theta(\mathbf{z}), \qquad \mathcal{L}_{\text{NEPA}} = \mathcal{D}(\mathbf{z}_{\text{target}}, \mathbf{p}), \tag{1}$$

where $\mathcal{D}$ is any distance between embeddings. For images, NEPA orders the patches into a sequence under causal attention, so the target is the next patch, $\mathbf{z}_{\text{target}} = \mathbf{z}_{2:N}$.

Latent diffusion and flow matching. Following latent diffusion models and DiT (Rombach et al., 2022; Peebles & Xie, 2023), we perform image generation in the latent space of a pretrained autoencoder: a frozen encoder $\mathcal{E}$ maps each image to a clean latent $\mathbf{x}_0$, which serves as the input to both models, and a frozen decoder maps generated latents back to images. During training, we sample Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a noise level t, then construct the noisy input $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}$. We train the diffusion transformer to predict the flow target $\mathbf{v}_t = \boldsymbol{\epsilon} - \mathbf{x}_0$ (Ma et al., 2024; Liu et al., 2023). Given a condition c[1], the generator G_ϕ is optimized with

$$\mathcal{L}_{\text{fm}} = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t, c}\Big[\, w(t) \,\big\|\, \mathbf{v}_t - G_\phi(\mathbf{x}_t, t, c) \,\big\|^2\Big], \tag{2}$$

where $w(t)$ is the timestep-dependent loss weighting. Conventionally, c is fixed across the sampling trajectory; we replace it with embeddings predicted by a NEPA model from both the condition c and the current noisy image state $\mathbf{x}_t$. This allows the conditioning signal to adapt throughout denoising, as described in Section 3.4.

3.2 Multi-Embedding Prediction

NEPA predicts one next embedding at a time. In language modeling, predicting several future tokens at once improves over next-token prediction (Gloeckle et al., 2024). Predicting one embedding at a time is similarly limiting when the next embeddings form a group that is needed together, such as the patch embeddings of an image. **Multi-Embedding Prediction** (MEP) extends NEPA to predict the

[1] A class label for class-to-image generation, or a text prompt for text-to-image generation.

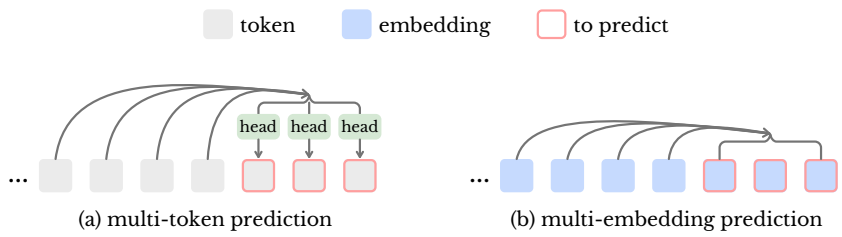


Figure 2: Multi-token vs. multi-embedding prediction. Both predict several items at once. (a) Tokens are discrete, so hidden states are decoded into tokens by output heads. (b) Embeddings are continuous, so the outputs of the model are the predictions themselves. In both, the same context predicts all targets in one forward pass.

next K embeddings simultaneously from the same context. Given the context embeddings $\mathbf{z}_{\leq n}$, the model predicts the next K embeddings in a single forward pass:

$$\mathbf{p}_{n+1:n+K} = h_\theta(\mathbf{z}_{\leq n}), \qquad \mathcal{L}_{\text{MEP}} = \frac{1}{K}\sum_{k=1}^{K} \mathcal{D}(\mathbf{z}_{n+k}, \mathbf{p}_{n+k}). \tag{3}$$

NEPA is the special case $K = 1$. Like multi-token prediction, MEP predicts several items at once; it differs in predicting in embedding space rather than token space. Predicted tokens must be decoded from hidden states by output heads, whereas predicted embeddings are the outputs of the model themselves (Figure 2). Section 3.3 instantiates MEP for image generation.

3.3 NEPA for Generation

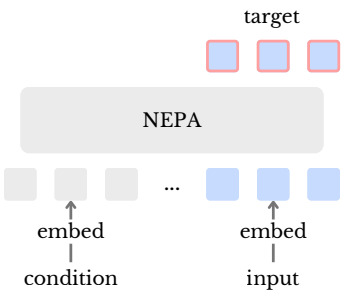


Figure 3: MEP for generation. The clean image embeddings are the next embeddings after the condition and the noisy image; the output at each noisy patch predicts the embedding that follows it.

Generation produces an image through a sequence of states: from the condition, through a noisy image, to the clean image. We write this as a sequence of embeddings, the condition, then the N patch embeddings of the noisy image, then the N patch embeddings of the clean image. In this sequence, the clean embeddings are the next N embeddings after the condition and the noisy image, and we train a NEPA model to predict them with MEP, taking $K = N$. The sequence is ordered by generation state rather than by patch position, so the embedding that follows noisy patch i is clean patch i. MEP predicts all N embeddings of the clean image in one forward pass, each from the full context of the condition and the noisy image. The model is defined by three choices: the sequence and its readout, an attention mask with the condition as a causal prefix, and an InfoNCE loss.

Given a clean input $\mathbf{x}_0$, we sample a noise level t and construct the noisy input $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}$. Both are patchified by the embedding layer into noisy and clean patch embeddings, $\mathbf{z}_t = f(\mathbf{x}_t)$ and $\mathbf{z}_0 = f(\mathbf{x}_0)$. As in NEPA, the targets come from the model's own patch embedding layer f, which is learned jointly with the model. The targets are compared through InfoNCE rather than regressed: the model only has to tell each patch apart from the other patches of the image, and the remaining detail is left to the generator. The noise level is sampled uniformly, $t \sim \mathcal{U}(0, 1)$, so that a single model covers every denoising step at inference, and the condition is replaced by a null token with probability 0.1, so that the same model also provides the unconditional predictions used for classifier-free guidance (Section 3.4). As shown in Figure 3, the NEPA model h_θ reads the condition c, a begin-of-image token $\langle\text{boi}\rangle$, and the noisy embeddings, and predicts the next N embeddings, the clean image:

$$\mathbf{p} = h_\theta([c, \langle\text{boi}\rangle], \mathbf{z}_t), \qquad \mathcal{L}_{\text{MEP}} = \mathcal{D}(\mathbf{z}_0, \mathbf{p}). \tag{4}$$

This is Equation 3 with the context $[c, \langle\text{boi}\rangle, \mathbf{z}_t]$ and the next $K = N$ embeddings $\mathbf{z}_0$. As in NEPA, where the output at each position predicts the embedding that follows it, the output at noisy patch i predicts its next embedding $\mathbf{z}_{0,i}$. The noise level sets the difficulty: at $t = 1$ the noisy image carries

no information and the clean image must be predicted from the condition alone, whereas as $t \to 0$ the noisy image approaches the clean one.

Attention mask. The condition tokens and ⟨boi⟩ form a causal prefix that never attends to image tokens. The image tokens of the noisy image attend to the whole prefix and to each other, and every prediction uses the full spatial context of the noisy image (Figure 4). Since the prefix does not depend on the image, it is encoded once and cached during sampling (Section 3.4).

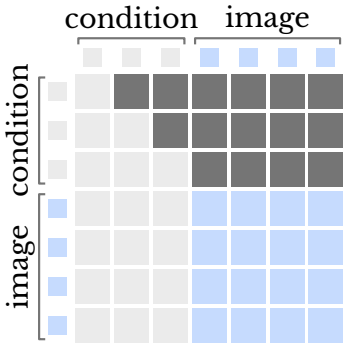


Figure 4: Attention mask. Rows are queries and columns are keys; dark cells are masked.

Embedding prediction loss. Each prediction $\mathbf{p}_i$ is trained to match its next embedding $\mathbf{z}_{0,i}$. The distance $\mathcal{D}$ in Equation 4 can take many forms (Section 4.1); we use the InfoNCE loss (van den Oord et al., 2019), with predicted embeddings as queries and clean embeddings as keys.

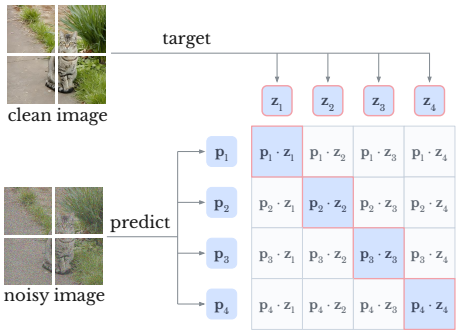


Figure 5: Embedding prediction loss.

As illustrated in Figure 5, the positives lie on the diagonal, pairing each $\mathbf{p}_i$ with $\mathbf{z}_{0,i}$, while the other patches of the same image serve as negatives:

$$\mathcal{D}(\mathbf{z}_0, \mathbf{p}) = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\mathbf{p}_i^\top \mathbf{z}_{0,i})}{\sum_{j=1}^{N} \exp(\mathbf{p}_i^\top \mathbf{z}_{0,j})}. \quad (5)$$

After this stage, the NEPA model maps the condition and the noisy image to predicted clean image embeddings.

3.4 Embedding Conditioned Generation

The second stage trains the generator G_ϕ with the NEPA model h_θ frozen as its conditioning module. The NEPA model predicts the clean embeddings, while the generator learns the denoising vector field.

At each denoising step, the current noisy input $\mathbf{x}_t$ is first fed into the NEPA model, which outputs predictive embeddings $\mathbf{p}_t = h_\theta([c, \langle\text{boi}\rangle], f(\mathbf{x}_t))$. The generator follows DiT (Peebles & Xie, 2023) (details in Appendix C), except that it is conditioned on $\mathbf{p}_t$ through the MM-DiT blocks of SD3 (Esser et al., 2024): the predicted embeddings and the noisy latent tokens are processed with separate weights and exchange information through joint attention, followed by single-stream blocks on the concatenated sequence. The generator receives the condition only through $\mathbf{p}_t$ and has no separate class embedding; the timestep enters through adaLN modulation, as in DiT. The training loss is the weighted flow-matching loss

$$\mathcal{L}_{\text{ECG}} = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon},t,c}\Big[\, w(t) \,\big\|\, \mathbf{v}_t - G_\phi(\mathbf{x}_t, t, \mathbf{p}_t) \,\big\|^2\Big]. \quad (6)$$

During this stage, gradients update only the generator; the VAE and the NEPA model remain frozen.

Condition caching. Because the condition prefix never attends to image tokens, its key-value cache in the NEPA model is computed once before the denoising loop and reused at every step; only the image tokens are re-encoded as $\mathbf{x}_t$ changes.

Classifier-free guidance (Ho & Salimans, 2021). During training, we randomly replace the condition c with a learned null token. At inference, the unconditional branch feeds the null token to the

NEPA model, which yields $\mathbf{p}_t^{\varnothing}$, and guidance is applied to the velocities predicted by the generator:

$$\hat{\mathbf{v}}_{\text{cond}} = G_\phi(\mathbf{x}_t, t, \mathbf{p}_t), \qquad \hat{\mathbf{v}}_{\text{uncond}} = G_\phi(\mathbf{x}_t, t, \mathbf{p}_t^{\varnothing}),$$
$$\hat{\mathbf{v}}_{\text{cfg}} = \hat{\mathbf{v}}_{\text{uncond}} + s \cdot (\hat{\mathbf{v}}_{\text{cond}} - \hat{\mathbf{v}}_{\text{uncond}}), \tag{7}$$

where s is the guidance scale.

4 Experiments

We evaluate our framework on class-conditional ImageNet-1K generation at 256×256 resolution. At small scale, Section 4.1 ablates the design of Multi-Embedding Prediction and Section 4.2 compares conditions of the generator; Section 4.3 then scales up the NEPA model and the generator, and Section 4.4 compares our final model with prior work.

Setup. Following ADM (Dhariwal & Nichol, 2021), images are center-cropped to 256×256 and encoded by the frozen SD-VAE (Rombach et al., 2022) into $32 \times 32 \times 4$ latents. The NEPA model follows Xu et al. (2025) and the generator follows DiT (Peebles & Xie, 2023), each at B, L, and XL scale; NEPA-DiT-B/L/XL denotes a NEPA model paired with a DiT-B/L/XL generator. Unless stated otherwise, the NEPA model is NEPA-XL with patch size 4, and the generator uses patch size 2. The NEPA model is trained with Multi-Embedding Prediction for 240 epochs, then frozen while the generator is trained with the flow-matching loss for 400K iterations (80 epochs), both at batch size 256. Appendix C gives further details.

Evaluation. We report the Fréchet Inception Distance on 50K samples (FID-50K), computed with the ADM evaluation suite and its ImageNet 256×256 reference batch (Dhariwal & Nichol, 2021). Unless stated otherwise, ablation and scaling experiments sample with a 64-step ODE solver and no classifier-free guidance; the sampling setup of our final model is given in Section 4.4.

4.1 Ablation Studies

Following previous work (Ma et al., 2024; Wang et al., 2026b), we run ablations at the B scale with NEPA-B and patch size 2, one factor at a time (defaults in gray); Table 6 then selects patch size 4.

Loss. We compare the InfoNCE loss of Section 3.3 with two regression losses, negative cosine similarity and mean squared error (MSE), which pull each prediction toward its own target without contrasting it against the others. InfoNCE gives the best FID, 31.36 against 35.23 for MSE and 37.82 for cosine similarity (Table 1). ImageNet-1K accuracy of the fine-tuned NEPA model follows the opposite order. InfoNCE must tell each patch apart from the other patches of the same image, so its predictions keep patch-level detail that helps generation; the regression losses favor smoother, more global predictions, which help classification instead.

Loss	FID↓	Acc. (%)↑
Cos. sim.	37.82	**83.1**
MSE	35.23	82.6
InfoNCE	**31.36**	81.7

Table 1: Loss. Acc. is ImageNet-1K top-1 accuracy of the fine-tuned NEPA model.

Detach	FID↓
w/	n.c.
w/o	**31.36**

Table 2: Stop-grad.

Target	FID↓
Norm.	34.76
Raw	**31.36**

Table 3: Target norm.

Timestep	FID↓
w/o	**31.36**
w/	31.40

Table 4: Timestep input.

Stop-gradient. Detaching the targets $\mathbf{z}_0$ from the gradient prevents training from converging (n.c. in Table 2), so gradients flow through the targets.

Target normalization. Normalizing the target embeddings raises FID from 31.36 to 34.76 (Table 3), so we use raw targets.

Timestep input. Feeding the diffusion timestep to the NEPA model changes FID from 31.36 to 31.40 (Table 4), presumably because the noise level is already evident from the noisy patches; we therefore keep the NEPA model timestep-free, while the generator still receives the timestep through adaLN.

Vocabulary. The InfoNCE loss scores each prediction against a bank of candidate targets. We compare banks built from the patches of the same image (instance), of all images in the per-device batch (batch), and of all images across devices (full). The instance, batch, and full banks give 31.36, 32.13, and 31.21 (Table 5). We use the instance bank, which requires no communication across devices; its negatives are the other patches of the same image, as in Equation 5.

Bank	FID↓
Instance	31.36
Batch	32.13
Full	**31.21**

Table 5: Vocabulary.

Patch	Tokens	FID↓
2	256	31.36
4	64	**31.00**
8	16	33.20

Table 6: Patch size.

Patch size. The patch size of the NEPA model sets the number of predicted embeddings (Table 6). Patch size 4 gives 31.00 against 31.36 for patch size 2 with a quarter of the tokens, which makes the query at every denoising step correspondingly cheaper, whereas patch size 8 is too coarse (33.20). We therefore use patch size 4 in all remaining experiments.

4.2 Condition of the Generator

At the same small scale, Table 7 keeps the generator and its training recipe fixed and varies only what it is conditioned on. The class embedding gives 36.39, and adding the patch embeddings of $\mathbf{x}_t$ to the class token gives 37.21. The last three rows give the generator the same extra network as ours, of the NEPA-XL architecture, so that their parameters and FLOPs per step match those of our model; what differs is how that network is trained. Trained end to end with the generator as a single network, it gives 29.52. Pretrained as a flow-matching model for 240 epochs, the same budget as MEP, and then frozen, its features give 30.86. Trained with MEP and frozen, its predicted embeddings give 25.04.

Condition of the generator	Extra network	Training	Objective	FID↓
Class embedding	–	–	–	36.39
Class token + noisy patches	–	–	–	37.21
Single network, end to end	NEPA architecture	joint	flow matching	29.52
Pretrained flow model	NEPA architecture	240 epochs, frozen	flow matching	30.86
NEPA model	NEPA	240 epochs, frozen	MEP	**25.04**

Table 7: Condition of the generator. All rows use the same generator and training recipe (400K iterations, best over sampling steps) and differ only in its condition; the last three rows also carry the extra network of our model.

4.3 Scaling Behavior

		Generator (FID↓)		
NEPA	Acc.↑	DiT-B	DiT-L	DiT-XL
NEPA-B	81.3	31.00	17.69	16.54
NEPA-L	83.4	27.46	14.35	13.74
NEPA-XL	**85.2**	25.04	12.86	**11.27**

Table 8: Scaling. FID-50K after 400K iterations; Acc. is ImageNet-1K top-1 accuracy (%) of the fine-tuned NEPA model.

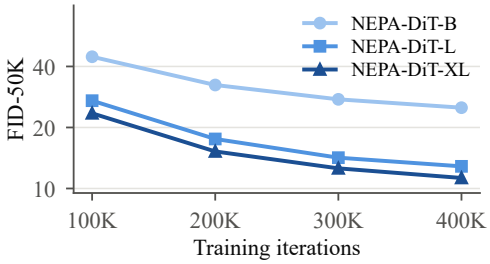


Figure 6: Training progress of NEPA-DiT-B/L/XL, FID-50K without guidance.

Scaling the NEPA model and the generator. Beyond the small-scale setting, Table 8 scales both models, training all nine combinations of NEPA-B/L/XL and DiT-B/L/XL, each for 400K iterations and evaluated with 64 ODE steps without guidance. FID decreases monotonically along both axes. Scaling the generator from DiT-B to DiT-XL improves FID by about 14 points for every NEPA model, and scaling the NEPA model from NEPA-B to NEPA-XL improves FID by about 5 points for every generator. All entries use patch size 4 for the NEPA model and patch size 2 for the generator, and the generators match DiT-B/L/XL in depth, width, and number of heads. Each NEPA model is trained once and shared by the three generators of its row.

Training progress. Figure 6 follows NEPA-DiT-B/L/XL over training; all three use the same frozen NEPA-XL model. Each point is FID-50K with 64 ODE steps and no guidance, evaluated every 100K iterations. Larger generators are better at every checkpoint, and the gap opens early: at 200K iterations, NEPA-DiT-L (17.55) and NEPA-DiT-XL (15.23) are already well ahead of NEPA-DiT-B at 400K (25.04). At 400K iterations, the three reach 25.04, 12.86, and 11.27, the NEPA-XL row of Table 8.

Table 9: Class-conditional generation on ImageNet 256×256 with guidance. Top (gray): methods with modified or alternative tokenizers. Bottom: methods in the standard SD-VAE latent space. Metrics and training epochs on ImageNet-1K are taken from each paper; for ours, epochs are those of the NEPA model + the generator. "–" means not reported. ‡AutoGuidance.

Method	Tokenizer	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Other tokenizers							
REPA + EQ-VAE (Kouzelis et al., 2025a)	EQ-VAE	200	1.70	5.13	283.0	0.79	0.62
LightningDiT-XL/1 (Yao et al., 2025)	VA-VAE	800	1.35	4.15	295.3	0.79	0.65
LightningDiT + IG (Zhou et al., 2026)	VA-VAE	680	1.19	4.11	269.0	0.79	0.66
DiT-XL + CMuon (Chen et al., 2026)	VA-VAE	200	1.18	–	–	–	–
REPA-E (Leng et al., 2025)	E2E-VAE	800	1.12	4.09	302.9	0.79	0.66
Send-VAE + REPA (Page et al., 2026)	Send-VAE	800	1.21	4.10	315.1	0.79	0.66
SFD-XL‡ (Pan et al., 2026b)	SemVAE	800	1.06	3.89	267.0	0.78	0.67
RAE DiT^{DH}-XL‡ (Zheng et al., 2026)	RAE	800	1.13	–	262.6	0.78	0.67
MixFlow + RAE (Li et al., 2026)	RAE	800+200	1.10	4.40	259.7	0.78	0.67
RAEv2 (Singh et al., 2026)	RAE	80	1.06	–	255.3	–	–
SD-VAE							
DiT-XL/2 (Peebles & Xie, 2023)	SD-VAE	1400	2.27	4.60	278.2	0.83	0.57
SiT-XL/2 (Ma et al., 2024)	SD-VAE	1400	2.06	4.49	277.5	0.83	0.59
REPA (Yu et al., 2025)	SD-VAE	800	1.42	4.70	305.7	0.80	0.65
TREAD (Krause et al., 2025)	SD-VAE	740	1.69	4.73	292.7	0.81	0.63
DDT-XL/2 (Wang et al., 2026b)	SD-VAE	400	1.26	–	310.6	0.79	0.65
REG (Wu et al., 2025)	SD-VAE	800	1.36	4.25	299.4	0.77	0.66
U-REPA (Tian et al., 2026)	SD-VAE	400	1.41	–	–	–	–
SRA (Jiang et al., 2026)	SD-VAE	800	1.58	4.65	311.4	0.80	0.63
LSEP (Yun et al., 2025)	SD-VAE	800	1.46	4.94	296.8	0.80	0.64
SPRINT + REPA (Park et al., 2026)	SD-VAE	400	1.59	–	–	0.80	0.64
SiT-XL/2 + IG (Zhou et al., 2026)	SD-VAE	800	1.46	4.79	265.7	0.80	0.64
SiT-XL/2 + Sparse Guid. (Krause et al., 2026)	SD-VAE	400	1.58	4.45	249.7	0.80	0.63
SRA 2 (Wang et al., 2026a)	SD-VAE	800	1.52	4.63	316.2	0.82	0.62
UDT^{+}-XL/2 + REPA (Yun et al., 2026)	SD-VAE	320	1.38	4.38	306.3	0.79	0.66
RecFM-XL (Huang et al., 2026)	SD-VAE	160	2.49	–	–	–	–
NEPA-DiT-XL + REPA (ours), ODE-96	SD-VAE	240+80	1.57	4.82	298.0	0.79	0.63
NEPA-DiT-XL + REPA (ours), SDE-250	SD-VAE	240+80	1.32	4.32	311.0	0.79	0.68

4.4 Comparison with Previous Methods

Setup. Our final model combines Embedding Conditioned Generation, which sets the condition of the generator, with REPA (Yu et al., 2025), which aligns its features. NEPA-DiT-XL pairs NEPA-XL, the frozen model of Section 4.3, with a DiT-XL generator, and the generator is trained with both the flow-matching loss and the REPA alignment loss (DINOv2-B aligned at the third two-stream block, $\lambda = 0.5$) for 400K iterations. We report results with the 96-step ODE and the 250-step SDE samplers of SiT (Ma et al., 2024). Besides FID (Heusel et al., 2017), we report sFID (Nash et al., 2021), Inception Score (Salimans et al., 2016), and precision and recall (Kynkäänniemi et al., 2019) on the same 50K samples.

Scale	Interval	ODE-96	SDE-250
1.0	$[0, 1]$	8.77	7.59
1.4	$[0, 1]$	2.58	2.32
2.4	$[0.3, 1]$	1.72	1.46
3.6	$[0.4, 1]$	**1.57**	**1.32**

Table 10: Guidance. Applied only for t in the interval. Shaded: setting of Table 9.

Guidance. Table 10 lists guidance settings. Restricting guidance to an interval (Kynkäänniemi et al., 2024) admits larger scales: scale 3.6 on $t \in [0.4, 1]$ gives 1.57 and 1.32, the best with both samplers, selected by FID-50K. All guided results in Table 9 use this setting.

Query rate. With the 96-step ODE sampler, computing the NEPA predictions only once, at the first step, and reusing them for all later steps gives an FID of 242.63, as all later steps are then conditioned on embeddings predicted from pure noise. Querying every 8 steps gives 1.82, against 1.57 at every step (Table 14). Querying at every step also sets the sampling cost of the NEPA model, 92 GFLOPs



Figure 7: Qualitative results of NEPA-DiT-XL on ImageNet 256×256.

per step next to 309 for the generator; with 96 ODE steps, one image takes 62 TFLOPs in all, below the 91 TFLOPs of SiT-XL/2 with REPA with 250 SDE steps.

Results. Among the methods of Table 9 in the SD-VAE latent space, NEPA-DiT-XL + REPA reaches an FID of 1.32 with the 250-step SDE sampler and 1.57 with the 96-step ODE sampler. Most of its training compute goes into the NEPA model, trained once: with it included, NEPA-DiT-XL, 1.39B parameters in all, is trained with 3.1×10^{20} FLOPs, about a third of what SiT-XL/2 with REPA uses over its 800 epochs to reach 1.42 (Appendix C). Figure 7 and Appendix E show samples.

5 Conclusion

We have shown that predictive embeddings can serve as the condition of an image generator. In Embedding Conditioned Generation, a DiT generator is conditioned at every denoising step on the clean-image embeddings that a NEPA model, trained with Multi-Embedding Prediction, predicts from the condition and the current noisy image. On ImageNet 256×256, combined with REPA, NEPA-DiT-XL reaches an FID of 1.32 with the 250-step SDE sampler and 1.57 with the 96-step ODE sampler, trained with about a third of REPA's compute even though it trains and runs two networks. Code and model weights will be released. Future work is discussed in Appendix B.

AI USE STATEMENT

We used large language models only to polish the writing of this paper, such as correcting grammar and improving wording. The authors reviewed all edits and take full responsibility for the content.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, June 2023. 3

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew J. Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *CoRR*, abs/2506.09985, 2025. 3

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1298–1312. PMLR, 17–23 Jul 2022. 2

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers, 2023. arXiv:2211.01324. 2

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22669–22679, June 2023. 2

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. 2

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Featured Certification. 3

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, June 2023. 2

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. 3

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924. Curran Associates, Inc., 2020. 2

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, October 2021. 2

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11315–11325, June 2022. 3

Chuyan Chen, Peng Sun, and Kun Yuan. Cmuon: Accelerating and stabilizing diffusion transformer training via chunked momentum orthogonalization, 2026. arXiv:2608.02502. 8

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset, 2025a. arXiv:2505.09568. 2

Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-Σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXXII*, volume 15090 of *Lecture Notes in Computer Science*, pp. 74–91. Springer, 2024a. 2

Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 57611–57640, 2024b. 2, 18

Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 96539–96560, 2025b. 2

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. 2

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning, 2023. arXiv:2208.04202. 2, 18

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025. arXiv:2505.14683. 18

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. 2, 6

Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, June 2021. 2, 3

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 21–27 Jul 2024. 1, 2, 5, 18

Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao WANG, and Ying Shan. Making llama see and draw with seed tokenizer. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 35206–35231, 2024. 2

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 2

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Roziere, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15706–15734. PMLR, 21–27 Jul 2024. 2, 3

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284. Curran Associates, Inc., 2020. 2

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *The Twelfth International Conference on Learning Representations*, 2024. 2

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, June 2022. 2

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *International Conference on Learning Representations*, 2023. 2

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 8

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 5

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. 2, 18

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David Fleet. Video diffusion models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 8633–8646. Curran Associates, Inc., 2022. 2

Jiahe Huang, Sihan Xu, Sharvaree Vadgama, and Rose Yu. Recursive flow matching, 2026. arXiv:2605.26535. 2, 8

Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14569–14589. PMLR, 23–29 Jul 2023. 2, 18

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves, 2026. arXiv:2505.02831. 2, 8

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577. Curran Associates, Inc., 2022. 18

Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. EQ-VAE: Equivariance regularized latent space for improved generative image modeling. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31648–31666. PMLR, 13–19 Jul 2025a. 8

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyridon Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 16685–16714. Curran Associates, Inc., 2025b. 2

Felix Krause, Timy Phan, Ming Gui, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Tread: Token routing for efficient architecture-agnostic diffusion training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15703–15713, October 2025. 8

Felix Krause, Stefan Andreas Baumann, Johannes Schusterbauer, Olga Grebenkova, Ming Gui, Vincent Tao Hu, and Björn Ommer. Guiding token-sparse diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35789–35799, June 2026. 8

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 8

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 122458–122483. Curran Associates, Inc., 2024. 8, 19

Yann LeCun. A Path Towards Autonomous Machine Intelligence, 2022. OpenReview. 3

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11523–11532, June 2022. 3

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18262–18272, October 2025. 2, 8

Hui Li, Jiayue Lyu, Fu-Yun Wang, Kaihui Cheng, Siyu Zhu, and Jingdong Wang. Mixflow training: Alleviating exposure bias with slowed interpolation mixture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9095–9105, June 2026. 8

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36115–36125, June 2026. 18

Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2142–2152, June 2023. 3

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 125441–125468. Curran Associates, Inc., 2024a. 1, 2

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 56424–56445. Curran Associates, Inc., 2024b. 3

Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, Dayou Chen, Jiajun He, Jiahao Li, Wenyue Li, Chen Zhang, Rongwei Quan, Jianxiang Lu, Jiabin Huang, Xiaoyan Yuan, Xiaoxiao Zheng, Yixuan Li, Jihong Zhang, Chao Zhang, Meng Chen, Jie Liu, Zheng Fang, Weiyan Wang, Jinbao Xue, Yangyu Tao, Jianchen Zhu, Kai Liu, Sihuan Lin, Yifu Sun, Yun Li, Dongdong Wang, Mingtao Chen, Zhichao Hu, Xiao Xiao, Yan Chen, Yuhong Liu, Wei Liu, Di Wang, Yong Yang, Jie Jiang, and Qinglin Lu. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding, 2024c. arXiv:2405.08748. 2

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 2

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 3

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan LI, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 5775–5787. Curran Associates, Inc., 2022. 2

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. *SiT: Exploring Flow and Diffusion-Based Generative Models with Scalable Interpolant Transformers*, pp. 23–40. Springer Nature Switzerland, 2024. ISBN 9783031729805. 1, 2, 3, 6, 8, 19

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 2

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047, June 2023. 2

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter Battaglia. Generating images with sparse representations. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7958–7968. PMLR, 18–24 Jul 2021. 8

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 18–24 Jul 2021. 2

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16784–16804. PMLR, 17–23 Jul 2022. 2

Mehdi Noroozi and Paolo Favaro. *Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles*, pp. 69–84. Springer International Publishing, 2016. ISBN 9783319464664. 2

John Page, Xuesong Niu, Kai Wu, and Kun Gai. Boosting latent diffusion models via disentangled representation alignment, 2026. arXiv:2601.05823. 8

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between modalities with metaqueries, 2025. arXiv:2504.06256. 2

Xichen Pan, Aashu Singh, Satya Narayan Shukla, Xiangjun Fan, Shlok Kumar Mishra, and Saining Xie. Repfusion: Leveraging multimodal priors for denoising in representation space, 2026a. arXiv:2606.14700. 18

Yueming Pan, Ruoyu Feng, Qi Dai, Yuqi Wang, Wenfeng Lin, Mingyu Guo, Chong Luo, and Nanning Zheng. Semantics lead the way: Harmonizing semantic and texture modeling with asynchronous latent diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43664–43674, June 2026b. 8

Dogyun Park, Moayed Haji-Ali, Yanyu Li, Willi Menapace, Sergey Tulyakov, Hyunwoo J. Kim, Aliaksandr Siarohin, and Anil Kag. Sprint: Sparse-dense residual fusion for efficient diffusion transformers, 2026. arXiv:2510.21986. 8

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023. 1, 2, 3, 5, 6, 8, 18

Gao Peng, Le Zhuo, Dongyang Liu, DU DU, Xu Luo, Longtian Qiu, Yuhang Zhang, Rongjie Huang, Shijie Geng, Renrui Zhang, Junlin Xie, Wenqi Shao, Zhengkai Jiang, Tianshuo Yang, Weicai Ye, Tong He, HE HE, Junjun He, Yu Qiao, and Hongsheng Li. Lumina-t2x: Scalable flow-based large diffusion transformer for flexible resolution generation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 37254–37292, 2025. 2

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024. 2

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. OpenAI. 3

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. arXiv:2204.06125. 2

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022. 2, 3, 6, 18

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494. Curran Associates, Inc., 2022. 2, 18

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. arXiv:1606.03498. 8

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders, 2026. arXiv:2605.18324. 2, 8

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. 2

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. 2

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023. 2

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation, 2024a. arXiv:2406.06525. 3

Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 12352–12380, 2024b. 2

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 84839–84865. Curran Associates, Inc., 2024. 3

Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-repa: Aligning diffusion u-nets to vits, 2026. arXiv:2503.18414. 2, 8

Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders, 2026. arXiv:2601.16208. 2

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. arXiv:1807.03748. 2, 5

Mengmeng Wang, Dengyang Jiang, Liuzhuozheng Li, Yucheng Lin, Guojiang Shen, Xiangjie Kong, Yong Liu, Guang Dai, and Jingdong Wang. Sra 2: Variational autoencoder self-representation alignment for efficient diffusion training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32978–32987, June 2026a. 8

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40633–40642, June 2026b. 2, 6, 8, 18

Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14668–14678, June 2022. 2

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 7714–7743. Curran Associates, Inc., 2025. 2, 8

Tian Xia, Xuweiyi Chen, and Sihan Xu. Unictrl: Improving the spatiotemporal consistency of text-to-video diffusion models via training-free unified attention control. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. 2

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 20383–20407, 2025. 2

Ji Xie, trevor darrell, Luke Zettlemoyer, and Xudong Wang. Reconstruction alignment improves unified multimodal models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 120095–120137, 2026. 2

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9653–9663, June 2022. 2

Sihan Xu, Ziqiao Ma, Yidong Huang, Honglak Lee, and Joyce Chai. Cyclenet: Rethinking cycle consistency in text-guided diffusion for image manipulation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10359–10384. Curran Associates, Inc., 2023. 2

Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with language-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9452–9461, June 2024. 2

Sihan Xu, Ziqiao Ma, Wenhao Chai, Xuweiyi Chen, Weiyang Jin, Joyce Chai, Saining Xie, and Stella X. Yu. Next-embedding prediction makes strong vision learners, 2025. arXiv:2512.16922. 1, 3, 6, 19

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 15703–15712, June 2025. 2, 8

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *International Conference on Learning Representations*, 2022. 3

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 128940–128966. Curran Associates, Inc., 2024. 3

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 87400–87442, 2025. 1, 2, 8, 18

Junno Yun, Yaşar Utku Alçalar, and Mehmet Akçakaya. No alignment needed for generation: Learning linearly separable representations in diffusion models, 2025. arXiv:2509.21565. 8

Junno Yun, Yaşar Utku Alçalar, and Mehmet Akçakaya. Udt: Reconciling u-nets and diffusion transformers with data-adaptive token reduction, 2026. arXiv:2608.01298. 8

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3836–3847, October 2023. 2

Richard Zhang, Phillip Isola, and Alexei A. Efros. *Colorful Image Colorization*, pp. 649–666. Springer International Publishing, 2016. ISBN 9783319464879. 2

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026. 1, 2, 8, 18

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022. 2

Xingyu Zhou, Qifan Li, Xiaobin Hu, Hai Chen, and Shuhang Gu. Guiding a diffusion transformer with the internal dynamics of itself. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11536–11545, June 2026. 8

A Relation to Other Methods

Relation to RepFusion. RepFusion (Pan et al., 2026a) also conditions a diffusion transformer on the outputs of a model that reads the noisy input at every denoising step. It builds on two existing components: the latent space of a representation autoencoder (Zheng et al., 2026), which it adopts so that noisy latents are compatible with pretrained language-model priors, and a pretrained multimodal LLM, which it repurposes as the encoder of noisy representations. Our NEPA model depends on neither. It is trained from scratch with Multi-Embedding Prediction directly on the latents of a standard VAE, so the conditioning interface is defined by a training objective rather than by an existing model. The same recipe can therefore be applied to any Transformer as the NEPA model and to any generator that accepts conditioning tokens.

Relation to DDT. DDT (Wang et al., 2026b) also splits the generator into a condition encoder that reads the noisy latent and a velocity decoder conditioned on the encoder output. The two are trained jointly: the encoder is supervised indirectly through the velocity loss of the decoder, together with a REPA alignment to DINOv2 features. Our NEPA model is instead trained separately to predict the clean embeddings of the image and is then frozen, so the condition is a representation learned with its own objective rather than an intermediate feature of the generator. DDT also shares the encoder output between adjacent denoising steps to accelerate sampling, and Table 14 studies the corresponding choice for our NEPA model.

Relation to $\mathbf{x}_0$-prediction and self-conditioning. A diffusion model can be parameterized to predict the clean sample $\mathbf{x}_0$ (Ho et al., 2020; Karras et al., 2022), and JiT (Li & He, 2026) predicts it directly with a Transformer on raw pixels. Self-conditioning (Chen et al., 2023) and RIN (Jabri et al., 2023) feed such an estimate, or the latent state behind it, back into the same denoiser at the next step. In both cases the estimate comes from the denoiser itself and is trained by its denoising loss. Our NEPA model is a separate network, trained with its own embedding prediction objective and then frozen, and its predictions are recomputed from the current noisy input at every step. Table 7 includes, in the same setting, a network of the same architecture trained end to end with the generator by flow matching, and the same network pretrained as a flow-matching model and then frozen.

B Future Work

Text conditions. Text-to-image models either encode the prompt once with a separate text encoder (Saharia et al., 2022; Chen et al., 2024b; Esser et al., 2024), which cannot adapt to the evolving noisy image and often fixes the length of the condition, or process the prompt together with the image in a single unified model (Deng et al., 2025), where the condition and the image share one backbone. A NEPA model could instead read the whole prompt before the noisy image and predict the clean image embeddings from both, as it does for class labels in this paper.

Unified models. Since the NEPA model is an autoregressive Transformer, a single NEPA model could in principle serve both as a language model with an output head and as the conditioning model of a DiT generator. Studying text conditions, other latent spaces, and higher resolutions are directions for future work.

C Hyperparameter and More Implementation Details

Further implementation details. Throughout the experiments, the generator matches DiT (Peebles & Xie, 2023) in depth, width, and number of heads, but uses MM-DiT blocks (Esser et al., 2024): the first blocks are two-stream and process the predicted embeddings and the latent tokens with separate weights, and the remaining blocks are single-stream blocks on the concatenated sequence. The NEPA model is kept frozen when training the generator. We use mixed precision (bfloat16) with gradient clipping at 1.0 for both stages, and keep an EMA of the generator weights with decay 0.9999. The timestep of the generator is sampled from a logit-normal distribution with zero mean and unit variance (Esser et al., 2024). We use the `stabilityai/sd-vae-ft-mse` VAE (Rombach et al., 2022) for encoding images during training and the `stabilityai/sd-vae-ft-ema` VAE for decoding samples. For the REPA projector, we use a three-layer MLP with SiLU activations and hidden dimension 2,048, following Yu et al. (2025). We provide a detailed hyperparameter setup in Tables 11 and 12.

Table 11: Hyperparameter setup of the NEPA model.

	NEPA-B	NEPA-L	NEPA-XL
Architecture			
Input dim.	$32 \times 32 \times 4$	$32 \times 32 \times 4$	$32 \times 32 \times 4$
Patch size	4	4	4
Num. layers	12	24	28
Hidden dim.	768	1,024	1,152
Num. heads	12	16	16
Interpolants			
α_t	$1 - t$	$1 - t$	$1 - t$
σ_t	t	t	t
t	$\mathcal{U}(0, 1)$	$\mathcal{U}(0, 1)$	$\mathcal{U}(0, 1)$
Training objective	InfoNCE	InfoNCE	InfoNCE
Optimization			
Training epochs	240	240	240
Batch size	256	256	256
Optimizer	AdamW	AdamW	AdamW
lr	0.0001	0.0001	0.0001
(β_1, β_2)	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)
Weight decay	0.05	0.05	0.05
lr schedule	cosine	cosine	cosine
Warm-up steps	2K	2K	2K
Condition dropout	0.1	0.1	0.1

Pretrained encoder. For REPA, we use the official DINOv2-B checkpoint with registers.

ImageNet-1K accuracy. The accuracy reported in Tables 1 and 8 is the top-1 accuracy of the NEPA model after fine-tuning it for ImageNet-1K classification with the fine-tuning recipe of NEPA (Xu et al., 2025).

Sampling. For the system-level comparison, we use the SDE Euler-Maruyama sampler of SiT (Ma et al., 2024) with 250 steps, or the Euler ODE sampler with 96 steps, and apply classifier-free guidance with a scale of 3.6 only for $t \in [0.4, 1]$ (Kynkäänniemi et al., 2024). Other results use the Euler ODE sampler, and the number of steps is given with each table.

DiT baseline. The DiT-B baseline uses the same generator as NEPA-DiT-B, but conditions it directly on a learned class embedding instead of the embeddings predicted by the NEPA model. The class embedding enters the text stream of the MM-DiT blocks as a single token, in place of the predicted embeddings; in both models, the timestep enters through adaLN modulation. All other settings, including the training objective, timestep distribution, and optimization, are the same (Table 12).

Computing resources. All models are trained on NVIDIA H200 GPUs. The NEPA model is trained once for 240 epochs and is then frozen and shared by all generators; each generator is trained for 400K iterations at batch size 256 (80 epochs). Table 13 lists the parameters and the sampling FLOPs of both models, counted with the PyTorch FLOP counter for one 256×256 image (one multiply-add counts as two FLOPs); SiT-XL/2 is counted in the same way. Training FLOPs are counted with the same counter and include on-the-fly VAE encoding: the NEPA model (240 epochs) and the generator with REPA (400K iterations) together take 3.1×10^{20} FLOPs, and SiT-XL/2 with REPA takes about 1.0×10^{21} FLOPs for its 800 epochs. With classifier-free guidance, the generator and the NEPA model both run on the conditional and the unconditional branch at each guided step.

Table 12: Hyperparameter setup of the generator. Num. layers counts two-stream + single-stream MM-DiT blocks. The last column is the model in Section 4.4.

	NEPA-DiT-B	NEPA-DiT-L	NEPA-DiT-XL	NEPA-DiT-XL + REPA
Architecture				
Input dim.	$32 \times 32 \times 4$	$32 \times 32 \times 4$	$32 \times 32 \times 4$	$32 \times 32 \times 4$
Patch size	2	2	2	2
Num. layers	4+8	8+16	10+18	10+18
Hidden dim.	768	1,024	1,152	1,152
Num. heads	12	16	16	16
NEPA model	NEPA-XL	NEPA-XL	NEPA-XL	NEPA-XL
REPA				
λ	-	-	-	0.5
Alignment depth	-	-	-	3 (two-stream)
$\text{sim}(\cdot, \cdot)$	-	-	-	cos. sim.
Encoder	-	-	-	DINOv2-B (reg.)
Optimization				
Training iteration	400K	400K	400K	400K
Batch size	256	256	256	256
Optimizer	AdamW	AdamW	AdamW	AdamW
lr	0.0001	0.0001	0.0001	0.0001
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Weight decay	0	0	0	0
Condition dropout	0.1	0.1	0.1	0.1
Interpolants				
α_t	$1 - t$	$1 - t$	$1 - t$	$1 - t$
σ_t	t	t	t	t
t	logit-normal	logit-normal	logit-normal	logit-normal
Training objective	v-prediction	v-prediction	v-prediction	v-prediction

Table 13: Parameters and sampling FLOPs. GFLOPs per sampling step for one image; the NEPA model runs on the 64 image tokens with the condition cached. Totals include guidance on $t \in [0.4, 1]$ and VAE decoding. SiT-XL/2 is counted in the same way.

	Params		GFLOPs / step		
Model	Generator	NEPA	Generator	NEPA	TFLOPs / image
NEPA-DiT-B	135M	711M	59	92	–
NEPA-DiT-L	458M	711M	210	92	–
NEPA-DiT-XL, ODE-96	683M	711M	309	92	62
NEPA-DiT-XL, SDE-250	683M	711M	309	92	160
SiT-XL/2 + REPA, SDE-250	675M	–	229	–	91

D Additional Ablations

NEPA query	FID↓
Every step	**1.57**
Every 8 steps	1.82
Once, at the first step	242.63

Table 14: Refresh rate. NEPA-DiT-XL + REPA, 96-step ODE sampling with the guidance of Table 9. The generator runs at every step; between queries, it reuses the last predicted embeddings.

Refresh rate. Table 14 queries the NEPA model less often than every denoising step, reusing the last prediction in between. Querying it every 8 steps raises FID from 1.57 to 1.82, and querying it only once, at the first step, gives 242.63.

E Additional Qualitative Results

Figures 8–10 show more samples of NEPA-DiT-XL on ImageNet 256×256, selected from 16 samples per class.



(a) Goldfish (1)



(b) Bald eagle (22)



(c) Macaw (88)



(d) Sulphur-crested cockatoo (89)



(e) King penguin (145)



(f) Golden retriever (207)



(g) White wolf (270)



(h) Arctic fox (279)

Figure 8: Samples of NEPA-DiT-XL on ImageNet 256 × 256.



(a) Lion (291)



(b) Tiger (292)



(c) Otter (360)



(d) Lesser panda (387)



(e) Giant panda (388)



(f) Balloon (417)



(g) Space shuttle (812)



(h) Cheeseburger (933)

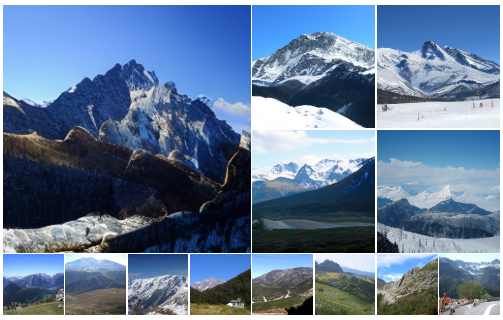
Figure 9: Samples of NEPA-DiT-XL on ImageNet 256 × 256 (continued).



(a) Mushroom (947)



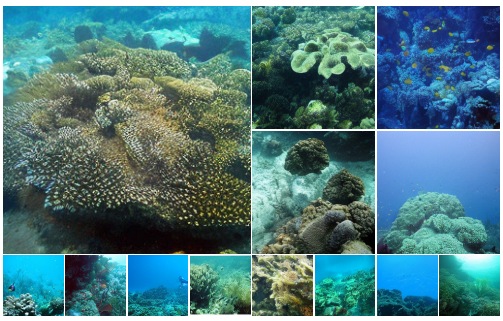
(b) Pizza (963)



(c) Alp (970)



(d) Cliff (972)



(e) Coral reef (973)



(f) Valley (979)



(g) Volcano (980)



(h) Daisy (985)

Figure 10: Samples of NEPA-DiT-XL on ImageNet 256 × 256 (continued).