Representation Preconditioning for Efficient Diffusion Model Training

Deyuan Liu[1*] **Peng Sun**[2,1*] **Xufeng Li**[1,3] **Tao Lin**[1†]
[1]Westlake University [2]Zhejiang University [3]Nanjing University
[*]Equal contribution. [†]Corresponding author.

Abstract

Diffusion transformers learn useful internal representations during training, and learning them from scratch can add a burden to efficient training. Alignment mitigates this burden by adding an auxiliary loss that regularizes diffusion hidden states with features from pretrained visual encoders. In this work, we revisit the role of this alignment signal. Rather than using pretrained representations only as a regularizer during full diffusion training, we use them as a preconditioning signal before diffusion training. The key observation is that early layer alignment from clean latents to pretrained representations can be much cheaper than optimizing the full flow matching objective over the entire backbone, yet provides a better initialization for subsequent diffusion training. We instantiate this idea as Embedded Representation Warmup (ERW). ERW first aligns early layers from clean latents to pretrained representations, and then trains the full model with the standard diffusion objective and decaying alignment. This initialization allows subsequent diffusion training with alignment to converge more efficiently than starting alignment from scratch. Counting warmup in the total budget, ERW improves convergence in SiT based diffusion training. On ImageNet 256×256, ERW reaches FID=1.41 in 350 epochs with SiT-XL/2; it also reaches FID=2.04 on ImageNet 512×512 and improves FID on MS-COCO text-to-image generation.

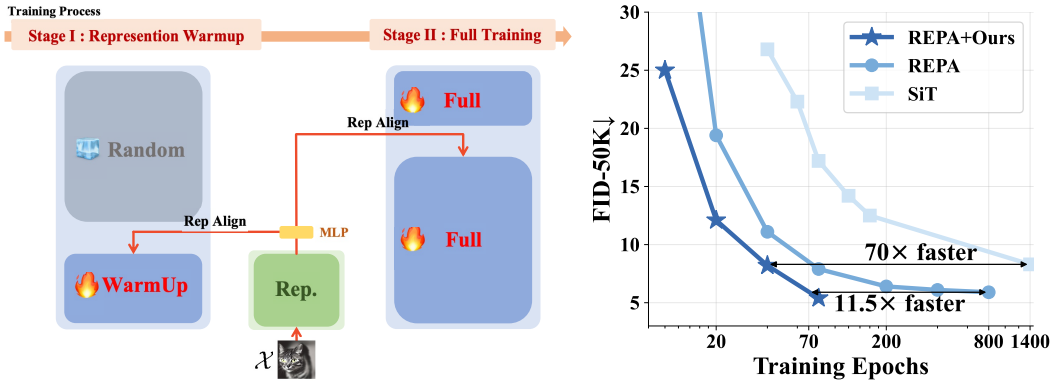


Figure 1: **ERW as representation preconditioning.** In Phase 1, ERW performs a low cost clean latent alignment warmup by updating early layers to align with pretrained visual representations. In Phase 2, the same backbone is trained with the standard diffusion objective and representation alignment. The learning curves count warmup in the total training budget and compare against alignment that starts from scratch during diffusion training.

1 Introduction

Deep generative models, particularly diffusion models [24, 66], have achieved remarkable success in high-fidelity image generation [70, 16, 71, 58, 42, 38]. Despite this progress, training diffusion

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

transformers remains expensive, especially at high resolution and large scale [52, 48, 3], motivating work on more efficient architectures and training recipes [44, 81, 61, 78]. During training, diffusion transformers optimize denoising or flow matching objectives while also forming internal representations used by these objectives [77, 36, 64]. Diffusion models have been shown to learn semantic features during denoising [76, 73], yet these representations can lag behind those learned by dedicated self-supervised encoders [2, 50, 79]. These observations suggest that the quality of internal representations can affect diffusion training efficiency [6, 14, 77].

Alignment methods make this representation bottleneck explicit. By regularizing diffusion hidden states with features from pretrained visual encoders, they provide external supervision for the representations learned inside the generative model [77]. This improvement indicates that pretrained representations can reduce part of the representation learning burden during diffusion training. Recent studies further examine end-to-end tuning, representation entanglement, and the role of global versus spatial structure in alignment [36, 72, 64]. However, in most formulations, alignment is applied together with the diffusion objective from the beginning of training. The representation target is therefore introduced as an auxiliary regularizer during full diffusion training.

This training order can still leave a representation cold start in the aligned early layers. At initialization, the early layers of the diffusion backbone do not yet produce representations aligned with pretrained semantic features. The alignment loss is therefore applied to hidden states produced from noisy latents while the full backbone is optimized for flow matching. Early-layer representation learning therefore begins under the same noisy flow matching objective that alignment is intended to improve. The issue is not the use of pretrained representations, but their role in the optimization: they are used as regularizers during full diffusion training rather than as a preconditioning step before full diffusion training. This motivates the following question:

$\mathbb{Q}$: Should pretrained representation alignment be used only as a regularizer during diffusion training, or can it first serve as an optimization preconditioner?

Our approach. We study alignment as optimization preconditioning rather than only as training-time regularization. Regularization adds an auxiliary term to the training objective; preconditioning changes the initialization from which the training objective is optimized. We instantiate this view with Embedded Representation Warmup **(ERW)**. Before full diffusion training, ERW performs a lower-cost warmup that aligns early layers from clean VAE latents to pretrained visual representations. This warmup is substantially cheaper than full flow matching training because it updates only early layers and avoids full backbone gradient updates. We apply it to early layers because REPA finds early layers effective for representation alignment [77], and our placement ablations show the same trend for preconditioning. The backbone initialized by warmup is then trained with the standard diffusion objective and representation alignment, starting from a better-conditioned initialization rather than from scratch, without changing the architecture or sampler.

Validation. We evaluate whether representation preconditioning improves convergence under a matched total training budget, where warmup steps are counted in the budget. On ImageNet 256×256, ERW reaches **FID = 1.41** in **350 epochs** with SiT-XL/2, and improves convergence over alignment baselines under the same total budget. The same trend extends to ImageNet 512×512, and ERW gives a small gain on MS-COCO text-to-image generation. Ablations show that the benefit is strongest when warmup targets early layers, remains robust across pretrained encoders, and depends on moderate projection depth and alignment strength. Representation diagnostics further indicate improved early-layer feature quality during training.

Our contributions are:

(a) We formulate representation alignment as optimization preconditioning and instantiate it with ERW, which aligns early layers from clean latents before full diffusion training.
(b) We show improved convergence under a matched total training budget on ImageNet 256×256, ImageNet 512×512, and MS-COCO text-to-image generation.
(c) We provide ablations on warmup placement, projection depth, alignment strength, and target encoders, together with representation diagnostics characterizing when preconditioning is effective.

2 Related Work

We discuss prior work on pretrained visual representations for diffusion training, efficient diffusion training, and internal representations in diffusion models.

Pretrained visual representations for diffusion training. Pretrained visual encoders [53, 50] have been used to provide external signals for generation, including as GAN discriminators [59, 33] or teacher models for distillation [39]. For diffusion models, representation alignment uses pretrained visual representations more directly: REPA [77] aligns diffusion hidden states with pretrained visual representations during diffusion training. Subsequent work studies related design choices, including end-to-end tuning of the latent autoencoder [36], representation entanglement [72], and global versus spatial alignment structure [64]. Related studies on image generation also suggest that representation alignment can be sensitive to architecture and training design [63]. Together, these works study which pretrained visual representations to align, how to structure representation alignment, or which components to train. ERW studies a different question: when representation alignment should be optimized. It uses representation alignment before full diffusion training to precondition the backbone, then continues diffusion training with representation alignment from this initialization.

Efficient diffusion training and sampling. Diffusion efficiency has been improved along several axes [13]. Sampling methods reduce inference cost through distillation [57, 60, 62], consistency training [65, 22], and improved solvers [47]. Training-side methods reduce cost through compressed latent spaces [21], attention compression [78], efficient attention architectures [83], micro-budget training setups [61], architectural decoupling [28, 25, 55], and timestep curricula [75]. These methods mainly change the sampler, architecture, latent space, or general training schedule. ERW addresses a different axis: when representation alignment is optimized.

Internal representations in diffusion models. Several studies show that diffusion models learn discriminative semantic features during denoising [76, 37, 73], a phenomenon further discussed in recent surveys and self-supervised analyses [6, 14]. Other methods add auxiliary self-supervised objectives to improve the representations learned during diffusion training, exemplified by MAGE [41] and MaskDiT [81], which draw inspiration from contrastive learning [4, 20] and masked autoencoding [19]. These approaches improve internal representations by adding objectives during diffusion training. ERW instead uses pretrained visual representations before full diffusion training, then continues with the standard diffusion objective and representation alignment.

3 Representation Preconditioning in Latent Diffusion

This section formalizes the view that representation alignment can serve as an optimization preconditioner, not only as a training-time regularizer. To develop this view, we first review the flow matching and alignment setup and describe the representation cold start under existing alignment; we then ask whether the alignment signal can be optimized under a simpler boundary condition, answered by a score decomposition and a variance reduction result; and finally, we instantiate this view as clean latent preconditioning with an early-layer warmup (Section 3.6).

3.1 Flow Matching and Representation Alignment

We use Scalable Interpolant Transformers (SiT) [48, 11, 43, 45] as the latent diffusion training backbone. Given an image $\mathbf{x} \sim p(\mathbf{x})$, a fixed VAE encoder maps it to a clean latent $\mathbf{z}_0 = \mathcal{H}(\mathbf{x})$. With Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, SiT defines a continuous-time interpolation:

$$\mathbf{z}_t = \alpha_t \, \mathbf{z}_0 \; + \; \sigma_t \, \boldsymbol{\epsilon} \, , \tag{1}$$

where $t \in [0, 1]$, and α_t, σ_t are differentiable functions satisfying

$$(\alpha_0, \sigma_0) = (1, 0) \quad \text{and} \quad (\alpha_1, \sigma_1) = (0, 1) \, . \tag{2}$$

Thus $t = 0$ gives the clean latent and $t = 1$ gives pure noise. SiT trains a velocity model $\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$ by differentiating the interpolation with respect to t,

$$\dot{\mathbf{z}}_t = \dot{\alpha}_t \, \mathbf{z}_0 \; + \; \dot{\sigma}_t \, \boldsymbol{\epsilon} \, . \tag{3}$$

The diffusion objective for training the velocity model is

$$\mathcal{L}_{\text{diffusion}}(\boldsymbol{\theta}) \; = \; \mathbb{E}_{\mathbf{z}_0, \boldsymbol{\epsilon}, t} \left[\left\| \mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z}_t, t) - \left(\dot{\alpha}_t \, \mathbf{z}_0 + \dot{\sigma}_t \, \boldsymbol{\epsilon} \right) \right\|^2 \right] . \tag{4}$$

For representation alignment, we use a fixed pretrained visual encoder $\mathbf{f}_{\text{rep}}$ and a projection head $\mathcal{T}_{\boldsymbol{\theta}}$. Given hidden states produced by early layers $\mathcal{R}_{\boldsymbol{\theta}_{\text{early}}}(\cdot)$, alignment compares projected diffusion features with $\mathbf{f}_{\text{rep}}(\mathbf{x})$ using the contrastive loss $\ell_{\text{NT-Xent}}$.



Figure 2: **Selected Samples on ImageNet** 256×256**.** Images generated by SiT-XL/2 with ERW using classifier-free guidance (CFG) with a scale of $w = 1.62$ at 350 epochs.

3.2 Alignment as Training Time Regularization

Existing alignment methods introduce pretrained visual representations as an auxiliary regularizer during diffusion training from the outset. In this view, the model is trained from a random initialization with an objective of the form

$$\mathcal{L}_{\text{reg}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{diffusion}}(\boldsymbol{\theta}) + \lambda\, \mathbb{E}_{\mathbf{x},\boldsymbol{\epsilon},t} \ell_{\text{NT-Xent}}\Big(\mathcal{T}_{\boldsymbol{\theta}}(\mathcal{R}_{\boldsymbol{\theta}_{\text{early}}}(\mathbf{z}_t, t)), \mathbf{f}_{\text{rep}}(\mathbf{x})\Big). \tag{5}$$

This formulation is effective, but the alignment signal is introduced only after full diffusion training has started. At that point, early layers are still randomly initialized, the alignment input is a noisy latent $\mathbf{z}_t$, and the full backbone is already optimized for the flow matching target. Thus, representation alignment must bootstrap early-layer features under the same noisy diffusion optimization that it is intended to improve. This is the representation cold start discussed in the introduction: the pretrained visual representation is available from the beginning, but its effect is mediated through randomly initialized early layers and noisy inputs. This creates three mismatches at initialization:

(i) **Initialization mismatch**: early layers are random when alignment is first applied.
(ii) **Noise mismatch**: hidden states are produced from noisy latents $\mathbf{z}_t$ at arbitrary noise levels, while the target representation comes from clean images.
(iii) **Objective mismatch**: the same backbone is simultaneously optimized for flow matching and representation alignment.

3.3 A Probabilistic View of the Alignment Signal

The three mismatches identified above suggest that alignment under noisy diffusion conditions is harder than necessary. The central question for ERW is not whether alignment should be used, but *when* the representation target should enter the optimization—specifically, whether it can be optimized under a simpler boundary condition before the full diffusion problem is learned. To formalize this, we view the alignment target as an auxiliary variable in an augmented probability space and ask how the corresponding score decomposes. Let:

$$\mathbf{r} = \mathbf{f}_{\text{rep}}(\mathbf{x}), \qquad \mathbf{z}_0 = \mathcal{H}(\mathbf{x}). \tag{6}$$

The pair $(\mathbf{z}_0, \mathbf{r})$ is observed during training, while $\mathbf{r}$ is not used as an inference-time input. For analysis, consider the augmented conditional distribution $p(\mathbf{z}_0, \mathbf{r} \mid \mathbf{z}_t, t)$. Where the displayed conditional log-probabilities are defined and differentiable in $\mathbf{z}_t$, the chain rule gives,

$$\log p(\mathbf{z}_0, \mathbf{r} \mid \mathbf{z}_t, t) = \log p(\mathbf{z}_0 \mid \mathbf{z}_t, \mathbf{r}, t) + \log p(\mathbf{r} \mid \mathbf{z}_t, t). \tag{7}$$

Taking gradients with respect to the noisy latent gives

$$\nabla_{\mathbf{z}_t} \log p(\mathbf{z}_0, \mathbf{r} \mid \mathbf{z}_t, t) = \nabla_{\mathbf{z}_t} \log p(\mathbf{z}_0 \mid \mathbf{z}_t, \mathbf{r}, t) + \nabla_{\mathbf{z}_t} \log p(\mathbf{r} \mid \mathbf{z}_t, t). \tag{8}$$

This identity separates conditional log-density gradients into generation and representation components. It motivates studying when to apply alignment, without equating the practical losses to exact estimation of these components. Marginalizing the auxiliary representation preserves the standard diffusion marginal (Appendix B.2). The corresponding optimal velocity satisfies the tower identity

$$\mathbf{F}^*(\mathbf{z}_t, t) = \mathbb{E}[\mathbf{v} \mid \mathbf{z}_t, t] = \mathbb{E}_{\mathbf{r}|\mathbf{z}_t,t}[\mathbb{E}[\mathbf{v} \mid \mathbf{z}_t, \mathbf{r}, t]]\,, \tag{9}$$

where $\mathbf{v} = \dot{\alpha}_t \mathbf{z}_0 + \dot{\sigma}_t \boldsymbol{\epsilon}$. The teacher representation is training supervision, not an inference input; the auxiliary alignment loss does not guarantee that a finite model attains this optimum.
At each fixed t, access to the true $\mathbf{r}$ cannot increase the optimal MSE (Appendix B.4):

$$\underbrace{\mathbb{E}\left[\left\|\mathbf{v} - \mathbb{E}\left[\mathbf{v} \mid \mathbf{z}_t, t\right]\right\|^2\right]}_{\text{optimal MSE without } \mathbf{r}} = \underbrace{\mathbb{E}\left[\left\|\mathbf{v} - \mathbb{E}\left[\mathbf{v} \mid \mathbf{z}_t, \mathbf{r}, t\right]\right\|^2\right]}_{\text{optimal MSE with } \mathbf{r}} + \underbrace{\Delta(t)}_{\geq 0}. \tag{10}$$

This is an oracle comparison with additional information. It does not establish monotonicity of $\Delta(t)$ in noise level or a convergence guarantee for ERW. The empirical motivation is that clean-latent warmup may provide a useful initialization for learning from noisy inputs.

3.4 Clean Latent Alignment as Boundary Preconditioning

ERW changes the optimization order. At the clean boundary $t = 0$, $\mathbf{z}_t$ reduces to $\mathbf{z}_0$. Warmup optimizes the clean-latent-to-representation objective:

$$\mathcal{L}_{\text{warm}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}} \ell_{\text{NT-Xent}}\Big(\mathcal{T}_{\boldsymbol{\theta}}(\mathcal{R}_{\boldsymbol{\theta}_{\text{early}}}(\mathbf{z}_0, 0)), \mathbf{f}_{\text{rep}}(\mathbf{x})\Big), \quad \mathbf{z}_0 = \mathcal{H}(\mathbf{x}). \tag{11}$$

Only the selected early layers and the projection head are updated in this stage; finite-step warmup need not attain a global minimum. Compared with full flow matching training, this warmup avoids time sampling, noisy latent inputs, and gradient updates through the full backbone. Appendix B.3 identifies NT-Xent with a classification negative log-likelihood over the given in-batch candidates. This provides a representation-learning surrogate, rather than exact maximum likelihood for the full continuous conditional distribution. Its role is not to introduce an additional inference module, but to change the initialization from which the subsequent diffusion objective is optimized. We use the term preconditioning in this optimization sense. Alignment first prepares the early layers, after which Phase 2 starts from the resulting initialization:

$$\boldsymbol{\theta}^0_{\text{ERW}} = \text{Warmup}(\boldsymbol{\theta}^0), \qquad \mathcal{L}_2(\boldsymbol{\theta}; k) = \mathcal{L}_{\text{diffusion}}(\boldsymbol{\theta}) + \lambda_{\text{train}}(k)\, \mathbb{E}_t[\mathcal{L}_{\text{align}}(\boldsymbol{\theta}; t)]. \tag{12}$$

Here $\mathcal{L}_{\text{align}}$ is the unweighted alignment loss defined below. This contrasts with training-time regularization, which starts the joint diffusion and alignment objective directly from a random initialization $\boldsymbol{\theta}^0$.

Takeaway. ERW does not change the representation target used by alignment. It changes its optimization role: before diffusion training, the target forms a preconditioned initialization; during diffusion training, the same alignment objective is optimized with flow matching.

3.5 Choice of Early Layers

We apply the warmup to early layers for an empirical reason rather than a strict layer-wise specialization assumption. Prior alignment work finds early layers effective targets for representation alignment [77], and these layers are the first trainable transformation from VAE latents to the hidden states used by the rest of the backbone. Our placement ablations show the same trend for preconditioning: warmup is most effective when it targets early layers, and becomes less effective when moved to middle or late layers. This empirical observation is sufficient for our method.

3.6 Two-Phase Training

ERW implements the preconditioning view with two phases. Phase 1 aligns early layers on clean VAE latents. Phase 2 trains the full diffusion model with the standard diffusion objective and representation alignment. We use the same unweighted alignment loss at a specified noise level:

$$\mathcal{L}_{\text{align}}(\boldsymbol{\theta}; t) = \mathbb{E}_{\mathbf{x}, \boldsymbol{\epsilon}}\left[\ell_{\text{NT-Xent}}\left(\mathcal{T}_{\boldsymbol{\theta}}(\mathcal{R}_{\boldsymbol{\theta}_{\text{early}}}(\mathbf{z}_t, t)), \mathbf{f}_{\text{rep}}(\mathbf{x})\right)\right]. \tag{13}$$

Here $\mathbf{z}_0 = \mathcal{H}(\mathbf{x})$ and $\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}$. Phase 1 fixes $t = 0$, so $\mathbf{z}_t = \mathbf{z}_0$, and optimizes the selected early layers and projection head.

Phase 2 samples $t \sim \mathcal{U}[0,1]$ and trains the full backbone with alignment weight $\lambda_{\text{train}}(k) = c_0 \exp(-k/\tau_{\text{d}})$. Here k counts optimizer steps from the start of Phase 2; c_0 and $\tau_{\text{d}} > 0$ control the initial weight and decay. The NT-Xent temperature $\tau_{\text{c}} > 0$ is a separate parameter. Although the warmup is performed on clean latents at t=0, the early layers initialized by warmup are further adapted to noisy inputs during Phase 2, which continues alignment training over $t \sim \mathcal{U}[0,1]$; the warmup provides a better starting point rather than a fixed representation. The exponential decay of $\lambda_{\text{train}}(k)$ is a practical schedule choice: as the backbone develops increasingly strong generative representations during training, gradually reducing the external alignment signal allows the model to devote its full capacity to the flow matching objective. The full training objective is

$$\mathcal{L}_{\text{total}}(\boldsymbol{\theta}; k) = \mathcal{L}_{\text{diffusion}}(\boldsymbol{\theta}) + \lambda_{\text{train}}(k)\, \mathbb{E}_t[\mathcal{L}_{\text{align}}(\boldsymbol{\theta}; t)]. \tag{14}$$

with $\mathcal{L}_{\text{diffusion}}$ defined in Eq. (4) and $\mathcal{L}_{\text{align}}$ defined in Eq. (13). In practice, we instantiate $\ell_{\text{NT-Xent}}$ with in-batch negatives and use the same projection head $\mathcal{T}_{\boldsymbol{\theta}}$ across both phases (Section 4.1). Thus, Phase 2 uses the same alignment objective as prior work, but starts from early layers initialized by warmup. Both phases share the same alignment loss; they differ in the noise level of the input, the optimization scope, and the alignment weight schedule.

Table 1: **ImageNet 256×256 comparison with classifier-free guidance.** For ERW, epochs include clean latent warmup and full diffusion training. Asterisks indicate advanced CFG scheduling; for ERW, we use guidance interval scheduling [34]. ↓ and ↑ indicate whether lower or higher values are better.

Model	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Pixel diffusion						
ADM-U	400	3.94	6.14	186.7	0.82	0.52
VDM++	560	2.40	-	225.3	-	-
Simple diffusion	800	2.77	-	211.8	-	-
CDM	2160	4.88	-	158.7	-	-
Latent diffusion, U-Net						
LDM-4	200	3.60	-	247.7	0.87	0.48
Latent diffusion, Transformer + U-Net hybrid						
U-ViT-H/2	240	2.29	5.68	263.9	0.82	0.57
DiffiT*	-	1.73	-	276.5	0.80	0.62
MDTv2-XL/2*	1080	1.58	4.52	314.7	0.79	0.65
Latent diffusion, Transformer						
MaskDiT	1600	2.28	5.67	276.6	0.80	0.61
SD-DiT	480	3.23	-	-	-	-
DiT-XL/2	1400	2.27	4.60	278.2	**0.83**	0.57
SiT-XL/2	1400	2.06	4.50	270.3	0.82	0.59
+ REPA	200	1.96	4.49	264.0	0.82	0.60
+ REPA*	800	1.42	4.70	305.7	0.80	0.65
+ ReDi	800	1.61	4.66	295.1	0.78	0.64
+ ERW (ours)	200	1.64	4.71	260.2	0.78	**0.66**
+ ERW (ours)*	**350**	**1.41**	**4.46**	**293.9**	0.79	0.65

Scope. We use preconditioning in the optimization sense of changing the starting point of training. We do not claim a Hessian preconditioner, a strict modular decomposition of the backbone, or that alignment should be removed during full diffusion training.

4 Experiments

We evaluate ERW through three controlled questions:

(i) **Matched-budget convergence**: does clean latent preconditioning improve convergence when warmup is counted in the total budget?

(ii) **Post-warmup convergence**: does the preconditioned initialization make subsequent diffusion-and-alignment training faster?

(iii) **Mechanism and scope**: when does preconditioning help, and does the effect extend to higher resolution and text-to-image generation?

Table 2: **ImageNet 512×512 comparison with classifier-free guidance.** We follow the REPA setup and use CFG with $w = 1.35$. For ERW, epochs include clean latent warmup and full diffusion training. ↓ and ↑ indicate whether lower or higher values are better.

Model	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Pixel diffusion						
VDM++	-	2.65	-	278.1	-	-
ADM-G, ADM-U	400	2.85	5.86	221.7	0.84	0.53
Simple diffusion (U-Net)	800	4.28	-	171.0	-	-
Simple diffusion (U-ViT, L)	800	4.53	-	205.3	-	-
Latent diffusion, Transformer						
MaskDiT	800	2.50	5.10	256.3	0.83	0.56
DiT-XL/2	600	3.04	5.02	240.8	0.84	0.54
SiT-XL/2	600	2.62	4.18	252.2	0.84	0.57
+ REPA	80	2.44	4.21	247.3	0.84	0.56
+ REPA	100	2.32	**4.16**	255.7	**0.84**	0.56
+ REPA	200	2.08	4.19	274.6	0.83	0.58
+ ERW (ours)	160	2.12	4.28	270.6	0.82	**0.59**
+ ERW (ours)	180	**2.04**	**4.16**	**280.7**	**0.84**	0.58

System-level comparisons provide context, while our main conclusions are drawn from controlled SiT/REPA settings where the backbone, sampler, and alignment target are matched.

4.1 Setup

• ***Implementation Details.*** We follow the experimental setups of DiT [52] and SiT [48], unless otherwise noted. Specifically, we utilize the ImageNet dataset [8], preprocessing each image to a resolution of 256 × 256 pixels. Following the protocols of ADM [9], each image is encoded into a compressed latent vector $\mathbf{z} \in \mathbb{R}^{32\times32\times4}$ using the Stable Diffusion VAE [54]. For our model configurations, we employ the B/2 and XL/2 architectures as introduced in the SiT papers, which process inputs with a patch size of 2. We maintain a batch size of 256 throughout training. Further experimental details, including hyperparameter settings are provided in Appendix D .

• ***Budget accounting.*** For ERW, reported training budgets include both clean latent warmup and full diffusion training unless a table explicitly reports full diffusion iterations after warmup. This accounting is used to separate two questions: whether the complete training pipeline is more efficient under a matched total budget, and whether the preconditioned initialization changes early full diffusion convergence. We use two forms of comparison:

(i) **Matched total budget**: warmup and full diffusion steps are both counted.
(ii) **Post-warmup convergence**: full diffusion steps are counted after loading the warmup checkpoint, to isolate the initialization effect.

• ***Evaluation.*** We report Fréchet inception distance (FID; [23]), sFID [49], inception score (IS; [56]), precision (Pre.) and recall (Rec.) [35] using 50K samples. We also include CKNNA [27] as discussed in ablation studies. Detailed setups for evaluation metrics are provided in Appendix E.

• ***Sampler and Alignment objective.*** Following SiT [48], we use the SDE Euler-Maruyama sampler (for SDE with $w_t = \sigma_t$) and set the number of function evaluations (NFE) to 250 by default. Unless otherwise noted, representation alignment uses the NT-Xent objective and DINOv2 features, following REPA [77].

• ***Baselines.*** We report system-level baselines across several diffusion model families for context. These include (a) *Pixel diffusion*: ADM [9], VDM++ [30], Simple diffusion [26], CDM [25], (b) *Latent diffusion with U-Net*: LDM [54], (c) *Latent diffusion with transformer+U-Net hybrid models*: U-ViT-H/2 [1], DiffiT [18], and MDTv2-XL/2 [15], and (d) *Latent diffusion with transformers*: MaskDiT [81], SD-DiT [85], DiT [52], and SiT [48]. Hybrid transformer and U-Net models contain skip connections that are not used in the pure transformer backbone. Details are provided in Appendix F.

Table 3: **Effect of warmup depth, projection depth, and alignment weight.** All models use SiT-XL/2 and are trained for 100K iterations without classifier-free guidance on ImageNet 256×256. The target representation model is DINOv2-B, and the objective is NT-Xent. The best FID in this sweep is obtained with 5 warmup layers, projection depth 18, and $\lambda = 0.5$. ↓ and ↑ indicate whether lower or higher values are better.

ERW Depth	Proj. Depth	λ	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-XL/2 + REPA [77]			19.4	6.06	67.4	0.64	0.61
3	8	0.5	14.4	**5.28**	82.7	0.68	0.62
4	8	0.5	13.8	5.31	87.1	0.68	0.62
5	8	0.5	13.4	5.29	87.8	0.68	0.63
6	8	0.5	13.6	5.30	87.3	0.67	0.63
8	8	0.5	15.4	5.37	82.3	0.66	0.63
12	8	0.5	16.2	5.36	79.2	0.66	0.63
5	10	0.5	12.9	5.29	90.4	0.68	0.63
5	12	0.5	12.5	5.24	92.0	0.69	0.62
5	14	0.5	12.5	5.26	91.5	0.69	0.63
5	16	0.5	12.3	5.25	93.4	0.69	0.62
5	18	0.5	**12.1**	**5.25**	**94.2**	**0.69**	0.63
5	20	0.5	12.6	5.27	92.3	0.69	0.63
5	18	0.1	16.6	5.31	75.8	0.67	0.60
5	18	1.0	12.7	5.41	92.8	0.68	0.64
5	18	2.0	13.3	5.39	90.5	0.68	0.63
5	18	4.0	13.1	5.38	92.2	0.68	**0.64**
5	18	6.0	13.4	5.45	91.6	0.67	0.63

4.2 Comparison

Table 1 reports ImageNet 256×256 results with classifier-free guidance. The system-level comparison provides context across diffusion backbones, while the controlled comparison is between SiT, REPA, and ERW under the same SiT-XL/2 backbone. At the same 200 epoch budget, ERW improves over REPA under the same SiT-XL/2 backbone. With warmup counted in the training budget, ERW reaches FID **1.41** at **350 epochs**, comparable to REPA reaching FID 1.42 at 800 epochs. This is consistent with the view that preconditioning improves the convergence path without changing the architecture or sampler. Figure 2 shows generated samples from ERW. At 200 epochs, ERW reduces FID from 1.96 to 1.64 relative to REPA.

Table 4: **Early convergence without classifier-free guidance on ImageNet** 256×256**.** All results use SiT-XL/2. For ERW, the iteration count reports full diffusion training after clean latent warmup; the warmup cost is analyzed separately in the cost analysis figure. ↓ and ↑ indicate whether lower or higher values are better.

Model	#Params	Iter.	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-XL/2	675M	7M	8.3	6.32	131.7	0.68	0.67
+REPA	675M	50K	52.3	31.24	24.3	0.45	0.53
+ERW (ours)	675M	50K	**25.0**	**12.06**	**56.1**	**0.62**	**0.57**
+REPA	675M	100K	19.4	6.06	67.4	0.64	0.61
+ERW (ours)	675M	100K	**12.1**	**5.25**	**94.2**	**0.69**	**0.63**

Takeaway. Under the same SiT-XL/2 backbone and a matched total budget, clean latent warmup improves convergence over starting alignment from scratch. This supports the view that ERW gains from a better initialization rather than from changing the architecture or sampler.

4.3 Scaling and Transfer

We use ImageNet 512×512 to test whether the same training order remains useful at higher resolution, and MS-COCO to test whether it transfers beyond class-conditional ImageNet generation. On ImageNet 512×512, ERW reaches FID 2.04 at 180 epochs, slightly better than REPA at 200 epochs under the same evaluation setup. For MS-COCO text-to-image generation, ERW matches REPA under the ODE sampler and gives a small improvement under the SDE sampler. These results suggest that the preconditioning effect is not tied to a single resolution, while the text-to-image gain is modest.

Takeaway. The same training order extends to ImageNet 512×512 and MS-COCO, but the gains are modest. We treat these results as additional evidence for representation preconditioning, while keeping the main claim anchored in controlled ImageNet comparisons.

4.4 ERW Efficiency

We next isolate early convergence before guidance scheduling.

• ***Early full diffusion convergence.*** In Table 4, we isolate the effect of the preconditioned initialization by comparing early full diffusion training after warmup with REPA trained from scratch. At 50K full diffusion iterations, ERW reduces FID from 52.3 to 25.0; at 100K iterations, it reduces FID from 19.4 to 12.1.

Table 5: **Effect of warmup placement.** All models use SiT-B/2 and are trained for 50K iterations without classifier-free guidance on ImageNet 256×256. Aligning the earliest layers gives the best result, while moving the warmup window to later layers reduces the benefit. ↓ and ↑ indicate whether lower or higher values are better.

Target Repr.	Depth	Objective	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-B/2 + REPA [77]			78.2	11.71	17.1	0.33	0.48
DINOv2-B	0-8	NT-Xent	**54.2**	**8.12**	**27.2**	**0.45**	**0.59**
DINOv2-B	1-9	NT-Xent	69.1	13.0	18.7	0.37	0.51
DINOv2-B	2-10	NT-Xent	67.7	13.4	19.0	0.38	0.52
DINOv2-B	3-11	NT-Xent	67.5	11.8	19.5	0.38	0.52
DINOv2-B	4-11	NT-Xent	67.8	13.1	19.0	0.38	0.52

• ***Effect of the preconditioned initialization.*** This comparison isolates the downstream effect of clean latent warmup: the subsequent diffusion and alignment training starts from early layers that have already been aligned to pretrained visual representations. The gains appear before guidance scheduling and long training, so they are better explained by a changed optimization trajectory than by a final sampling configuration. This is an initialization test: after warmup, the same diffusion and alignment objective converges faster from the preconditioned early layers.

Takeaway. After warmup, the same diffusion and alignment objective converges faster from the preconditioned initialization. This indicates that ERW changes the early optimization trajectory, while the warmup itself remains a cheaper clean latent alignment step than full backbone training.

Target representation model. The warmup stage is less expensive per iteration than full diffusion training because it aligns clean latents using only early layers and does not backpropagate through the full backbone for flow matching. Figure 3 reports this cost difference. This cost profile is important for total budget accounting: ERW reallocates part of training from full backbone flow matching to a cheaper clean latent alignment step before full diffusion training.

Table 6: **MS-COCO text-to-image generation.** We compare MMDiT, MMDiT with representation alignment, and MMDiT with ERW under the same sampler settings. ERW matches REPA under the ODE sampler and gives a small gain under the SDE sampler. ↓ indicates lower is better.

Method	Type	FID↓
AttnGAN [74]	GAN	35.49
DM-GAN [84]	GAN	32.64
VQ-Diffusion [17]	Discrete Diffusion	19.75
DF-GAN [69]	GAN	19.32
XMC-GAN [80]	GAN	9.33
Frido [12]	Diffusion	8.97
LAFITE [82]	GAN	8.12
U-Net [1]	Diffusion	7.32
U-ViT-S/2 [1]	Diffusion	5.95
U-ViT-S/2 (Deep) [1]	Diffusion	5.48
MMDiT (ODE; NFE=50)	Diffusion	6.05
MMDiT+REPA (ODE; NFE=50)	Diffusion	**4.73**
MMDiT+ERW (ODE; NFE=50)	Diffusion	**4.73**
MMDiT (SDE; NFE=250)	Diffusion	5.30
MMDiT+REPA (SDE; NFE=250)	Diffusion	4.14
MMDiT+ERW (SDE; NFE=250)	Diffusion	**4.10**

4.5 Ablation Studies

We use ablations to study which components are important for representation preconditioning. The goal is not to search for an isolated best hyperparameter, but to test whether the effect is tied to early layers, target encoder choice, projection capacity, and alignment weight.

Target representation. We first compare alignment with multiple pretrained visual encoders: MoCov3, CLIP, and DINOv2, as summarized in Table 7. • ***Target encoder robustness.*** Across pretrained encoders, ERW gives similar early training FID ranges, indicating that the warmup is not tied to a single target encoder. • ***DINOv2 scale.*** DINOv2-B, DINOv2-L, and DINOv2-g yield comparable gains, suggesting that ERW does not require the largest possible target encoder for effective representation transfer. Increasing target encoder scale does not monotonically improve ERW in this sweep, suggesting that the benefit is not simply due to using a larger target encoder.

Warmup placement. We next test which layers should be used for representation preconditioning. We specify the alignment target using "Depth X–Y", which means the alignment loss is computed on the output of layer Y, using a projection head that takes features from layers X through Y as input. This ablation keeps the target encoder and alignment objective fixed, and only moves where warmup is applied.

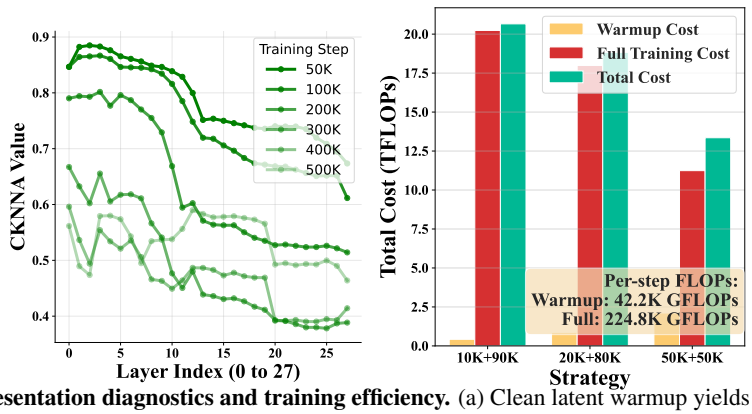


Figure 3: **Representation diagnostics and training efficiency.** (a) Clean latent warmup yields higher CKNNA in early layers, indicating improved representation quality before full diffusion training. (b) Warmup updates only early layers with a clean latent alignment objective, making it substantially cheaper per iteration than full backbone flow matching.

• ***Early-layer placement.*** In Table 5, warmup on the earliest layers (0–8) notably outperforms warmup on middle or later sections (FID 54.2 vs. > 67). • ***Consistent with Representation Preconditioning.*** This suggests that preconditioning is most useful when applied to the early layers that first process latent inputs. The degradation from 0–8 to later windows suggests that ERW is not merely adding extra alignment optimization. Warmup placement is the strongest ablation for the preconditioning view: moving the same alignment objective away from early layers weakens the benefit.

Projection depth and alignment weight. We also investigate how the final projection head depth and the alignment loss coefficient λ affect training (Table 3). The projection head, $\mathcal{T}_{\theta}$, is a deep MLP that maps diffusion features to the dimensionality of the target representation, following REPA. • ***Projection capacity.*** Warmup depth shows a mild optimum around early layers rather than improving monotonically with more layers. Projection depth improves up to depth 18 and then saturates. • ***Alignment weight.*** Too weak an alignment signal underuses the pretrained representation target, while stronger weights do not further improve FID in this setting. Together, these trends suggest that ERW benefits from sufficient but moderate alignment capacity, rather than from maximizing the strength or depth of the auxiliary objective.

Table 7: **Effect of target representation model.** All models use SiT-B/2 and are trained for 50K iterations without classifier-free guidance on ImageNet 256×256. We fix $\lambda = 0.5$. Across pretrained encoders, ERW gives similar early training FID ranges, with DINOv2-B/L performing best in this sweep. $\downarrow$ and $\uparrow$ indicate whether lower or higher values are better.

Target Repr.	Depth	Objective	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
MoCov3-B	8	NT-Xent	61.1	**7.6**	22.38	0.42	**0.58**
MoCov3-L	8	NT-Xent	73.0	8.0	17.96	0.38	0.52
CLIP-L	8	NT-Xent	58.9	7.7	23.68	**0.44**	0.54
DINOv2-B	8	NT-Xent	55.6	7.8	25.45	**0.44**	0.56
DINOv2-L	8	NT-Xent	**55.5**	7.8	25.45	**0.44**	0.56
DINOv2-g	8	NT-Xent	59.4	**7.6**	**25.53**	**0.44**	0.56

Representation diagnostics. Finally, we measure layer-wise representation quality using class-conditional k-Nearest Neighbor Accuracy (CKNNA) [2], which indicates how well hidden features capture class discriminability. • ***Early-layer feature quality.*** ERW yields higher CKNNA scores in early layers, indicating that clean latent warmup improves the representation state used by later diffusion training. • ***Connection to representation cold start.*** The layer-wise CKNNA results provide representation-level evidence for the cold start claim: the gain appears in the early layers before full diffusion training, not only in the final FID.

Takeaway. The ablations point to the same mechanism: warmup is most effective in early layers, while target encoder choice, projection depth, and alignment weight are secondary design factors. Layer-wise CKNNA provides representation-level evidence that clean latent warmup improves the early features used by later diffusion training.

5 Conclusion and Future Work

In this work, we introduce Embedded Representation Warmup (ERW), a representation preconditioning strategy for diffusion transformer training. Rather than using pretrained representations only as training-time regularization, ERW first aligns early diffusion layers on clean latents and then continues standard diffusion and alignment training from the resulting initialization. Empirically, this change in optimization order improves matched-budget convergence: ERW reaches FID=1.41 at 350 epochs on ImageNet 256×256, remains competitive on ImageNet 512×512, and transfers to

MS-COCO text-to-image generation. Our ablations further indicate that preconditioning the earliest layers is most effective, supporting the view that ERW improves early-layer representations before full diffusion training begins.

Acknowledgments and Disclosures

This work was supported in part by the National Science and Technology Major Project (No. 2022ZD0115101), Research Center for Industries of the Future (RCIF) at Westlake University, Westlake Education Foundation, and Westlake University Center for High-performance Computing. The authors declare no competing interests.

References

[1] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE International Conference on Computer Vision*, 2021.

[3] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations*, 2024.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*, 2024.

[7] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *IEEE International Conference on Computer Vision*, 2021.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

[10] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

[11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

[12] Wan-Cyuan Fan, Yen-Chun Chen, DongDong Chen, Yu Cheng, Lu Yuan, and Yu-Chiang Frank Wang. Frido: Feature pyramid diffusion for complex scene image synthesis. In *AAAI Conference on Artificial Intelligence*, 2023.

[13] Michael Fuest, Pingchuan Ma, Ming Gui, Johannes S Fischer, Vincent Tao Hu, and Bjorn Ommer. Diffusion models and representation learning: A survey. *arXiv preprint arXiv:2407.00783*, 2024.

[14] Michael Fuest, Pingchuan Ma, Ming Gui, Johannes Schusterbauer, Vincent Tao Hu, and Björn Ommer. Diffusion models and representation learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[15] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. MDTv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023.

[16] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, et al. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025.

[17] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[18] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. DiffiT: Diffusion vision transformers for image generation. In *European Conference on Computer Vision*, 2024.

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[21] Wenkun He, Yuchao Gu, Junyu Chen, Dongyun Zou, Yujun Lin, Zhekai Zhang, Haocheng Xi, Muyang Li, Ligeng Zhu, Jincheng Yu, et al. Dc-gen: Post-training diffusion acceleration with deeply compressed latent space. *arXiv preprint arXiv:2509.25180*, 2025.

[22] Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024.

[23] 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 *Advances in Neural Information Processing Systems*, 2017.

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[25] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[26] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, 2023.

[27] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. In *International Conference on Machine Learning*, 2024.

[28] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[29] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

[30] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. *Advances in Neural Information Processing Systems*, 2024.

[31] Diederik P Kingma. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[32] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019.

[33] Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Ensembling off-the-shelf models for GAN training. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[34] 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. *arXiv preprint arXiv:2404.07724*, 2024.

[35] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 2019.

[36] 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*, pages 18262–18272, 2025.

[37] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *IEEE International Conference on Computer Vision*, 2023.

[38] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024.

[39] Daiqing Li, Huan Ling, Amlan Kar, David Acuna, Seung Wook Kim, Karsten Kreis, Antonio Torralba, and Sanja Fidler. DreamTeacher: Pretraining image backbones with deep generative models. In *IEEE International Conference on Computer Vision*, 2023.

[40] T Li, D Katabi, and K He. Return of unconditional generation: A self-supervised representation generation method. In *Advances in Neural Information Processing Systems*, 2024.

[41] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[42] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. *arXiv preprint arXiv:2405.08748*, 2024.

[43] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[44] Deyuan Liu, Peng Sun, Yansen Han, Zhenglin Cheng, Chuyan Chen, and Tao Lin. Self-adversarial one step generation via condition shifting. *arXiv preprint arXiv:2604.12322*, 2026.

[45] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

[46] I Loshchilov. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

[47] Cheng Lu, Yuhao Zhou, Fan Bao, Jian Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[48] 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. In *European Conference on Computer Vision (ECCV)*, 2024.

[49] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. In *International Conference on Machine Learning*, 2021.

[50] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[51] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[52] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[55] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022.

[56] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 2016.

[57] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[58] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[59] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected GANs converge faster. *Advances in Neural Information Processing Systems*, 2021.

[60] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

[61] Vikash Sehwag, Xianghao Kong, Jingtao Li, Michael Spranger, and Lingjuan Lyu. Stretching each dollar: Diffusion training from scratch on a micro-budget. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28596–28608, 2025.

[62] Shitong Shao, Xu Dai, Lujun Li, Huanran Chen, Yang Hu, and Shouyi Yin. Catch-up distillation: You only need to train once for accelerating sampling. *arXiv preprint arXiv:2305.10769*, 2023.

[63] Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think. *arXiv preprint arXiv:2603.14366*, 2026.

[64] Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? *arXiv preprint arXiv:2512.10794*, 2025.

[65] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[66] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[67] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[68] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[69] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. DF-GAN: A simple and effective baseline for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16515–16525, 2022.

[70] Z-Image Team. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

[71] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[72] Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *arXiv preprint arXiv:2507.01467*, 2025.

[73] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *IEEE International Conference on Computer Vision*, 2023.

[74] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[75] Tianshuo Xu, Peng Mi, Ruilin Wang, and Yingcong Chen. Towards faster training of diffusion models: An inspiration of a consistency phenomenon. *arXiv preprint arXiv:2404.07946*, 2024.

[76] Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *IEEE International Conference on Computer Vision*, 2023.

[77] 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. *arXiv preprint arXiv:2410.06940*, 2024.

[78] Zhihang Yuan, Hanling Zhang, Pu Lu, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. Ditfastattn: Attention compression for diffusion transformer models. *Advances in Neural Information Processing Systems*, 37:1196–1219, 2024.

[79] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[80] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[81] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Transactions on Machine Learning Research*, 2024.

[82] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. LAFITE: Towards language-free training for text-to-image generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[83] Lianghui Zhu, Zilong Huang, Bencheng Liao, Jun Hao Liew, Hanshu Yan, Jiashi Feng, and Xinggang Wang. Dig: Scalable and efficient diffusion models with gated linear attention. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7664–7674, 2025.

[84] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[85] Rui Zhu, Yingwei Pan, Yehao Li, Ting Yao, Zhenglong Sun, Tao Mei, and Chang Wen Chen. SD-DiT: Unleashing the power of self-supervised discrimination in diffusion transformer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

Contents

1 Introduction **1**

2 Related Work **2**

3 Representation Preconditioning in Latent Diffusion **3**
3.1 Flow Matching and Representation Alignment 3
3.2 Alignment as Training Time Regularization 4
3.3 A Probabilistic View of the Alignment Signal 4
3.4 Clean Latent Alignment as Boundary Preconditioning 5
3.5 Choice of Early Layers 5
3.6 Two-Phase Training 5

4 Experiments **6**
4.1 Setup 6
4.2 Comparison 7
4.3 Scaling and Transfer 7
4.4 ERW Efficiency 8
4.5 Ablation Studies 8

5 Conclusion and Future Work **9**

A Limitations and Scope **17**

B Theoretical Analysis **18**
B.1 Setup 18
B.2 Augmented Space and Score Decomposition 18
B.3 NT-Xent as a Finite-Candidate Classification Likelihood 19
B.4 Conditional Variance Reduction via Representation 19
B.5 Representation Quality and Velocity Approximation 22

C Analysis Details **22**
C.1 CKNNA Metric Details 22
C.2 Description of pretrained visual encoders 23

D Hyperparameter and More Implementation Details **23**
D.1 Hyperparameter Tuning 23

E Evaluation Details **24**

F Baselines **25**

A Limitations and Scope

Our goal is to study the optimization role of representation alignment in diffusion transformer training. For this reason, the main controlled comparisons are SiT and REPA-style representation alignment, where the backbone, latent autoencoder, sampler, and representation target can be kept fixed. Many recent diffusion systems improve FID through better tokenizers, architectures, attention mechanisms, guidance schedules, solvers, or data recipes. Applying ERW on top of all such systems is an interesting direction, but it would also introduce additional factors that make it harder to isolate whether the gain comes from representation preconditioning. Thus, our claim is not that ERW is a complete state-of-the-art training stack, but that changing representation alignment from training-time regularization to preconditioning improves optimization efficiency in controlled alignment settings.

ERW is primarily designed for training acceleration. We count the clean latent warmup in the total training budget, and report both matched-budget results and post-warmup convergence. The warmup is cheaper than full diffusion training because it updates only early layers with a clean latent alignment objective, but the exact wall-clock benefit can depend on implementation details such as teacher feature computation, caching, hardware utilization, and data loading.

Our experiments cover ImageNet 256×256, ImageNet 512×512, and an MS-COCO text-to-image setting. These results provide evidence beyond the main ImageNet 256×256 setting, but they do not fully characterize ERW for very large text-to-image foundation models, video generation, multi-stage high-resolution pipelines, or other modalities. Studying these larger-scale settings is left for future work.

Finally, ERW relies on pretrained visual representations and on the empirical observation that early layers are effective targets for alignment. Our placement ablations support this choice, and prior alignment work reports a similar trend, but we do not claim a strict modular decomposition of the diffusion backbone. The best warmup length, projection depth, alignment weight, and target encoder may vary across architectures and datasets.

B Theoretical Analysis

We formalize the augmented probability space used in the main text, the finite-candidate interpretation of NT-Xent, and an oracle MMSE comparison. These results motivate representation preconditioning under stated assumptions; they do not establish a convergence rate for the practical training algorithm.

B.1 Setup

Let $\mathcal{X}$ be the data space, $\mathcal{Z}$ the latent space from a VAE, and $\mathcal{R}$ the representation space. We use the following notation throughout:

- $\mathbf{x} \sim p(\mathbf{x})$, $\mathbf{z}_0 = \mathcal{H}(\mathbf{x}) \in \mathcal{Z}$, $\mathbf{r} = \mathbf{f}_{\text{rep}}(\mathbf{x}) \in \mathcal{R}$.
- $\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $t \in [0, 1]$, with $(\alpha_0, \sigma_0) = (1, 0)$, $(\alpha_1, \sigma_1) = (0, 1)$.
- $\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$: velocity model; $\mathcal{R}_{\boldsymbol{\theta}_{\text{early}}}$: early-layer subnetwork; $\mathcal{T}_{\boldsymbol{\theta}}$: projection head.
- $\mathcal{D}$: the VAE decoder mapping latents back to pixel space, i.e., $\mathbf{x} \approx \mathcal{D}(\mathbf{z}_0)$.

The VAE encoder $\mathcal{H}$, decoder $\mathcal{D}$, and teacher $\mathbf{f}_{\text{rep}}$ are fixed; $\boldsymbol{\theta}$ denotes trainable diffusion and projection parameters. The velocity target is $\mathbf{v} := \dot{\alpha}_t \mathbf{z}_0 + \dot{\sigma}_t \boldsymbol{\epsilon}$. The noise $\boldsymbol{\epsilon}$ is independent of the data and time. In the variance analysis we fix t and take outer expectations over the data and noise, so that $\Delta(t)$ is a function of this fixed noise level. We assume finite second moments wherever MMSEs are used.

B.2 Augmented Space and Score Decomposition

The data and fixed encoders induce a joint distribution of $(\mathbf{z}_0, \mathbf{r}) = (\mathcal{H}(\mathbf{x}), \mathbf{f}_{\text{rep}}(\mathbf{x}))$. We retain its general conditional distribution $p(\mathbf{r} \mid \mathbf{z}_0)$: approximate VAE reconstruction does not imply $\mathbf{r} = \mathbf{f}_{\text{rep}}(\mathcal{D}(\mathbf{z}_0))$. The augmented joint distribution is

$$p(\mathbf{z}_0, \mathbf{z}_t, \mathbf{r}, t) = p(\mathbf{z}_t \mid \mathbf{z}_0, t)\, p(\mathbf{r} \mid \mathbf{z}_0)\, p(\mathbf{z}_0)\, p(t). \tag{15}$$

For $\sigma_t > 0$, the forward kernel is $\mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{z}_0, \sigma_t^2 \mathbf{I})$; at $t = 0$ it is the clean-latent point mass. Here $p(t) = \mathcal{U}[0, 1]$ and $\mathbf{z}_t \perp\!\!\!\perp \mathbf{r} \mid (\mathbf{z}_0, t)$ follows from independent forward noise. Integrating out $\mathbf{z}_0$ gives

$$p(\mathbf{z}_t, \mathbf{r}, t) = p(t) \int p(\mathbf{z}_t \mid \mathbf{z}_0, t)\, p(\mathbf{r} \mid \mathbf{z}_0)\, p(\mathbf{z}_0)\, d\mathbf{z}_0. \tag{16}$$

Integrals over representations are understood with respect to the induced conditional measure when no ordinary density exists. Integrating out $\mathbf{r}$ recovers the standard marginal

$$p(\mathbf{z}_t \mid t) = \int p(\mathbf{z}_t \mid \mathbf{z}_0, t)\, p(\mathbf{z}_0)\, d\mathbf{z}_0. \tag{17}$$

Likewise, the tower property yields Eq. (9). Neither construction changes the data distribution or forward process. This probabilistic identity does not imply that adding alignment to a finite model's optimization leaves its optimum unchanged.

Assumption (Semantic Sufficiency). For the semantic interpretation of alignment, we assume that $p(\mathbf{z}_0 \mid \mathbf{r})$ concentrates on latents with consistent high-level content. This is a qualitative assumption about representation quality, not a consequence of the score identity below. The non-increasing MMSE result in Proposition 2 does not require semantic sufficiency; a strict decrease requires a change in the conditional velocity mean.

Theorem 1 (Conditional Log-Density Decomposition) . *Where the displayed conditional log-probabilities are defined and differentiable in $\mathbf{z}_t$,*

$$\begin{aligned} \nabla_{\mathbf{z}_t} \log p(\mathbf{z}_0, \mathbf{r} \mid \mathbf{z}_t, t) = \nabla_{\mathbf{z}_t} \log p(\mathbf{z}_0 \mid \mathbf{z}_t, \mathbf{r}, t) \\ + \nabla_{\mathbf{z}_t} \log p(\mathbf{r} \mid \mathbf{z}_t, t). \end{aligned} \tag{18}$$

Proof. The probability chain rule gives $p(\mathbf{z}_0, \mathbf{r} \mid \mathbf{z}_t, t) = p(\mathbf{z}_0 \mid \mathbf{z}_t, \mathbf{r}, t) p(\mathbf{r} \mid \mathbf{z}_t, t)$. Taking its logarithm and differentiating gives Eq. (18). □

We refer to the two terms as generation and representation components. The identity motivates an auxiliary representation-learning objective, but is not an equivalence between the practical flow-matching/alignment losses and exact estimation of these components. At the clean boundary, we use the finite-candidate objective below, without requiring differentiation of a singular conditional density.

B.3 NT-Xent as a Finite-Candidate Classification Likelihood

Phase 1 retains NT-Xent as its alignment objective. Its exact likelihood interpretation is conditional on the given in-batch candidate set.

Proposition 1 (Finite-Candidate Likelihood for NT-Xent) . *Let $\mathcal{B} = (\mathbf{r}_1, \ldots, \mathbf{r}_N)$ be an indexed set of in-batch candidate representations, with positive index i for $\mathbf{z}_0$. Write $\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_0) = \mathcal{T}_{\boldsymbol{\theta}}(\mathcal{R}_{\boldsymbol{\theta}_{\text{early}}}(\mathbf{z}_0, 0))$. Define the categorical model*

$$q_{\boldsymbol{\theta}}(j \mid \mathbf{z}_0, \mathcal{B}) = \frac{\exp(\operatorname{sim}(\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_0), \mathbf{r}_j)/\tau_{\mathrm{c}})}{\sum_{\ell=1}^{N} \exp(\operatorname{sim}(\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_0), \mathbf{r}_\ell)/\tau_{\mathrm{c}})}, \quad j = 1, \ldots, N, \tag{19}$$

where sim *is cosine similarity on nonzero feature vectors and $\tau_{\mathrm{c}} > 0$ is the contrastive temperature. For these candidates, NT-Xent equals*

$$\ell_{\text{NT-Xent}}(\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_0), \mathbf{r}_i; \mathcal{B}) = -\log q_{\boldsymbol{\theta}}(i \mid \mathbf{z}_0, \mathcal{B}). \tag{20}$$

The positive-index classification likelihood is therefore maximized by minimizing this loss.

Proof. Expanding the negative log-probability gives

$$\begin{aligned} -\log q_{\boldsymbol{\theta}}(i \mid \mathbf{z}_0, \mathcal{B}) = & -\frac{\operatorname{sim}(\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_0), \mathbf{r}_i)}{\tau_{\mathrm{c}}} \\ & + \log \sum_{\ell=1}^{N} \exp\left(\frac{\operatorname{sim}(\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_0), \mathbf{r}_\ell)}{\tau_{\mathrm{c}}}\right), \end{aligned} \tag{21}$$

which is the NT-Xent loss for the specified positive index and candidates. Taking the expectation over sampled batches and positive pairs preserves this equality, and hence the optimizing parameter sets coincide. □

The candidate-set dependence is suppressed in the main-text loss notation, and batch sampling is included in its expectation. The categorical model $q_{\boldsymbol{\theta}}(j \mid \mathbf{z}_0, \mathcal{B})$ is not a density over all continuous representations. Thus, this equality alone does not establish maximum likelihood for $p(\mathbf{r} \mid \mathbf{z}_0)$, or recovery of its true score. NT-Xent supplies a finite-candidate representation-learning surrogate at the clean boundary. The temperature τ_{c} is distinct from the alignment-weight decay constant τ_{d}.

B.4 Conditional Variance Reduction via Representation

We compare an oracle with access to the true representation $\mathbf{r}$ against an estimator given only $(\mathbf{z}_t, t)$. Additional conditioning cannot increase optimal squared error. This comparison quantifies potentially useful side information, without asserting that the practical model has access to it at inference.

Proposition 2 (Oracle MMSE with Additional Representation Information) . *Fix t and assume $\mathbb{E}[\|\mathbf{v}\|^2] < \infty$ for $\mathbf{v} = \dot{\alpha}_t \mathbf{z}_0 + \dot{\sigma}_t \boldsymbol{\epsilon}$. The minimum achievable MSE satisfies*

$$\mathbb{E}\left[\|\mathbf{v} - \mathbb{E}\left[\mathbf{v} \mid \mathbf{z}_t, \mathbf{r}, t\right]\|^2\right] \;\le\; \mathbb{E}\left[\|\mathbf{v} - \mathbb{E}\left[\mathbf{v} \mid \mathbf{z}_t, t\right]\|^2\right] \tag{22}$$

with equality if and only if $\mathbb{E}[\mathbf{v} \mid \mathbf{z}_t, \mathbf{r}, t] = \mathbb{E}[\mathbf{v} \mid \mathbf{z}_t, t]$ almost surely. Conditional independence $\mathbf{v} \perp\!\!\!\perp \mathbf{r} \mid (\mathbf{z}_t, t)$ is sufficient but is not required. The inequality is strict when these conditional means differ on a set of positive probability. The gap is

$$\Delta(t) := \mathbb{E}\left[\operatorname{Tr}\big(\operatorname{Cov}(\mathbb{E}\left[\mathbf{v} \mid \mathbf{z}_t, \mathbf{r}, t\right] \mid \mathbf{z}_t, t)\big)\right] \;\ge\; 0, \tag{23}$$

which measures variation in the conditional velocity mean beyond what $(\mathbf{z}_t, t)$ determines.

Proof. **Step 1: MMSE characterization.** For any random vector $\mathbf{v}$ and σ-algebra $\mathcal{G}$, the conditional expectation $\mathbb{E}\left[\mathbf{v} \mid \mathcal{G}\right]$ is the MMSE estimator:

$$\mathbb{E}\left[\mathbf{v} \mid \mathcal{G}\right] = \operatorname*{arg\,min}_{g \in L^2(\mathcal{G})} \mathbb{E}\left[\|\mathbf{v} - g\|^2\right]. \tag{24}$$

The corresponding MSE equals the expected trace of the conditional covariance:

$$\mathbb{E}\left[\|\mathbf{v}-\mathbb{E}\left[\mathbf{v}\mid\mathcal{G}\right]\|^2\right]=\mathbb{E}\left[\sum_{i=1}^{d}\mathrm{Var}(v_i\mid\mathcal{G})\right]=\mathbb{E}\left[\mathrm{Tr}(\mathrm{Cov}(\mathbf{v}\mid\mathcal{G}))\right]. \tag{25}$$

Applied to $\mathcal{G}=\sigma(\mathbf{z}_t,t)$, this gives the optimal MSE without $\mathbf{r}$; applied to $\mathcal{G}=\sigma(\mathbf{z}_t,\mathbf{r},t)$, it gives the optimal MSE with $\mathbf{r}$.

Step 2: Law of total variance. For nested σ-algebras $\mathcal{G}_1:=\sigma(\mathbf{z}_t,t)\subset\mathcal{G}_2:=\sigma(\mathbf{z}_t,\mathbf{r},t)$, we derive the law of total variance. Define $\boldsymbol{\mu}:=\mathbb{E}\left[\mathbf{v}\mid\mathcal{G}_2\right]$ and $\bar{\boldsymbol{\mu}}:=\mathbb{E}\left[\mathbf{v}\mid\mathcal{G}_1\right]=\mathbb{E}\left[\boldsymbol{\mu}\mid\mathcal{G}_1\right]$ (tower property). Write the residual decomposition $\mathbf{v}-\bar{\boldsymbol{\mu}}=(\mathbf{v}-\boldsymbol{\mu})+(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})$. Taking the outer product and conditioning on $\mathcal{G}_1$:

$$\begin{aligned}\mathrm{Cov}(\mathbf{v}\mid\mathcal{G}_1)&=\mathbb{E}\left[(\mathbf{v}-\bar{\boldsymbol{\mu}})(\mathbf{v}-\bar{\boldsymbol{\mu}})^\top\mid\mathcal{G}_1\right]\\&=\mathbb{E}\left[(\mathbf{v}-\boldsymbol{\mu})(\mathbf{v}-\boldsymbol{\mu})^\top\mid\mathcal{G}_1\right]+\mathbb{E}\left[(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^\top\mid\mathcal{G}_1\right]\\&\quad+\mathbb{E}\left[(\mathbf{v}-\boldsymbol{\mu})(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^\top\mid\mathcal{G}_1\right]+\mathbb{E}\left[(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})(\mathbf{v}-\boldsymbol{\mu})^\top\mid\mathcal{G}_1\right].\end{aligned} \tag{26}$$

The cross terms vanish: since $\boldsymbol{\mu}-\bar{\boldsymbol{\mu}}$ is $\mathcal{G}_2$-measurable and $\mathbb{E}\left[\mathbf{v}-\boldsymbol{\mu}\mid\mathcal{G}_2\right]=\mathbf{0}$, the tower property gives $\mathbb{E}\left[(\mathbf{v}-\boldsymbol{\mu})(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^\top\mid\mathcal{G}_1\right]=\mathbb{E}\left[\mathbb{E}\left[(\mathbf{v}-\boldsymbol{\mu})\mid\mathcal{G}_2\right](\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^\top\mid\mathcal{G}_1\right]=\mathbf{0}$. The remaining two terms are:

$$\mathbb{E}\left[(\mathbf{v}-\boldsymbol{\mu})(\mathbf{v}-\boldsymbol{\mu})^\top\mid\mathcal{G}_1\right]=\mathbb{E}\left[\mathrm{Cov}(\mathbf{v}\mid\mathcal{G}_2)\mid\mathcal{G}_1\right], \tag{27}$$

$$\mathbb{E}\left[(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^\top\mid\mathcal{G}_1\right]=\mathrm{Cov}(\boldsymbol{\mu}\mid\mathcal{G}_1)=\mathrm{Cov}(\mathbb{E}\left[\mathbf{v}\mid\mathcal{G}_2\right]\mid\mathcal{G}_1). \tag{28}$$

Substituting back into (26):

$$\mathrm{Cov}(\mathbf{v}\mid\mathcal{G}_1)=\mathbb{E}\left[\mathrm{Cov}(\mathbf{v}\mid\mathcal{G}_2)\mid\mathcal{G}_1\right]+\mathrm{Cov}\big(\mathbb{E}\left[\mathbf{v}\mid\mathcal{G}_2\right]\,\big|\,\mathcal{G}_1\big). \tag{29}$$

Both terms on the right are positive semidefinite (as covariance matrices).

Step 3: Trace and expectation. Taking $\mathrm{Tr}(\cdot)$ on both sides of (29):

$$\mathrm{Tr}(\mathrm{Cov}(\mathbf{v}\mid\mathbf{z}_t,t))=\mathrm{Tr}(\mathbb{E}\left[\mathrm{Cov}(\mathbf{v}\mid\mathbf{z}_t,\mathbf{r},t)\mid\mathbf{z}_t,t\right])+\mathrm{Tr}(\mathrm{Cov}(\mathbb{E}\left[\mathbf{v}\mid\mathbf{z}_t,\mathbf{r},t\right]\mid\mathbf{z}_t,t)). \tag{30}$$

Since $\mathrm{Tr}(\mathbb{E}\left[\mathbf{A}\mid\mathcal{G}\right])=\mathbb{E}\left[\mathrm{Tr}(\mathbf{A})\mid\mathcal{G}\right]$ for any random matrix $\mathbf{A}$, the first term equals $\mathbb{E}\left[\mathrm{Tr}(\mathrm{Cov}(\mathbf{v}\mid\mathbf{z}_t,\mathbf{r},t))\mid\mathbf{z}_t,t\right]$. Taking outer expectations over $\mathbf{z}_t$ at the fixed t:

$$\underbrace{\mathbb{E}\left[\mathrm{Tr}(\mathrm{Cov}(\mathbf{v}\mid\mathbf{z}_t,t))\right]}_{\text{MSE without }\mathbf{r}}=\underbrace{\mathbb{E}\left[\mathrm{Tr}(\mathrm{Cov}(\mathbf{v}\mid\mathbf{z}_t,\mathbf{r},t))\right]}_{\text{MSE with }\mathbf{r}}+\underbrace{\mathbb{E}\left[\mathrm{Tr}(\mathrm{Cov}(\mathbb{E}\left[\mathbf{v}\mid\mathbf{z}_t,\mathbf{r},t\right]\mid\mathbf{z}_t,t))\right]}_{\Delta(t)\geq 0}. \tag{31}$$

Rearranging yields (22). By the tower property, $\Delta(t)=\mathbb{E}[\|\mathbb{E}[\mathbf{v}\mid\mathbf{z}_t,\mathbf{r},t]-\mathbb{E}[\mathbf{v}\mid\mathbf{z}_t,t]\|^2]$. It vanishes exactly when the two conditional means agree almost surely, establishing the stated equality condition. □

Corollary 1 (Velocity-Space Form under OT Path) . *For fixed $0<t\leq 1$ under the OT interpolation $\alpha_t=1-t$, $\sigma_t=t$, the velocity target is $\mathbf{v}=\boldsymbol{\epsilon}-\mathbf{z}_0=(\mathbf{z}_t-\mathbf{z}_0)/t$. The conditional variance decomposes as*

$$\mathrm{Cov}(\mathbf{v}\mid\mathbf{z}_t,t)=\frac{1}{t^2}\,\mathrm{Cov}(\mathbf{z}_0\mid\mathbf{z}_t,t), \tag{32}$$

so the variance reduction gap becomes

$$\Delta(t)=\frac{1}{t^2}\,\mathbb{E}\left[\mathrm{Tr}\big(\mathrm{Cov}(\mathbb{E}\left[\mathbf{z}_0\mid\mathbf{z}_t,\mathbf{r},t\right]\mid\mathbf{z}_t,t)\big)\right]. \tag{33}$$

Proof. **Step 1: Express $\mathbf{v}$ in terms of $\mathbf{z}_0$.** Under the OT path, $\mathbf{z}_t=(1-t)\mathbf{z}_0+t\boldsymbol{\epsilon}$, so

$$\mathbf{v}=\dot{\alpha}_t\mathbf{z}_0+\dot{\sigma}_t\boldsymbol{\epsilon}=-\mathbf{z}_0+\boldsymbol{\epsilon}. \tag{34}$$

Solving $\boldsymbol{\epsilon}=(\mathbf{z}_t-(1-t)\mathbf{z}_0)/t$ and substituting:

$$\mathbf{v}=-\mathbf{z}_0+\frac{\mathbf{z}_t-(1-t)\mathbf{z}_0}{t}=\frac{-t\mathbf{z}_0+\mathbf{z}_t-(1-t)\mathbf{z}_0}{t}=\frac{\mathbf{z}_t-\mathbf{z}_0}{t}. \tag{35}$$

Step 2: Compute the conditional covariance. Given $(\mathbf{z}_t, t)$, the quantity $\mathbf{z}_t$ is a constant. Therefore:

$$\mathrm{Cov}(\mathbf{v} \mid \mathbf{z}_t, t) = \mathrm{Cov}\left(\frac{\mathbf{z}_t - \mathbf{z}_0}{t} \,\middle|\, \mathbf{z}_t, t\right) = \frac{1}{t^2}\,\mathrm{Cov}(-\mathbf{z}_0 \mid \mathbf{z}_t, t) = \frac{1}{t^2}\,\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_t, t), \tag{36}$$

where the last equality uses $\mathrm{Cov}(-\mathbf{x}) = \mathrm{Cov}(\mathbf{x})$.

Step 3: Apply the variance decomposition to $\mathbf{z}_0$. By the law of total variance on $\mathbf{z}_0$ with $\sigma(\mathbf{z}_t, t) \subset \sigma(\mathbf{z}_t, \mathbf{r}, t)$:

$$\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_t, t) = \mathbb{E}\left[\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_t, \mathbf{r}, t) \mid \mathbf{z}_t, t\right] + \mathrm{Cov}(\mathbb{E}\left[\mathbf{z}_0 \mid \mathbf{z}_t, \mathbf{r}, t\right] \mid \mathbf{z}_t, t). \tag{37}$$

Multiplying both sides by t^{-2}, taking traces, and taking outer expectations:

$$\Delta(t) = \frac{1}{t^2}\,\mathbb{E}\left[\mathrm{Tr}\big(\mathrm{Cov}(\mathbb{E}\left[\mathbf{z}_0 \mid \mathbf{z}_t, \mathbf{r}, t\right] \mid \mathbf{z}_t, t)\big)\right], \tag{38}$$

which is (33). □

Connection to the denoising score via Tweedie's formula. For $0 < t < 1$ under the OT path, and where the conditional scores exist, the posterior mean $\mathbb{E}\left[\mathbf{z}_0 \mid \mathbf{z}_t, t\right]$ relates to the score function through Tweedie's formula. Since $\mathbf{z}_t \mid \mathbf{z}_0 \sim \mathcal{N}((1-t)\mathbf{z}_0,\, t^2\mathbf{I})$, we have:

$$\mathbb{E}\left[\mathbf{z}_0 \mid \mathbf{z}_t, t\right] = \frac{\mathbf{z}_t + t^2\,\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t)}{1 - t}. \tag{39}$$

To derive this, write the joint: $p(\mathbf{z}_t, \mathbf{z}_0) = p(\mathbf{z}_t \mid \mathbf{z}_0)\,p(\mathbf{z}_0)$. The marginal score satisfies $\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t) = \mathbb{E}\left[\nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t \mid \mathbf{z}_0) \mid \mathbf{z}_t\right]$. Since $\log p(\mathbf{z}_t \mid \mathbf{z}_0) = -\left\|\mathbf{z}_t - (1-t)\mathbf{z}_0\right\|^2/(2t^2) + \mathrm{const}$, we get $\nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t \mid \mathbf{z}_0) = -(\mathbf{z}_t - (1-t)\mathbf{z}_0)/t^2$. Taking conditional expectations:

$$\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t) = \mathbb{E}\left[-\frac{\mathbf{z}_t - (1-t)\mathbf{z}_0}{t^2} \,\middle|\, \mathbf{z}_t\right] = \frac{(1-t)\mathbb{E}\left[\mathbf{z}_0 \mid \mathbf{z}_t\right] - \mathbf{z}_t}{t^2}. \tag{40}$$

Rearranging gives (39). Similarly, when conditioning on $\mathbf{r}$:

$$\mathbb{E}\left[\mathbf{z}_0 \mid \mathbf{z}_t, \mathbf{r}, t\right] = \frac{\mathbf{z}_t + t^2\,\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t \mid \mathbf{r})}{1 - t}. \tag{41}$$

The variance gap $\Delta(t)$ from (33) can therefore be expressed in terms of the score difference:

$$\begin{aligned}\Delta(t) &= \frac{1}{t^2}\,\mathbb{E}\left[\mathrm{Tr}\big(\mathrm{Cov}(\mathbb{E}\left[\mathbf{z}_0 \mid \mathbf{z}_t, \mathbf{r}, t\right] \mid \mathbf{z}_t, t)\big)\right] \\ &= \frac{t^2}{(1-t)^2}\,\mathbb{E}\left[\mathrm{Tr}\big(\mathrm{Cov}(\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t \mid \mathbf{r}) \mid \mathbf{z}_t, t)\big)\right],\end{aligned} \tag{42}$$

where the second line substitutes (41) and uses $\mathrm{Cov}(a + b\mathbf{x}) = b^2\mathrm{Cov}(\mathbf{x})$. This shows that $\Delta(t)$ measures the variability of the *conditional score* $\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t \mid \mathbf{r})$ across different representations $\mathbf{r}$, weighted by $t^2/(1-t)^2$. The identity does not imply that $\Delta(t)$ is monotone in t, and does not determine the optimization benefit of warmup. The formulas with t^{-1} or $(1-t)^{-1}$ are used only on their stated domains, not at their singular endpoints.

B.5 Representation Quality and Velocity Approximation

The oracle result assumes access to the true representation $\mathbf{r}$. The following Lipschitz bound describes a hypothetical conditional predictor supplied with an approximate representation $\hat{\mathbf{r}}$. Its assumptions concern representation quality at the fixed noise level under consideration.

Proposition 3 (Approximation Error from Imperfect Representations) . *Fix t. Suppose the optimal conditional velocity $\mathbf{v}^*(\mathbf{z}_t, \mathbf{r}, t) := \mathbb{E}[\mathbf{v} \mid \mathbf{z}_t, \mathbf{r}, t]$ has a version that is L-Lipschitz in $\mathbf{r}$, with the same constant L for almost every $\mathbf{z}_t$:*

$$\|\mathbf{v}^*(\mathbf{z}_t, \mathbf{r}, t) - \mathbf{v}^*(\mathbf{z}_t, \mathbf{r}', t)\| \le L \|\mathbf{r} - \mathbf{r}'\| \tag{43}$$

for all evaluated $\mathbf{r}, \mathbf{r}' \in \mathcal{R}$, including the model estimates below. If the early layers produce an estimate $\hat{\mathbf{r}} = \mathcal{T}_{\boldsymbol{\theta}}(\mathcal{R}_{\boldsymbol{\theta}_{\text{early}}}(\mathbf{z}_t, t))$ satisfying $\mathbb{E}\left[\|\hat{\mathbf{r}} - \mathbf{r}\|^2\right] \le \varepsilon^2$, then

$$\mathbb{E}\left[\|\mathbf{v}^*(\mathbf{z}_t, \hat{\mathbf{r}}, t) - \mathbf{v}^*(\mathbf{z}_t, \mathbf{r}, t)\|^2\right] \le L^2 \varepsilon^2. \tag{44}$$

Proof. **Step 1: Apply the Lipschitz condition pointwise.** For any realization of $(\mathbf{z}_t, t, \mathbf{r}, \hat{\mathbf{r}})$:

$$\|\mathbf{v}^*(\mathbf{z}_t, \hat{\mathbf{r}}, t) - \mathbf{v}^*(\mathbf{z}_t, \mathbf{r}, t)\| \le L \|\hat{\mathbf{r}} - \mathbf{r}\| . \tag{45}$$

Squaring both sides (valid since both sides are non-negative):

$$\|\mathbf{v}^*(\mathbf{z}_t, \hat{\mathbf{r}}, t) - \mathbf{v}^*(\mathbf{z}_t, \mathbf{r}, t)\|^2 \le L^2 \|\hat{\mathbf{r}} - \mathbf{r}\|^2 . \tag{46}$$

Step 2: Take expectations. Since (46) holds pointwise (almost surely), taking expectations preserves the inequality:

$$\mathbb{E}\left[\|\mathbf{v}^*(\mathbf{z}_t, \hat{\mathbf{r}}, t) - \mathbf{v}^*(\mathbf{z}_t, \mathbf{r}, t)\|^2\right] \le L^2 \mathbb{E}\left[\|\hat{\mathbf{r}} - \mathbf{r}\|^2\right] \le L^2 \varepsilon^2, \tag{47}$$

which establishes (44). □

Total error decomposition. Let $\mathbf{v}^*(\mathbf{z}_t, \mathbf{r}, t) = \mathbb{E}[\mathbf{v} \mid \mathbf{z}_t, \mathbf{r}, t]$, let $\hat{\mathbf{r}}$ be the model's deterministic representation estimate, and assume all terms below have finite second moments. Conditional-expectation orthogonality gives

$$\begin{aligned}\mathbb{E}[\|\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z}_t, t) - \mathbf{v}\|^2] = \underbrace{\mathbb{E}[\|\mathbf{v} - \mathbf{v}^*(\mathbf{z}_t, \mathbf{r}, t)\|^2]}_{\text{oracle irreducible error}} \\ + \mathbb{E}[\|\mathbf{v}^*(\mathbf{z}_t, \mathbf{r}, t) - \mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)\|^2].\end{aligned} \tag{48}$$

Decomposing the second residual through $\mathbf{v}^*(\mathbf{z}_t, \hat{\mathbf{r}}, t)$, applying $\|\mathbf{a} + \mathbf{b}\|^2 \le 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$, and using Proposition 3 yields

$$\mathbb{E}[\|\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z}_t, t) - \mathbf{v}\|^2] \le \mathbb{E}[\|\mathbf{v} - \mathbf{v}^*(\mathbf{z}_t, \mathbf{r}, t)\|^2] + 2L^2\varepsilon^2 \tag{49}$$

$$+ 2\mathbb{E}[\|\mathbf{v}^*(\mathbf{z}_t, \hat{\mathbf{r}}, t) - \mathbf{F}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)\|^2]. \tag{50}$$

Only cross terms involving the oracle residual vanish by orthogonality; the two remaining approximation terms need not be orthogonal. The bound assumes an L-Lipschitz version of $\mathbf{v}^*$ on the representations being evaluated, including $\hat{\mathbf{r}}$. Clean-latent alignment at $t = 0$ alone does not establish a small ε at arbitrary noisy inputs, nor guarantee faster convergence. Warmup is motivated as a way to improve initialization; its practical benefit is assessed empirically.

C Analysis Details

C.1 CKNNA Metric Details

CKNNA (Centered Kernel Nearest Neighbor Alignment) is a *relaxed version* of the popular Centered Kernel Alignment (CKA; [32]) that mitigates the strict definition of alignment. We generally follow the notations in the original paper for an explanation [27].

First, CKA has measured *global* similarities of the models by considering all possible data pairs:

$$\text{CKA}(\mathbf{K}, \mathbf{L}) = \frac{\text{HSIC}(\mathbf{K}, \mathbf{L})}{\sqrt{\text{HSIC}(\mathbf{K}, \mathbf{K})\text{HSIC}(\mathbf{L}, \mathbf{L})}}, \tag{51}$$

where $\mathbf{K}$ and $\mathbf{L}$ are two kernel matrices computed from the dataset using two different networks. Specifically, it is defined as $\mathbf{K}_{ij} = \kappa(\phi_i, \phi_j)$ and $\mathbf{L}_{ij} = \kappa(\psi_i, \psi_j)$ where ϕ_i, ϕ_j and ψ_i, ψ_j are representations computed from each network at the corresponding data $\mathbf{x}_i, \mathbf{x}_j$ (respectively). By letting κ as a inner product kernel, HSIC is defined as

$$\mathrm{HSIC}(\mathbf{K}, \mathbf{L}) = \frac{1}{(n-1)^2}\Big(\sum_i \sum_j \big(\langle\phi_i, \phi_j\rangle - \mathbb{E}_l[\langle\phi_i, \phi_l\rangle]\big)\big(\langle\psi_i, \psi_j\rangle - \mathbb{E}_l[\langle\psi_i, \psi_l\rangle]\big)\Big). \quad (52)$$

CKNNA considers a relaxed version of Eq. (51) by replacing $\mathrm{HSIC}(\mathbf{K}, \mathbf{L})$ into $\mathrm{Align}(\mathbf{K}, \mathbf{L})$, where $\mathrm{Align}(\mathbf{K}, \mathbf{L})$ computes Eq. (52) only using a k nearest neighbor embedding in the datasets:

$$\mathrm{Align}(\mathbf{K}, \mathbf{L}) = \frac{1}{(n-1)^2}\Big(\sum_i \sum_j \alpha(i,j)\big(\langle\phi_i, \phi_j\rangle - \mathbb{E}_l[\langle\phi_i, \phi_l\rangle]\big)\big(\langle\psi_i, \psi_j\rangle - \mathbb{E}_l[\langle\psi_i, \psi_l\rangle]\big)\Big), \quad (53)$$

where $\alpha(i, j)$ is defined as

$$\alpha(i, j; k) = \mathbb{1}[i \neq j \text{ and } \phi_j \in \mathrm{knn}(\phi_i; k) \text{ and } \psi_j \in \mathrm{knn}(\psi_i; k)], \quad (54)$$

so this term only considers k nearest neighbors at each i. In this paper, we randomly sample 10,000 images in the validation set in ImageNet [8] and report CKNNA with $k = 10$ based on observation in [27] that smaller k shows better alignment.

C.2 Description of pretrained visual encoders

- **MoCov3** [7] presents an empirical study of training MoCo [20, 5] on vision transformers and how they can be scaled up.
- **CLIP** [53] proposes a contrastive learning scheme on large-scale image-text pairs.
- **DINOv2** [51] proposes a self-supervised learning method that combines pixel-level and patch-level discriminative objectives by leveraging advanced self-supervised techniques and a large pretraining dataset.

D Hyperparameter and More Implementation Details

D.1 Hyperparameter Tuning

We adopt a bisection style search to determine the key hyperparameters for ERW, specifically the ERW *Depth* (i.e., which early layers to initialize), the *Projection Depth*, and the initial value of λ in Eq. (14). To keep the search computationally manageable, we do the following for each candidate hyperparameter setting:

(a) We run a short warmup stage for 10k iterations, followed by 20k iterations of main diffusion training.

(b) To evaluate performance quickly, we reduce the sampling steps from the usual 250 to 50 and generate only 10k samples (instead of 50k) to compute a preliminary FID score.

This procedure substantially reduces the search cost while retaining sufficient fidelity to guide hyperparameter choices. In practice, around three to five such tests suffice to converge upon near optimal settings for ERW Depth, Projection Depth, and λ, enabling both efficient training and high-quality generation.

Further implementation details. We implement our model based on the original SiT implementation [48]. Throughout the experiments, we use the exact same structure as DiT [52] and SiT [48]. We use AdamW [31, 46] with constant learning rate of 1e-4, $(\beta_1, \beta_2) = (0.9, 0.999)$ without weight decay. To speed up training, we use mixed-precision (fp16) with gradient clipping at norm 1.0. We also precompute compressed latent vectors from raw pixels via stable diffusion VAE [54] and use these latent vectors. Because of this, we do not apply any data augmentation, but we find this does not lead to a big difference, as similarly observed in EDM2 [29]. We also use `stabilityai/sd-vae-ft-ema` decoder for decoding latent vectors to images. For MLP used for a projection, we use a three-layer MLP with SiLU activations [10]. We provide a detailed hyperparameter setup in Table 8.

Pretrained encoders. For MoCov3-B and -L models, we use the checkpoint in the implementation of RCG [40];[1] for other checkpoints, we use their official checkpoints released in their official implementations. To adjust a different number of patches between the diffusion transformer and the pretrained encoder, we interpolate positional embeddings of pretrained encoders.

[1] https://github.com/LTH14/rcg

Table 8: Hyperparameter setup.

	Figure 1,2,3	Table 3,4 (SiT-B)	Table 1,2,5 (SiT-XL)
Architecture			
Input dim.	32×32×4	32×32×4	32×32×4
Num. layers	28	12	24
Hidden dim.	1,152	768	1,152
Num. heads	16	12	16
ERW			
$\mathrm{sim}(\cdot,\cdot)$	NT-Xent	NT-Xent	NT-Xent
Encoder $f(\mathbf{x})$	DINOv2-B	DINOv2-B	DINOv2-B
Optimization			
Batch size	256	256	256
Optimizer	AdamW	AdamW	AdamW
lr	0.0001	0.0001	0.0001
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Interpolants			
α_t	$1-t$	$1-t$	$1-t$
σ_t	t	t	t
w_t	σ_t	σ_t	σ_t
Training objective	v-prediction	v-prediction	v-prediction
Sampler	Euler-Maruyama	Euler-Maruyama	Euler-Maruyama
Sampling steps	250	250	250
Guidance	-	-	-

Table 9: **Impact of Training Tricks in ERW**. Using the SD-VAE [54], ERW achieves an FID of 51.7 at 50K training steps on ImageNet class-conditional generation. This table illustrates how each training trick incrementally improves the FID, demonstrating that advanced design techniques enhance the original DiT performance.

Training Trick	**Training Step**	**FID-50k↓**
Representation Alignment Loss		
+ REPA [77]	50K	78.2
Architecture Improvements		
+ Rotary Pos Embed [67]	50K	73.6
Initialization		
+ ERW (Ours)	50K	51.7

Sampler. For sampling, we use the Euler-Maruyama sampler with the SDE with a diffusion coefficient $w_t = \sigma_t$. We use the last step of the SDE sampler as 0.04, and it gives a significant improvement, similar to the original SiT paper [48].

Training Tricks. We explore the influence of various training techniques on ERW's performance. Notably, we observe performance improvements when incorporating Rotary Positional Embeddings [67].

E Evaluation Details

We strictly follow the setup and use the same reference batches of ADM [9] for evaluation, following their official implementation.[2] We use 8×NVIDIA H800 80GB GPUs for evaluation and enable tf32 precision for faster generation, and we find the performance difference is negligible compared to the original fp32 precision.

In what follows, we explain the main concept of metrics that we used for the evaluation.

- **FID** [23] measures the feature distance between the distributions of real and generated images. It uses the Inception-v3 network [68] and computes distance based on an assumption that both feature distributions are multivariate gaussian distributions.
- **sFID** [49] proposes to compute FID with intermediate spatial features of the Inception-v3 network to capture the generated images' spatial distribution.

[2] https://github.com/openai/guided-diffusion/tree/main/evaluations

- **IS** [56] also uses the Inception-v3 network but uses logits for evaluation of the metric. Specifically, it measures a KL divergence between the original label distribution and the distribution of logits after the softmax normalization.
- **Precision and recall** [35] are based on their classic definitions: the fraction of realistic images and the fraction of training data manifold covered by generated data.

F Baselines

In what follows, we explain the main idea of baseline methods that we used for the evaluation.

- **ADM** [9] improves U-Net-based architectures for diffusion models and proposes classifier-guided sampling to balance the quality and diversity tradeoff.
- **VDM++** [30] proposes a simple adaptive noise schedule for diffusion models to improve training efficiency.
- **Simple diffusion** [26] proposes a diffusion model for high-resolution image generation by exploring various techniques to simplify a noise schedule and architectures.
- **CDM** [25] introduces cascaded diffusion models: similar to Progressive GAN [28], it trains multiple diffusion models starting from the lowest resolution and applying one or more super-resolution diffusion models for generating high-fidelity images.
- **LDM** [54] proposes latent diffusion models by modeling image distribution in a compressed latent space to improve the training efficiency without sacrificing the generation performance.
- **U-ViT** [1] proposes a ViT-based latent diffusion model that incorporates U-Net-like long skip connections.
- **DiffiT** [18] proposes a time-dependent multi-head self-attention mechanism for enhancing the efficiency of transformer-based image diffusion models.
- **MDTv2** [15] proposes an asymmetric encoder-decoder scheme for efficient training of a diffusion-based transformer. They also apply U-Net-like long shortcuts in the encoder and dense input shortcuts in the decoder.
- **MaskDiT** [81] proposes an asymmetric encoder-decoder scheme for efficient training of diffusion transformers, where they train the model with an auxiliary mask reconstruction task similar to MAE [19].
- **SD-DiT** [85] extends MaskDiT architecture but incorporates self-supervised discrimination objective using a momentum encoder.
- **DiT** [52] proposes a pure transformer backbone for training diffusion models based on proposing AdaIN-zero modules.
- **SiT** [48] extensively analyzes how DiT training can be efficient by moving from discrete diffusion to continuous flow-based modeling.
- **REPA** [77] proposes a representation alignment method for diffusion models by aligning the representation of the diffusion model with a pretrained encoder.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main claims on representation preconditioning, clean latent warmup, and matched-budget convergence, which are supported by the experiments in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix A, covering the controlled SiT/REPA comparison, the acceleration focus, and the limited scope beyond ImageNet and MS-COCO.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Appendix B states the regularity and moment conditions, proves the finite-candidate NT-Xent likelihood identity and oracle MMSE comparison, and gives a conditional Lipschitz approximation bound. The text distinguishes these statements from empirical optimization claims.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The setup sections and appendix describe the backbones, datasets, evaluation metrics, budget accounting, and key hyperparameters sufficient to reproduce the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code and model weights are not released at submission time, although the experiments use publicly available datasets and pretrained components.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The training and evaluation setup, including backbones, samplers, metrics, and hyperparameter choices, is described in the main text and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars or multi-seed confidence intervals are not reported, as the main results are presented as fixed-protocol comparisons at model scale.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The appendix states the training and evaluation hardware, precision, and runtime information needed to reproduce the reported experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This work is purely methodological research on diffusion transformer training and representation preconditioning.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is foundational research on training optimization for diffusion transformers. While improved image generation could in principle facilitate misuse such as deepfakes, the contribution is a training methodology rather than a deployed system.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No model weights or datasets are publicly released at submission time, so safeguards for public release are not applicable at this stage.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: All external datasets and pretrained models used are properly cited in the references; we do not redistribute any third party assets and defer detailed license enumeration to the code release.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets (code, model checkpoints, or datasets) are publicly released with this submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve any research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as any component of the core method; any use was limited to writing assistance only.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.