UNLOCKING THE POTENTIAL OF CONTRASTIVE REGULARIZATION FOR DIFFUSION TRANSFORMER TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Representation regularization has emerged as an effective way to improve diffusion transformer training. Yet one of the most canonical regularizers–InfoNCE-style contrastive learning–remains largely underexplored in this setting. We naturally ask a question: can standard contrastive learning serve as an effective regularizer for diffusion transformers? Surprisingly, we find that naive InfoNCE regularization brings only marginal gains. Our analysis reveals that the objective is easily dominated by a shortcut: near-orthogonal noise components in diffusion inputs provide a bypass for enlarging negative-pair distances while weakening positive-pair alignment, especially at high noise levels. To unlock the benefit of contrastive regularization, we propose Clean Representation-ANchored Contrastive (CRAN) regularization. Specifically, CRAN anchors contrastive dispersion with clean-sample representation distances, transferring clean relational semantic structure to noisy diffusion representations to suppress the noise shortcut. Extensive experiments on ImageNet show that CRAN consistently improves diffusion transformer training across model scales and training budgets, outperforming internal feature-alignment and dispersive regularization baselines while matching external DINO-guided REPA. These results re-establish InfoNCE-based contrastive learning as an effective regularizer for diffusion training, once its noise-driven over-dispersion is properly constrained.

1 INTRODUCTION

Transformer-based diffusion and flow (Ho et al., 2020; Song et al., 2020; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023; Lipman et al., 2023) models now dominate generation tasks, from image synthesis (Podell et al., 2023; Saharia et al., 2022; Esser et al., 2024) to video generation (Polyak et al., 2024; Brooks et al., 2024). Recent studies have substantially improved diffusion transformer training through *representation-level regularization*, including MAE-style auxiliary objectives, external feature alignment, internal self-representation alignment, and feature-dispersion regularization (Zheng et al., 2023; Zhu et al., 2024; Krause et al., 2025; Yu et al., 2024; Jiang et al., 2026; Haghighi et al., 2025; Wang & He, 2025). However, existing methods rely on specialized alignment or dispersion objectives, while the canonical contrastive learning objective, *i.e.*, InfoNCE (He et al., 2020; Chen et al., 2020), remains largely underexplored in diffusion transformer training. This is surprising because InfoNCE has been one of the most successful objectives for representation learning in discriminative vision tasks. We therefore ask a natural question:

Can standard InfoNCE contrastive learning serve as an effective representation regularizer for diffusion transformers?

Method	Regularization	FID (↓)
Baseline	None	33.0
SRA	Positive only	28.7
InfoNCE	Positive + Negative	28.5
CRAN (ours)	Anchored InfoNCE	**26.3**

Table 1: FID-50K of SiT-B/2 trained on ImageNet 256×256 for 400K iterations. Standard InfoNCE regularization brings only marginal gains over the positive-only representation regularizer SRA (Jiang et al., 2026).

Evidence from prior work and our empirical study **suggests a negative answer**. Dispersive Loss (Wang & He, 2025) (see their Table 2) shows that standard contrastive losses can degrade

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

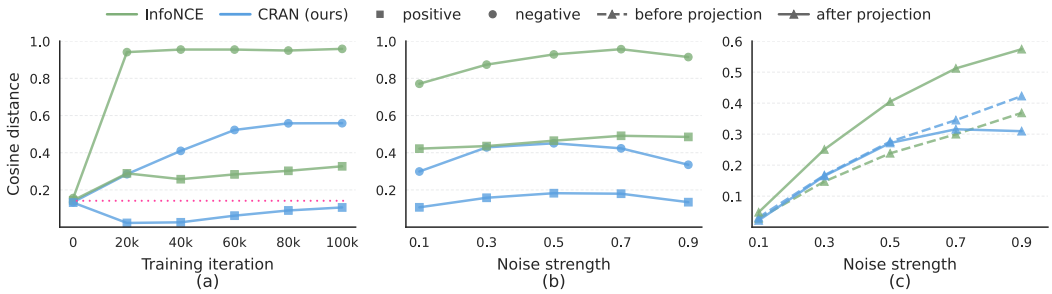


Figure 1: **Diagnostic studies of the noise shortcut.** (a) Standard InfoNCE rapidly over-disperses negative pairs and weakens positive-pair alignment, while CRAN alleviates both effects. (b) The over-dispersion and weakened alignment of InfoNCE become more severe at higher noise levels, whereas CRAN remains stable. (c) For a fixed clean image, the contrastive projection head amplifies noise-induced distances, revealing a noise shortcut; CRAN suppresses such amplification.

diffusion training with independently noised positive views, and require restricted noise gaps to obtain only limited gains. Consistent with prior evidence, we further test whether InfoNCE can become effective under a stronger self-representation alignment setup. We adopt the SRA framework (Jiang et al., 2026), where a middle SiT representation is aligned to a deeper and less noisy self-representation, and extend this positive-only regularizer into a full InfoNCE objective by adding negative-pair discrimination. However, as in Table 1, InfoNCE brings almost no gain over SRA, reducing FID only from 28.7 to 28.5.

To understand this issue, we argue that vanilla InfoNCE suffers from a **noise shortcut**. Diffusion inputs contain high-dimensional Gaussian noise, whose random components are known to be nearly orthogonal with high probability (Cai et al., 2013). Therefore, instead of learning semantic differences between images, the contrastive objective may simply exploit the naturally near-orthogonal noise components to enlarge negative-pair distances. This provides an easy way to satisfy the repulsive regularization, but can lead to over-dispersion and weaken positive-pair alignment[1].

To verify this argument, we conduct three diagnostic studies under standard InfoNCE regularization. (1) We track the average cosine distance ($1 - \cos(\mathbf{a}, \mathbf{b})$) of positive-pairs and negative-pairs during training. As shown in Figure 1 (a), negative-pairs are rapidly pushed toward orthogonality, which we refer to as **over-dispersion**. However, positive-pair distances are not properly minimized; instead, they first increase and *remain larger than their initial value*, indicating that positive-pair alignment is **weakened** rather than strengthened. (2) Figure 1 (b) shows that both positive- and negative-pair distances grow with noise strength, especially the negative-pair ones, suggesting that **stronger noise amplifies shortcut**-driven over-dispersion and weakened alignment. (3) We further consider a special case of positive-pairs: fixing the clean image while varying only the sampled noise, and measuring feature distances before and after the contrastive projection head. Figure 1 (c) shows that for the same image under different noise realizations, the contrastive projection head **amplifies noise-induced feature distances**, revealing that it exploits noise variation as a shortcut even for positive pairs.

Motivated by the analyses above, we propose Clean Representation-ANchored contrastive (CRAN) regularization. The key idea is to prevent InfoNCE from over-dispersing noisy representations beyond the discriminative structure present in clean representations. Specifically, CRAN anchors negative-pair dispersion in the contrastive objective using pairwise distances between clean-sample representations. In this way, clean representation geometry guides noisy diffusion representations, clipping unnecessary negative repulsion once the clean anchor is reached and thereby shifting optimization toward positive-pair alignment. As shown in Figure 1, CRAN mitigates the key failure modes of vanilla InfoNCE: it alleviates negative-pair over-dispersion and improves positive-pair alignment, reduces the sensitivity of representation distances to noise strength, and suppresses projection-head amplification of noise-induced distances. Consistently, Table 1 shows that CRAN reduces FID from 28.5 to 26.3 over vanilla InfoNCE, demonstrating that CRAN effectively blocks the noise shortcut and unlocks the potential of InfoNCE for diffusion transformer training.

Extensive experiments in Section 4 verify the effectiveness of CRAN, which achieves significant improvements over feature-alignment methods using internal features. Notably, using only internal

[1]Simply lowering the InfoNCE temperature does not remove this shortcut: Table 7 shows that temperature tuning still fails to recover the expected benefit of InfoNCE.

Table 2: **Unified view of representation regularization methods for diffusion training.** *P*: positive; *N*: negative; $\mathbf{x}_0^i/\mathbf{x}_t^i/\mathbf{x}_{t-\Delta t}^i$: clean/noisy/less-noisy query image; $\mathbf{z}_t^i = g_{\psi}(f_{\boldsymbol{\theta}}^k(\mathbf{x}_t^i))$: k-th block feature of noisy query projected by head g_{ψ}. sg: stop-gradient; $d(\cdot,\cdot)$: distance function; $f_{\bar{\boldsymbol{\theta}}}$: EMA model; $f^{k'}$ with $k' > k$: deeper-layer block; h: auxiliary reconstruction head. To ease the notation, we omit the temperature τ in Dispersive Loss (Wang & He, 2025), InfoNCE, and CRAN. † indicates the use of external knowledge.

Type	Method	Target / Pair Construction	Regularization Objective
P.-only	REPA† (Yu et al., 2024)	External pretrained representation $\mathbf{y}^i = E_{\mathsf{DINO}}(\mathbf{x}_0^i)$	$\mathbb{E}_i\left[d(\mathbf{z}_t^i, \mathrm{sg}(\mathbf{y}^i))\right]$
	LayerSync (Haghighi et al., 2025)	Detached deeper-layer representation $\mathbf{y}^i = f_{\boldsymbol{\theta}}^{k'}(\mathbf{x}_t^i)$	$\mathbb{E}_i\left[d(\mathbf{z}_t^i, \mathrm{sg}(\mathbf{y}^i))\right]$
	SRA (Jiang et al., 2026)	Deeper and less noisy EMA representation $\mathbf{y}^i = f_{\bar{\boldsymbol{\theta}}}^{k'}(\mathbf{x}_{t-\Delta t}^i)$	$\mathbb{E}_i\left[\lVert\mathbf{z}_t^i - \mathrm{sg}(\mathbf{y}^i)\rVert_2^2\right]$
	SRA2 (Wang et al., 2026)	Reconstructing VAE latent of clean image $\mathbf{y}^i = \mathbf{x}_0^i$	$\mathbb{E}_i\left[\lVert h(\mathbf{z}_t^i) - \mathrm{sg}(\mathbf{y}^i)\rVert_2^2\right]$
N.-only	Disp. Loss (Wang & He, 2025)	In-batch negatives $\{\mathbf{y}^j = \mathbf{z}_t^j\}_{j\neq i}$	$\log \mathbb{E}_{i,j}\left[\exp\left(-\lVert\mathbf{z}_t^i - \mathbf{y}^j\rVert_2^2\right)\right]$
P. + N.	InfoNCE	Positive $\mathbf{y}^i$ and in-batch negatives $\{\mathbf{y}^j\}_{j\neq i}$	$-\mathbb{E}_i\left[\log \frac{\exp(-d(\mathbf{z}_t^i,\mathbf{y}^i))}{\sum_j \exp(-d(\mathbf{z}_t^i,\mathbf{y}^j))}\right]$
	CRAN (ours)	InfoNCE + clean-representation anchor $m = \mathbb{E}_{i,j}[d(\mathbf{z}_0^i, \mathbf{z}_0^j)]$	$-\mathbb{E}_i\left[\log \frac{\exp(-d(\mathbf{z}_t^i,\mathbf{y}^i))}{\sum_j \exp(-\min(d(\mathbf{z}_t^i,\mathbf{y}^j), m))}\right]$

feature contrastive learning, CRAN matches REPA Yu et al. (2024), which relies on an external strong DINOv2 encoder Oquab et al. (2024), under the same training budget.

2 Preliminary

2.1 Flow and diffusion models

We use the stochastic interpolant formulation (Albergo et al., 2025) as a unified perspective for flow and diffusion models. Let $\mathbf{x}_0 \sim p_{\mathsf{data}}$ denote an image latent and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I)$ denote Gaussian noise. The interpolated state at timestep t is defined as $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$, where $\alpha_t = \alpha(t)$ and $\sigma_t = \sigma(t)$ are time-dependent coefficients, with $\alpha(t)$ decreasing and $\sigma(t)$ increasing over time. The probability-flow ODE is governed by the velocity field: $\dot{\mathbf{x}}_t = \mathbf{v}(\mathbf{x}_t, t)$. In practice, $\mathbf{v}_{\boldsymbol{\theta}}$ is trained to regress the sample-wise velocity target $\dot{\alpha}_t \mathbf{x}_0 + \dot{\sigma}_t \boldsymbol{\epsilon}$:

$$\mathcal{L}_{\mathsf{gen}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon},t}\left[\lVert\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - (\dot{\alpha}_t \mathbf{x}_0 + \dot{\sigma}_t \boldsymbol{\epsilon})\rVert_2^2\right]. \tag{1}$$

Here, $\dot{\alpha}_t$ and $\dot{\sigma}_t$ denote time derivatives. Different choices of α_t and σ_t recover common flow and diffusion parameterizations. For example, flow model uses $\alpha_t = 1 - t$ and $\sigma_t = t$, giving the target $\boldsymbol{\epsilon} - \mathbf{x}_0$; In VE diffusion (Song et al., 2020), $\alpha_t = 1$ and $\sigma_t = \sigma(t)$, yielding a pure noise-direction target $\dot{\sigma}_t \boldsymbol{\epsilon}$.

2.2 Existing representation regularizers for diffusion training

Recent studies show that representation regularization is effective for diffusion training. A unified view of diffusion representation regularization is summarized in Table 2. Based on how representation pairs are constructed and optimized, we categorize existing methods into three families.

Positive-only regularization. Positive-only regularizers improve diffusion training by aligning each noisy intermediate representation with a stronger target feature. Representative methods differ mainly in how this target feature is constructed:

- REPA (Yu et al., 2024) minimizes the cosine distance between intermediate diffusion features and representations from an external pretrained vision encoder, *e.g.*, DINOv2 (Oquab et al., 2024).
- LayerSync (Haghighi et al., 2025) directly minimizes the cosine distance between intermediate diffusion representations and detached deeper representations within the same diffusion transformer.
- SRA (Jiang et al., 2026) also aligns intermediate representations with detached deeper representations, but constructs the target from an EMA teacher network and a less noisy version of the current latent, using an ℓ_2 distance objective.

- SRA2 (Wang et al., 2026) uses intermediate diffusion representations to predict the VAE latent of the clean image.

Negative-only regularization. Negative-only regularizer disperses intermediate diffusion representations across samples without positive pairs. Dispersive Loss (Wang & He, 2025) instantiates this idea by minimizing a log-sum-exp objective over negative pairwise ℓ_2^2 distances.

Positive-negative contrastive regularization. Contrastive learning is one of the most successful objectives for visual representation learning (Chen et al., 2020; He et al., 2020; Radford et al., 2021). It learns discriminative features by aligning positive pairs while repelling negative pairs. Let $\mathbf{z}_i$ denote the query representation, $\mathbf{y}_i$ its positive target, and $\{\mathbf{y}_j\}_{j\neq i}$ the in-batch negatives. Given a distance function $d(\cdot,\cdot)$, the InfoNCE-based contrastive objective is

$$\mathcal{L}_{\mathsf{CL}}(\boldsymbol{\theta},\boldsymbol{\psi}) = -\mathbb{E}_i\left[\log \frac{\exp\big(-d(\mathbf{z}^i,\mathbf{y}^i)/\tau\big)}{\sum_{j=1}^{B}\exp(-d(\mathbf{z}^i,\mathbf{y}^j)/\tau)}\right]. \tag{2}$$

However, directly applying this full objective to diffusion training remains challenging. Dispersive Loss (Wang & He, 2025) shows that $\mathcal{L}_{\mathsf{CL}}$ can degrade generation quality, and only yields limited gains after carefully restricting the noise gap between positive views. We revisit this problem and find that vanilla InfoNCE is dominated by noise-driven dispersion rather than semantic discrimination. Moreover, this shortcut cannot be resolved by simply tuning the temperature τ: as shown in Table 7, using a smaller temperature does not recover the expected benefit of InfoNCE.

3 Methodology

Given a noisy query latent $\mathbf{x}_t^i$, let $\mathbf{z}_t^i = g_{\boldsymbol{\psi}}(f_{\boldsymbol{\theta}}^k(\mathbf{x}_t^i))$ denote the projected intermediate diffusion representation, where $f_{\boldsymbol{\theta}}^k(\cdot)$ produces the k-th block features and $g_{\boldsymbol{\psi}}(\cdot)$ is an MLP projection head parameterized by $\boldsymbol{\psi}$. Let $\mathbf{y}^i$ denotes the detached positive or negative representations, *e.g.*, a deeper and less noisy self-representation as in SRA (Jiang et al., 2026):

$$\mathbf{y}^i = f_{\bar{\boldsymbol{\theta}}}^{k'}(\mathbf{x}_{t-\Delta t}^i), \text{ where } k' > k \text{ and } 0 < t-\Delta t < t. \tag{3}$$

Here, $\bar{\boldsymbol{\theta}}$ denotes the EMA parameters. For convenience, we define

$$d_t^{ij} = d(\mathbf{z}_t^i, \mathbf{y}^j),$$

where $d(\cdot,\cdot)$ denotes a representation distance, such as squared ℓ_2 distance or cosine distance.

3.1 Fixed-anchor contrastive regularization

Evidence from prior work (Wang & He, 2025) and our empirical analysis in Table 1 suggest that standard InfoNCE in Eq. (2) is not directly effective for diffusion training. We attribute this failure to noise-shortcut-induced negative-pair over-dispersion. Since diffusion representations contain high-dimensional noise components that are naturally near-orthogonal, contrastive learning can satisfy its repulsive objective through noise-induced separation rather than semantic discrimination, as shown in Figure 1.

A direct remedy is therefore to stop the contrastive objective from further pushing negative pairs that are already sufficiently separated. To this end, we introduce Anchored Contrastive Learning (ACL) with a fixed distance anchor m_{fix} and clip the negative-pair distances in the InfoNCE denominator:

$$\mathcal{L}_{\mathsf{ACL}}(\boldsymbol{\theta},\boldsymbol{\psi},m_{\mathsf{fix}}) = -\mathbb{E}_i\left[\log \frac{\exp(-d_t^{ii}/\tau)}{\sum_j \exp\Big(-\min(d_t^{ij},m_{\mathsf{fix}})/\tau\Big)}\right]. \tag{4}$$

Here, $\min(d_t^{ij}, m_{\mathsf{fix}})$ limits the maximum separation enforced on negative pairs. Thus, the anchor prevents InfoNCE from blindly pushing negative pairs toward orthogonality. Once a negative pair reaches the distance bound, it no longer receives additional repulsive gradients, so the loss can be further reduced mainly by improving positive-pair alignment. This both alleviates over-dispersion and strengthens the alignment of matched representations.

Notably, our results in Table 7 show that a fixed anchor already validates this intuition: setting $m_{\mathsf{fix}} = 0.5$ ($\cos 60°$) improves FID from 28.5 to 27.0 on SiT-B/2.

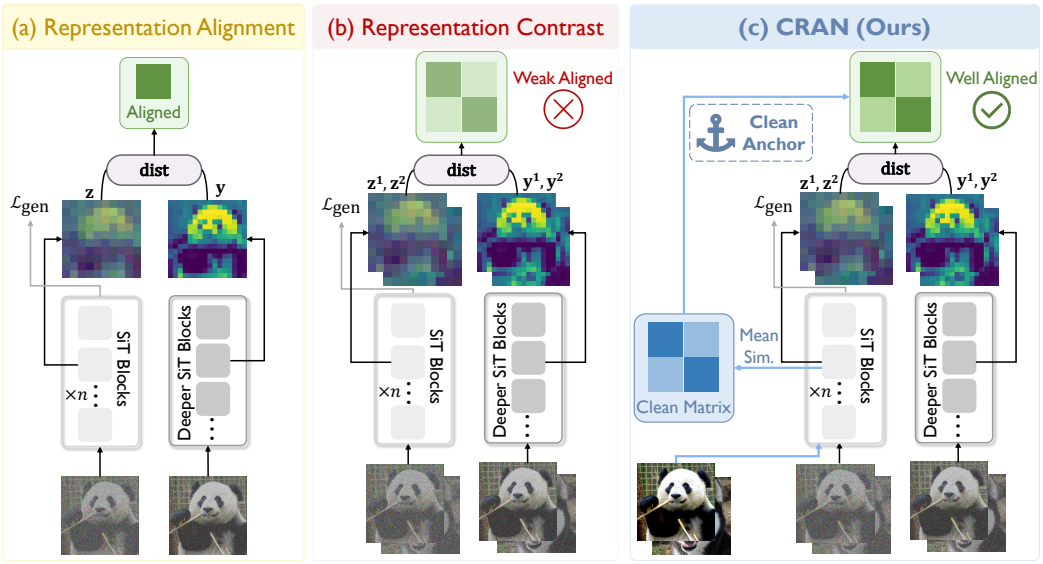


Figure 2: **(a)** Prior diffusion representation alignment methods, *e.g.*, SRA (Jiang et al., 2026). **(b)** Standard contrastive learning leads to severe over-dispersion and weakened alignment, resulting in limited performance gains. **(c)** CRAN uses clean-representation distances to anchor the maximum contrastive separation, alleviating over-dispersion and improving positive-pair alignment, which yields substantially stronger performance than representation alignment.

3.2 Clean Representation-Anchored Contrastive Regularization

Although the fixed-anchor variant validates the importance of limiting negative-pair dispersion, it remains suboptimal for two reasons: (1) it requires manually tuning an extra hyperparameter; (2) its fixed threshold cannot adapt to the evolving inter-representation geometry during training. To remove this hand-crafted design, we turn to clean-sample representations. Unlike noisy diffusion representations, clean representations are not dominated by injected Gaussian noise and thus provide a more reliable estimate of semantic representation geometry.

Motivated by this, we propose Clean Representation-ANchored contrastive (CRAN) regularization, as in Figure 2. Instead of using a fixed anchor, CRAN uses the average negative pairwise distance among clean-sample representations as an adaptive anchor (m_{cran}) for noisy contrastive learning.

$$\mathcal{L}_{\text{CRAN}}(\boldsymbol{\theta}, \boldsymbol{\psi}) = \mathcal{L}_{\text{ACL}}(\boldsymbol{\theta}, \boldsymbol{\psi}, m_{\text{cran}}), \text{ where } m_{\text{cran}} = \mathbb{E}_{i \neq j}\left[d(\mathbf{z}_0^i, \mathbf{z}_0^j)\right]. \tag{5}$$

Here, $\mathbf{z}_0^i = g_\psi(f_{\boldsymbol{\theta}}^k(\mathbf{x}_0^i))$ denotes the projected representation of the clean input, and m_{cran} is computed as the empirical mean within the current batch. This allows noisy contrastive learning to adaptively inherit the clean inter-sample geometry. The overall objective is given by:

$$\mathcal{L}_{\text{total}}(\boldsymbol{\theta}, \boldsymbol{\psi}) = \mathcal{L}_{\text{gen}}(\boldsymbol{\theta}) + \lambda \mathcal{L}_{\text{CRAN}}(\boldsymbol{\theta}, \boldsymbol{\psi}), \tag{6}$$

where $\mathcal{L}_{\text{gen}}$ is the flow or diffusion loss defined in Eq. (1) and λ controls the strength of regularization.

3.3 Clean Inter-Sample Geometry Estimation

During training, each batch naturally contains samples at different noise levels. We select samples whose noise levels are below a threshold $t_c = 0.1$ and use the average pairwise angle between their representations to determine the adaptive margin m_{cran}. In this way, m_{cran} provides an online estimate of the clean inter-sample geometry without requiring an additional clean-sample branch.

4 Experiments

In this section, we conduct extensive experiments to answer the following research questions (RQs):

RQ1: Can CRAN significantly improve the performance of diffusion transformers and surpass current state-of-the-art representation regularization methods?

Table 3: FID comparison across model scales and training budgets without CFG.

Model	#Params	Iter.	FID↓
SiT-B/2	130M	400k	33.0
+ SRA	130M	400k	29.1
+ SRA & Disp.	130M	400k	28.6
+ DiverseDiT	>130M	400k	28.1
+ SRA2	130M	400k	28.9
+ CRAN	130M	400k	**26.3**
SiT-L/2	458M	400k	18.8
+ SRA	458M	400k	14.4
+ DiverseDiT	>458M	400k	16.1
+ SRA2	458M	400k	14.3
+ CRAN	458M	400k	**13.3**
SiT-XL/2	675M	400k	17.2
+ DiverseDiT	713M	400k	12.4
+ SRA 2	675M	400k	11.7
+ CRAN	675M	400k	**10.3**
SiT-XL/2	675M	7M	8.3
SiT-XL/2	675M	1M	12.2
+ SRA2	675M	1M	8.2
+ CRAN	675M	1M	**7.5**
SiT-XL/2	675M	2M	10.1
+ SRA2	675M	2M	7.0
+ CRAN	675M	2M	**6.6**
SiT-XL/2	675M	4M	9.0
+ Disp.	675M	4M	7.7
+ SRA	675M	4M	7.4
+ SRA2	675M	4M	6.6
+ CRAN	675M	4M	**6.2**

Table 4: Comparison with SoTA methods under CFG.

Method	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Pixel diffusion						
ADM-U (2021)	400	3.94	6.14	186.7	0.82	0.52
VDM++ (2023)	560	2.40	-	225.3	-	-
Simple diffusion (2023)	800	2.77	-	211.8	-	-
CDM (2022)	2160	4.88	-	158.7	-	-
Latent diffusion, U-Net						
LDM-4 (2022)	200	3.60	-	247.7	0.87	0.48
Latent diffusion, Transformer + U-Net hybrid						
U-ViT-H/2 (2023)	240	2.29	5.68	263.9	0.82	0.57
DiffiT (2024)	-	1.73	-	276.5	0.80	0.62
MDTv2-XL/2 (2023)	1080	1.58	4.52	314.7	0.79	0.65
Latent diffusion, Transformer						
MaskDiT (2023)	1600	2.28	5.67	276.6	0.80	0.61
SD-DiT (2024)	480	3.23	-	-	-	-
DiT-XL/2 (2023)	1400	2.27	4.60	278.2	0.83	0.57
+ TRADE (2025)	740	1.69	4.70	305.7	0.80	0.65
SiT-XL/2 (2024)	1400	2.06	4.50	270.3	0.82	0.59
With external representations						
+ REPA (2024)	800	1.42	4.70	305.7	0.80	0.65
+ REG (2025)	800	1.36	4.25	299.4	0.77	0.66
Without external representations						
+ LayerSync* (2025)	800	1.56	4.82	287.0	0.80	0.63
+ SRA (2026)	800	1.58	4.65	305.7	0.80	0.65
+ SRA2 (2026)	800	1.52	4.63	316.2	0.82	0.62
+ CRAN	800	**1.45**	4.68	306.6	0.81	0.63

*LayerSync includes an additional reverse-process variance head (2021); we remove it for fair comparison.

RQ2: How do CRAN-specific choices (d, λ) affect performance, and can temperature tuning τ in naive InfoNCE or fixed-anchor contrastive regularization match the performance of our CRAN?

RQ3: How does the adaptive anchor m_{cran} evolve in SiT+CRAN training?

RQ4: What representation properties are improved by CRAN?

Implementation details. We conduct experiments on Scalable Interpolant Transformers (SiT) (Ma et al., 2024), which provide a unified view of diffusion and flow models. Our evaluation covers three model scales: SiT-B/2, SiT-L/2, and SiT-XL/2. For a fair comparison, we follow the REPA (Yu et al., 2024) training configuration. All experiments are carried out on ImageNet (Deng et al., 2009), with images center-cropped and resized to 256×256. They are then encoded by the Stable Diffusion VAE with a compression factor of 8, resulting in latents with shape $\mathbb{R}^{32\times32\times4}$. Training is performed with a batch size of 256 using AdamW, a constant learning rate of 1×10^{-4}, and no weight decay.

Inference and evaluation. For inference, we follow SiT and REPA and use the Euler-Maruyama sampler for the SDE with $w_t = \sigma_t$, with the default number of function evaluations (NFE) set to 250. For evaluation, we adopt the ADM TensorFlow evaluation protocol (Dhariwal & Nichol, 2021), generating 50K samples to ensure statistical reliability and comparing them against the corresponding reference statistics. We report Fréchet Inception Distance (FID) (Heusel et al., 2017), structural FID (sFID) (Nash et al., 2021), Inception Score (IS) (Salimans et al., 2016), precision, and recall (Kynkäänniemi et al., 2019) to assess generation quality from complementary perspectives. Additional implementation details are provided in the Appendix F.

4.1 The effectiveness of CRAN (RQ1)

We first compare the improvements brought by CRAN on SiT across different model scales with those of other representation alignment methods. We use FID as the primary comparison metric and do not use classifier-free guidance. As shown in Table 3, CRAN substantially improves SiT across different model sizes and training budgets. Compared with existing representation alignment methods, including positive regularizers such as SRA (Jiang et al., 2026) and SRA2 (Wang et al., 2026), as well as negative regularizers such as Dispersive Loss (Wang & He, 2025) and DiverseDiT (Yang et al., 2026), CRAN achieves the best performance across different training stages and model sizes. It



Figure 3: Generated images from SiT-XL/2 + CRAN. More visualizations are provided in Appendix H.

Table 5: Ablation studies on the design choices of CRAN. We study the distance metric, the setting of the maximum separation, and the weight of the CRAN objective. The best results are in blue.

Method	dis. metric	Dis. Anchor	λ	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
SiT-B/2	-	-	-	33.0	6.46	43.7	0.53	0.63
+ CRAN	ℓ_2	-	1.0	31.9	6.53	44.8	0.55	0.63
+ CRAN	cos	-	1.0	28.3	5.92	50.9	0.57	0.63
+ CRAN	cos	cos 30°	1.0	27.9	5.99	51.3	0.57	0.64
+ CRAN	cos	cos 60°	1.0	27.0	6.06	53.9	0.58	0.64
+ CRAN	cos	cos 70°	1.0	28.6	5.99	50.5	0.57	0.64
+ CRAN	cos	m_{cran}	1.0	26.3	6.16	56.2	0.58	0.63
+ CRAN	cos	m_{cran}	0.5	28.6	6.08	50.6	0.57	0.63
+ CRAN	cos	m_{cran}	1.0	26.3	6.16	56.2	0.58	0.63
+ CRAN	cos	m_{cran}	1.5	26.8	6.12	54.3	0.58	0.63

is worth noting that simply combining SRA with Dispersive Loss yields only marginal improvement and remains notably behind CRAN (28.6 vs. 26.3). Specifically, CRAN achieves 7.5 non-cfg FID using only 1M iterations, which outperforms vanilla SiT-XL/2 trained for 7M iterations. The performance of CRANcontinues to improve with more training iteration, and finally achieves 6.2 FID, which is the best result among methods that do not use external pretrained visual encoders.

Finally, we use SiT-XL/2 as the baseline and compare CRAN with other state-of-the-art methods under classifier-free guidance. As shown in Table 4, CRAN substantially improves the performance of SiT and reaches an FID of 1.45 at 800 epochs, outperforming MaskDiT (Zheng et al., 2023) and SD-DiT (Zhu et al., 2024), which introduce more complex architectures. Compared with representation alignment methods built on the same SiT-XL/2 backbone, CRAN achieves the best FID among methods that rely on features from the diffusion transformer itself, surpassing SRA (Jiang et al., 2026), SRA2 (Wang et al., 2026) and LayerSync (Haghighi et al., 2025).

4.2 Ablation Studies (RQ2)

As shown in Table 5, we study the key design choices in CRAN, including the distance metric, the setting of the similarity anchor, and the weight of the CRAN regularization. We first evaluate representation contrastive regularization with different distance metrics. The results show that plain representation contrastive regularization can improve SiT, but the gains remain relatively marginal. Between the ℓ_2 distance and the cosine distance, we choose the cosine distance as our distance metric,

as it achieves better performance and is widely used in representation learning to constrain the angular relationships between features (He et al., 2020; Radford et al., 2021; Chen et al., 2020).

Next, we study how different similarity anchor settings affect performance. A fixed threshold can improve performance when properly tuned. However, using the similarity between clean features as the anchor value achieves better results. We attribute this to the fact that the clean representation similarity more faithfully reflects the relationships among images at the representation level, while also allowing the hinge value to adapt dynamically during training.

We then study the effect of the CRAN weight. The results show that too small a weight degrades the performance of CRAN. We therefore use $\lambda = 1.0$ as the default weight in our final setting.

Table 6 isolates the two components of the contrastive objective. Removing either the positive or negative component degrades performance. The positive term provides cross-layer and cross-noise alignment, while the anchored negative term preserves useful inter-sample discrimination; their combination yields the strongest overall result. We further compare temperature scaling, fixed anchors, and the adaptive clean anchor in Appendix B (Table 7).

Table 6: Component ablation on SiT-B/2

Method	Pos.	Neg.	FID↓
SiT-B/2	No	No	33.0
CRAN	No	Yes	31.9
CRAN	Yes	No	28.7
CRAN	Yes	Yes	**26.3**

4.3 Dynamics of clean representation anchor (RQ3)

We visualize the evolution of the clean representation anchor m_{cran} during training on SiT-B/2, SiT-L/2, and SiT-XL/2.

As shown in Fig. 4, m_{cran} follows a consistent pattern across model scales. It increases rapidly in the early training stage, providing stronger discriminative regularization when the diffusion representations are still underdeveloped. After reaching its peak, the anchor gradually decreases and eventually stabilizes as the model becomes more stable. Among the three models, SiT-B/2 maintains the largest clean representation anchor. This may be because its smaller capacity requires a more uniformly distributed representation space, whereas larger models can rely on weaker regularization after convergence.

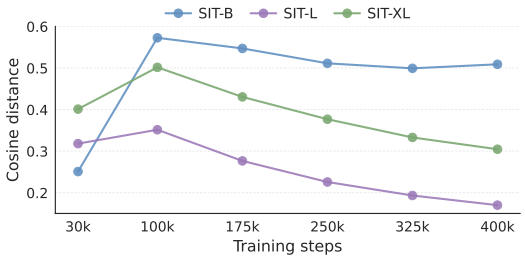


Figure 4: Evolution of the clean representation anchor during training.

4.4 Representation analysis (RQ4)

Spatial representation structure. To examine how CRAN affects diffusion representations, we follow the iREPA (Singh et al., 2025) evaluation paradigm, which emphasizes that spatial-structure metrics are more closely aligned with diffusion generative performance than linear probing accuracy. Specifically, we evaluate the learned representations using three metrics: Local vs. Distant Similarity (LDS) (Huang et al., 1997), which measures the contrast between local and distant patch similarities; Correlation Decay Slope (CDS), which quantifies how quickly patch similarity decreases with spatial distance; and RMS Spatial Contrast (RMSC), which measures the spatial variation of normalized patch features. These metrics allow us to quantitatively compare the representations learned by CRAN against those of competing methods. The detailed computation procedures for these metrics are provided in the Appendix G.

We compare our CRAN with a representative representation-alignment method, *i.e.*, SRA (Jiang et al., 2026) which aligns representation across layers and time steps. As shown in Fig. 5, CRANdemonstrates more significant enhancement in all three spatial-structure metrics at the regularized layer (layer 8). At deeper layers, CRAN brings more pronounced improvements.

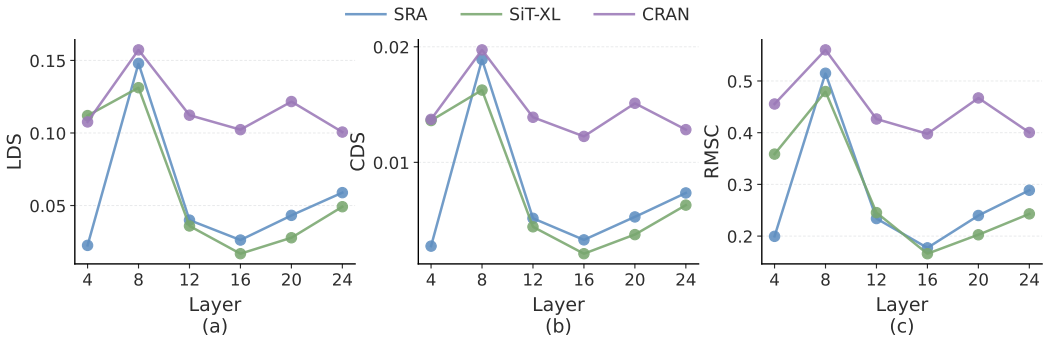


Figure 5: Spatial-structure metric analysis using SiT-XL/2. CRAN outperforms both SiT and SRA at the aligned layer, with even more pronounced improvements observed in deeper layers.

5 Related Work

Diffusion transformers for image generation. Diffusion (Ho et al., 2020; Song et al., 2020; Rombach et al., 2022; Podell et al., 2023) and flow-based models (Lipman et al., 2023; Liu et al., 2023; Esser et al., 2024) have become dominant paradigms for visual generation. With transformer backbones, diffusion models have scaled more effectively to larger architectures and stronger visual generation performance. DiT (Peebles & Xie, 2023) first demonstrated that vision transformers provide a strong backbone for diffusion modeling, while SiT (Ma et al., 2024) unified diffusion and flow-based training within a scalable stochastic-interpolant transformer framework. Despite these advances, diffusion transformer training remains costly, motivating representation-level regularization for more efficient and effective optimization.

Representation regularization for diffusion models. Representation regularization has shown promise in improving diffusion training. MaskDiT (Zheng et al., 2023) and SD-DiT (Zhu et al., 2024) adopt masked pretraining objectives, while MAEtok (Chen et al., 2025) combines reconstruction and alignment losses in tokenizer training. These approaches often require additional decoders or teacher branches (Jiang et al., 2026). REPA (Yu et al., 2024) and iREPA (Singh et al., 2025) instead align diffusion features with external pretrained representations. Without external encoders, SRA (Jiang et al., 2026) and LayerSync (Haghighi et al., 2025) use deeper internal features as alignment targets, DUPA (Peng et al., 2026) aligns low-frequency conditions across independently noised paths, and LSEP (Yun et al., 2025) promotes linear separability without alignment targets. Dispersive Loss (Wang & He, 2025) and DiverseDiT (Yang et al., 2026) explore negative-only feature dispersion. However, InfoNCE-based contrastive regularization, which jointly aligns positive pairs and disperses negative pairs, remains underexplored in diffusion transformer training despite its success in discriminative representation learning.

6 Conclusion and Limitation

In this work, we present CRAN, a successful attempt to introduce full InfoNCE-based contrastive learning regularization in diffusion transformers training. We find that vanilla contrastive learning fails because it tends to exploit the noise components in diffusion inputs to enlarge representation distances. As a result, it learns noise-induced dispersion rather than proper inter-sample relationships. To address this issue, CRAN anchors the maximum negative-pair distance in InfoNCE using pairwise distances among clean representations, thereby suppressing the noise-induced dispersion shortcut. As a result, CRAN unlocks the benefit of contrastive learning for diffusion transformer training and outperforms existing representation-alignment methods.

Limitation and possible remedy. CRAN requires estimating clean representation anchors during training, introducing a overhead from an additional near-clean forward pass. This cost is comparable to methods that require extra feature extraction, such as REPA (Yu et al., 2024), and does not affect inference time. A possible remedy for this overhead is to amortize anchor estimation m_{cran} with an EMA update when training samples are near the clean endpoint, which we leave for future work.

AI USE STATEMENT

We used large language models to assist with writing and formatting the manuscript.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce the main experimental results in Sec. 4 and Appendix F, including the model configurations, training settings, representation-alignment details, and evaluation protocols.

REFERENCES

Michael Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *ICLR 2023 Conference*, 2023.

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

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

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

Tony Cai, Jianqing Fan, and Tiefeng Jiang. Distributions of angles in random packing on spheres. *Journal of Machine Learning Research*, 14(21):1837–1864, 2013.

Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *Forty-second International Conference on Machine Learning*, 2025.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

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 *Forty-first international conference on machine learning*, 2024.

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.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5152–5161, 2022.

Yasaman Haghighi, Bastien van Delft, Mariam Hassan, and Alexandre Alahi. Layersync: Self-aligning intermediate layers. *arXiv preprint arXiv:2510.12581*, 2025.

Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. In *European Conference on Computer Vision*, pp. 37–55, 2024.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

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*, volume 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pp. 13213–13232, 2023.

Jing Huang, S. Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 762–768, 1997. doi: 10.1109/CVPR.1997.609412.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. Representation alignment for diffusion transformers without external components. In *The Fourteenth International Conference on Learning Representations*, 2026.

Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Felix Krause, Timy Phan, Ming Gui, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Tread: Token routing for efficient architecture-agnostic diffusion training. *arXiv preprint arXiv:2501.04765*, 2025.

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*, volume 32, 2019.

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

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

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

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*, pp. 23–40. Springer, 2024.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter Battaglia. Generating images with sparse representations. In *International Conference on Machine Learning*, pp. 7958–7968, 2021.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

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. *Transactions on Machine Learning Research Journal*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Changhao Peng, Yuqi Ye, Shuangjun Du, Wenxu Gao, and Wei Gao. Dual-path condition alignment for diffusion transformers. In *International Conference on Learning Representations*, 2026.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

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*, pp. 8748–8763. PmLR, 2021.

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

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. *Advances in neural information processing systems*, 35:36479–36494, 2022.

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*, volume 29, 2016.

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.

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.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model. In *International Conference on Learning Representations*, 2023.

Mengmeng Wang, Dengyang Jiang, Liuzhuozheng Li, Yucheng Lin, Guojiang Shen, Xiangjie Kong, Yong Liu, Guang Dai, and Jingdong Wang. Sra 2: Variational autoencoder self-representation alignment for efficient diffusion training. *arXiv preprint arXiv:2601.17830*, 2026.

Runqian Wang and Kaiming He. Diffuse and disperse: Image generation with representation regularization. *arXiv preprint arXiv:2506.09027*, 2025.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *arXiv preprint arXiv:2507.01467*, 2025.

Mengping Yang, Zhiyu Tan, Binglei Li, Xiaomeng Yang, Hesen Chen, and Hao Li. Diversedit: Towards diverse representation learning in diffusion transformers. *arXiv preprint arXiv:2603.04239*, 2026.

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.

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

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv preprint arXiv:2306.09305*, 2023.

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. *arXiv preprint arXiv:2403.17004*, 2024.

A SDE SAMPLER

For stochastic sampling, we additionally need the score function $\mathbf{s}(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. The reverse-time SDE can be written as

$$d\mathbf{x}_t = \mathbf{v}(\mathbf{x}_t, t)\, dt - \frac{1}{2} w_t \mathbf{s}(\mathbf{x}_t, t)\, dt + \sqrt{w_t}\, d\overline{\mathbf{W}}_t, \tag{7}$$

where w_t is the diffusion coefficient and $\overline{\mathbf{W}}_t$ is a reverse-time Wiener process (Song et al., 2020; Ma et al., 2024; Wu et al., 2025). Since ϵ is Gaussian, the score satisfies

$$\mathbf{s}(\mathbf{x}, t) = -\sigma_t^{-1} \mathbb{E}\left[\epsilon \mid \mathbf{x}_t = \mathbf{x}\right] = \sigma_t^{-1} \frac{\alpha_t \mathbf{v}(\mathbf{x}, t) - \dot{\alpha}_t \mathbf{x}}{\alpha_t \dot{\sigma}_t - \dot{\alpha}_t \sigma_t}. \tag{8}$$

Following the setup of SiT, we use a linear interpolant to construct the noise path, with $\alpha_t = 1 - t$ and $\sigma_t = t$, where $t \in [0, 1]$.

B ADDITIONAL ABLATIONS AND DIAGNOSTICS

Directly probing the noise shortcut. We freeze each SiT-B/2 backbone and train a linear probe to reconstruct the injected noise ϵ from intermediate representations. We report the normalized mean squared error $\text{NMSE} = \|\hat{\epsilon} - \epsilon\|_2^2 / \|\epsilon\|_2^2$; lower NMSE indicates that the input noise is more accurately decoded and therefore more strongly encoded in the representation.

NMSE of linear probes that reconstruct input noise from intermediate representations.

Noise level t	SiT	InfoNCE	CRAN
0.10	0.8950	0.8232	0.8789
0.40	0.3638	0.3150	0.3417
0.80	0.0288	0.0277	0.028

Noise is linearly decodable from diffusion representations at all three levels. Relative to SiT, InfoNCE consistently lowers NMSE, indicating greater use of the injected noise, whereas CRAN weakens this amplification. These results support the noise-shortcut hypothesis rather than claiming that noise is the only task-irrelevant cue that contrastive learning may exploit.

Weakening the negative term does not improve generation quality. Intuitively, another alternative for alleviating over-dispersion and weakened alignment is to directly adjust the weight of the negative-pair term.

As shown in Figure 6 (a), using a smaller τ alleviates over-dispersion and weakened alignment in the contrastive space. However, it does not lead to a significant performance improvement as shown in Table 7. Although using a smaller τ alleviates over-dispersion and weakened alignment, Figure 6(b) shows that, under different values of τ, the representation distance still increases as the noise level grows. In contrast, as shown in Figure 1(b), when using CRAN, the representation distance tends to remain stable, and even decreases with increasing noise when $t > 0.5$. In Figure 6(c), the representation distance induced by noise is still amplified by the contrastive projection head, and the degree of amplification further increases as the noise level grows. These observations suggest that reducing τ does not resolve the underlying representation issue behind over-dispersion and weakened alignment, which explains why simply decreasing τ brings only limited improvement.

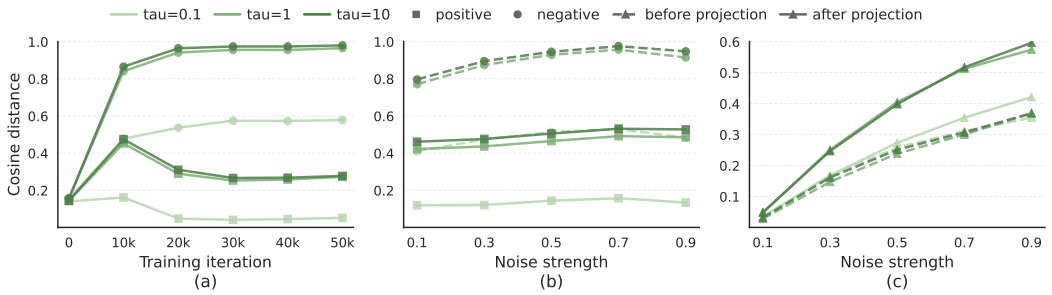


Figure 6: (a) Setting $\tau = 0.1$ alleviates over-dispersion and weakened alignment. (b) Setting $\tau = 0.1$ does not change the overall trend that over-dispersion and weakened alignment become more severe as the noise level increases. (c) For a fixed clean image, using a smaller τ cannot prevent the contrastive learning head from exploiting noise to increase the distance between representations.

Table 7: Union of the temperature and anchor ablations. Temperature scaling alone brings only marginal improvements, whereas limiting negative-pair separation with a fixed or adaptive anchor is substantially more effective.

Model	τ	Anchor	FID ↓
SiT-B/2	–	–	33.0
+ Pos. reg.	–	–	28.7
+ InfoNCE	10.0	–	28.3
+ InfoNCE	5.0	–	28.6
+ InfoNCE	1.0	–	28.5
+ InfoNCE	0.5	–	28.9
+ InfoNCE	0.1	–	28.3
+ InfoNCE	1.0	$\cos 60^\circ$	27.0
+ CRAN	1.0	m_{cran}	**26.3**

The effectiveness of clean image representation distances as the hinge. Next, we study how diffusion representation distances at different noise levels affect the ability of CRAN to correctly learn inter-sample relationships. Specifically, in addition to clean features, we use the inter-sample representation distances at $t = 0.5$ and $t = 0.9$ as the maximum distance in CRAN optimization. As shown in Table 8, when the noise weight in the samples becomes too large, the corresponding representation distances no longer provide reliable guidance for CRAN training.

Table 8: FID-50K of different feature regularization methods on SiT-B/2 trained on ImageNet 256×256 for 400K iterations.

Model	iter.	FID ↓	IS ↑
SiT-B/2	400k	33.0	43.7
SiT-B/2 + CRAN			
t=0.1	400k	**26.3**	**56.1**
t=0.5	400k	26.7	55.6
t=0.9	400k	28.2	50.8

Student–teacher layer pairing. We next vary the student and teacher layers on SiT-B/2. Changing the student layer or moving the teacher to a shallower layer clearly degrades performance, whereas moving the teacher from layer 8 to the deeper layer 10 has only a modest effect.

Table 9: Ablation of the student and teacher layers on SiT-B/2.

Student	Teacher	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
2	8	44.8	11.5	33.5	0.49	0.62
3	8	**42.9**	**11.4**	**36.0**	**0.50**	**0.62**
4	8	44.2	**11.4**	33.7	**0.50**	**0.62**
3	6	51.2	13.4	30.3	0.45	0.61
3	10	**42.9**	11.7	35.4	**0.50**	0.61

C GENERALIZATION BEYOND IMAGENET-256

C.1 LOW-CATEGORY AND STRUCTURALLY HOMOGENEOUS DATASETS

To test whether CRAN depends on the broad semantic diversity of ImageNet-1K, we evaluate it on FFHQ, a structurally homogeneous single-category face dataset. For a CIFAR-10-scale class-conditional setting, the native 32×32 resolution of CIFAR-10 is usually studied with U-Net diffusion models and is poorly matched to our focus on diffusion transformers. We therefore construct WebVision-10-50K from WebVision 1.0 (Li et al., 2017): we select ten ImageNet-style classes and sample 5,000 images per class, yielding the same number of classes and training images as CIFAR-10 while retaining a 256×256 setting.

Table 10: Results on datasets with substantially lower semantic diversity than ImageNet-1K. All methods use SiT-B/2 and 100K training steps.

Dataset	Method	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
FFHQ	SiT	6.73	7.55	4.4	0.71	0.47
FFHQ	REPA	6.49	7.63	4.5	0.71	0.48
FFHQ	CRAN	**5.59**	**7.02**	**4.6**	**0.71**	**0.49**
WebVision-10-50K	SiT	27.6	9.84	15.2	0.45	0.47
WebVision-10-50K	CRAN	**20.5**	**9.45**	**15.8**	**0.48**	**0.47**

On FFHQ, CRAN reduces FID by 16.9% relative to SiT and provides a larger gain than out-of-domain DINO-based alignment. On WebVision-10-50K, it reduces FID by 25.7% and improves all reported metrics. These results show that the benefit of CRAN does not require a large number of semantic categories.

C.2 HIGHER-RESOLUTION GENERATION

We train SiT-B/2 from scratch on ImageNet 512×512. Relative to the 256×256 setting, the latent patch sequence grows from $16 \times 16 = 256$ to $32 \times 32 = 1{,}024$ tokens.

Table 11: ImageNet 512×512 results with SiT-B/2.

Method	Steps	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
SiT	200K	56.7	8.11	26.8	0.53	**0.62**
CRAN	200K	**45.5**	**6.98**	**32.7**	**0.61**	**0.62**
SiT	400K	43.3	7.41	36.8	0.61	**0.64**
CRAN	400K	**32.9**	**6.57**	**47.4**	**0.67**	0.63

At 400K steps, CRAN reduces FID from 43.3 to 32.9 and increases IS from 36.8 to 47.4, showing that its benefit persists with a fourfold longer token sequence.

C.3 GENERALIZATION TO ϵ-PREDICTION

Our main experiments use the SiT velocity objective. To test whether the method depends on that formulation, we apply CRAN to DiT-B/2 (Peebles & Xie, 2023) with its original ϵ-prediction objective.

Table 12: Generalization from SiT velocity prediction to DiT ϵ-prediction on ImageNet 256×256.

Method	Steps	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
DiT-B/2	400K	43.5	8.71	32.7	0.49	0.63
DiT-B/2 + CRAN	400K	**38.6**	**8.24**	**37.6**	**0.51**	**0.64**

The consistent improvement across all five metrics indicates that CRAN is not specific to velocity prediction.

C.4 Human motion generation

We further apply CRAN to the Motion Diffusion Model (MDM) (Tevet et al., 2023) on HumanML3D (Guo et al., 2022), where a domain-matched large-scale pretrained representation encoder is unavailable. For an out-of-domain REPA baseline, we render skeletal motion sequences as RGB skeleton frames and extract DINO representations from those renderings.

Table 13: Text-to-motion generation on HumanML3D.

Method	FID↓	R-Precision↑
MDM	0.5320 ± 0.0464	0.6106 ± 0.0044
MDM + REPA	0.5400 ± 0.0391	0.6063 ± 0.0061
MDM + CRAN	$\mathbf{0.3918 \pm 0.0347}$	$\mathbf{0.6694 \pm 0.0075}$

The visual REPA target does not improve MDM in this out-of-domain setting, whereas CRAN substantially improves both distributional quality and text–motion correspondence using only internal diffusion representations.

D Efficiency, sampling robustness, and complementarity

D.1 Computational overhead across model scales

We profile SiT, SRA, REPA, and CRAN under matched hardware, global batch size, precision, software, and data-loading settings. Table 14 reports average per-step time, computation, and peak memory.

Table 14: Training efficiency across backbone scales.

Backbone	Method	Time/step	Ratio	GFLOPs/sample	Peak memory
B/2	SiT	36.92 ms	1.000×	139.2	6.29 GiB
B/2	SRA	41.19 ms	1.116×	173.5	6.41 GiB
B/2	CRAN	41.32 ms	1.119×	173.5	6.45 GiB
B/2	REPA	52.39 ms	1.419×	194.4	6.92 GiB
L/2	SiT	99.01 ms	1.000×	487.4	18.66 GiB
L/2	SRA	108.56 ms	1.096×	601.4	18.83 GiB
L/2	CRAN	109.50 ms	1.106×	601.4	18.84 GiB
L/2	REPA	115.86 ms	1.170×	545.8	19.29 GiB
XL/2	SiT	140.63 ms	1.000×	715.9	26.10 GiB
XL/2	SRA	155.23 ms	1.104×	893.5	26.30 GiB
XL/2	CRAN	156.11 ms	1.110×	893.6	26.32 GiB
XL/2	REPA	156.59 ms	1.114×	774.7	26.73 GiB

The relative per-step overhead of CRAN over SiT remains stable across scales: 11.9% on B/2, 10.6% on L/2, and 11.0% on XL/2. On XL/2, CRAN takes 0.3% less time per step than REPA while using less peak memory.

D.2 MATCHED WALL-CLOCK BUDGETS

Using the measured per-step times in Table 14, 400K CRAN steps match 448K SiT steps, while 400K REPA steps match 507K CRAN steps. Each adjacent pair in Table 15 therefore has the same wall-clock budget.

Table 15: Generation performance under matched wall-clock budgets on SiT-B/2.

Budget	Method	Steps	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
SiT	SiT	448K	33.7	6.59	44.4	0.54	**0.64**
SiT	CRAN	400K	**26.3**	**6.16**	**56.2**	**0.58**	0.63
REPA	REPA	400K	**24.4**	6.40	**59.9**	**0.59**	**0.65**
REPA	CRAN	507K	24.5	**6.12**	59.6	**0.59**	0.64

Under matched time, CRAN clearly outperforms SiT and remains comparable to REPA.

D.3 FIXED SAMPLING PROTOCOL

Guidance-interval tuning introduces an additional degree of freedom. We therefore compare all methods at CFG scale 1.35, applying guidance over the full sampling trajectory without interval scheduling. All results use the same 250-step Euler–Maruyama SDE sampler and 50K-sample evaluation protocol.

Table 16: Comparison under a shared CFG scale and sampling protocol.

Method	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
REPA	**1.80**	4.50	284.0	0.81	**0.61**
SRA	1.89	4.47	278.7	0.81	**0.61**
CRAN	1.83	**4.35**	**286.6**	**0.83**	0.59

Under this fixed protocol, CRAN is comparable to REPA and outperforms SRA overall, with a remaining recall trade-off.

D.4 CFG AND GUIDANCE-INTERVAL SWEEPS

For completeness, Table 17 reports the expanded sweeps used to select the balanced operating point in the main comparison. The main CRAN result uses CFG scale 1.8 and interval $[0, 0.675]$.

Table 17: CFG-scale and guidance-interval sweeps. The REPA entries are its published sweep; SRA is swept around its official setting.

Method	Interval	CFG	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
CRAN	$[0, 0.65]$	1.70	1.52	4.76	277.4	0.80	0.62
CRAN	$[0, 0.65]$	1.80	1.49	4.78	289.6	0.81	0.62
CRAN	$[0, 0.65]$	1.90	1.54	4.83	299.5	0.81	0.61
CRAN	$[0, 0.675]$	1.70	**1.44**	4.75	295.5	0.81	0.64
CRAN	$[0, 0.675]$	1.80	1.45	4.68	306.6	0.81	0.63
CRAN	$[0, 0.675]$	1.90	1.64	4.71	323.9	0.82	0.61
CRAN	$[0, 0.70]$	1.80	1.63	4.56	316.0	0.82	0.61
REPA	$[0, 0.60]$	2.00	1.56	5.11	290.7	0.78	0.66
REPA	$[0, 0.65]$	2.00	1.44	4.88	308.8	0.79	0.65
REPA	$[0, 0.70]$	1.80	**1.42**	4.70	305.7	0.80	0.64
REPA	$[0, 0.70]$	1.90	1.45	4.68	317.6	0.80	0.64
REPA	$[0, 0.70]$	2.00	1.48	4.67	324.0	0.82	0.62
REPA	$[0, 0.75]$	2.00	1.78	4.50	346.2	0.82	0.62
REPA	$[0, 0.80]$	2.00	2.23	4.40	360.9	0.84	0.60
SRA	$[0, 0.675]$	1.80	1.57	4.76	301.5	0.79	0.64
SRA	$[0, 0.70]$	1.70	**1.57**	4.70	294.9	0.79	0.64
SRA	$[0, 0.70]$	1.80	1.58	4.65	309.4	0.80	0.63
SRA	$[0, 0.70]$	1.90	1.69	4.61	325.9	0.81	0.63
SRA	$[0, 0.725]$	1.80	1.67	4.56	320.3	0.81	0.63

D.5 Complementarity with REPA

REPA aligns each noisy diffusion feature with a corresponding external DINOv2 target learned for discriminative visual understanding. In contrast, the internal teacher used by CRAN is optimized through the generation objective and retains generation-relevant layout, texture, instance detail, and denoising-trajectory information. The optimization signals also differ: REPA provides a sample-wise pull toward a fixed external target, whereas CRAN adds relational gradients determined by other samples and the adaptive clean anchor. These gradients are generally not collinear, making the two objectives complementary rather than redundant.

Table 18: Adding CRAN to REPA on SiT-B/2.

Steps	Method	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
100K	REPA	49.5	7.00	27.5	0.46	0.59
100K	REPA + CRAN	**42.4**	**6.64**	**32.5**	**0.50**	**0.60**
200K	REPA	33.2	**6.68**	43.7	0.54	**0.63**
200K	REPA + CRAN	**28.5**	6.87	**51.4**	**0.57**	**0.63**
400K	REPA	24.4	**6.40**	59.9	0.59	**0.65**
400K	REPA + CRAN	**21.0**	6.46	**68.6**	**0.60**	**0.65**

Adding CRAN improves FID, IS, and precision at every budget. At 400K steps, it reduces FID from 24.4 to 21.0 while preserving recall.

E Effectiveness of Clean Inter-Sample Geometry Estimation

Our default implementation estimates m_{cran} from the low-noise samples already present in the current batch, thereby avoiding an additional near-clean forward pass. Table 19 shows that this efficient estimate preserves generation quality: its FID of 26.3 is within 0.1 of the full m_{cran} computation.

Table 19: FID comparison on ImageNet 256×256 using SiT-B/2, 400K training steps, and no CFG. The estimated m_{cran} variant is our default.

Model	#Params	Iter.	FID↓
SiT-B/2	130M	400k	33.0
+ SRA	130M	400k	29.1
+ SRA & Disp.	130M	400k	28.6
+ DiverseDiT	>130M	400k	28.1
+ SRA2	130M	400k	28.9
+ CRAN, full m_{cran}	130M	400k	**26.2**
+ CRAN, estimated m_{cran}	130M	400k	26.3

We further compare the estimated and full anchors throughout training. Figure 7 reports 1K-step local averages every 2K steps from 30K to 400K. The estimated trajectory closely tracks the full computation: the mean absolute difference in cosine distance is 0.010 over the dense trajectory, and the final values differ by only 0.006.

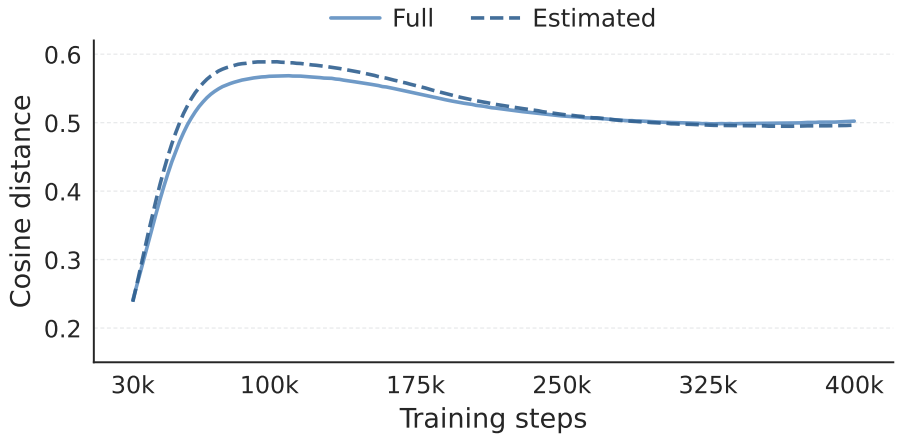


Figure 7: Dense trajectories of the estimated and full m_{cran} throughout SiT-B/2 training. Curves show 1K-step local averages sampled every 2K steps.

Finally, we quantify the saved computation using the same optimized CRAN benchmark as Table 14. We change only the source of m_{cran} and run the full and estimated variants in two paired timing runs under the same concurrent H200 load. The average measured time ratio is 0.956. Applying this ratio to the separately measured 43.20 ms full-geometry time gives 41.32 ms per step, corresponding to a 4.4% speedup.

Table 20: SiT-B/2 training cost of the full and estimated clean inter-sample geometry. The estimated time is obtained by scaling the full-geometry time with the paired measured ratio.

m_{cran} computation	Time/step	Relative time
Full	43.20 ms	1.000×
Estimated	**41.32 ms**	**0.956**×

F Experimental setup

Hyperparameter settings. Our default hyperparameter settings are summarized in Table 21.

Table 21: Hyperparameter settings.

Backbone	**SiT-B**	**SiT-L**	**SiT-XL**
Architecture			
Input	$32 \times 32 \times 4$	$32 \times 32 \times 4$	$32 \times 32 \times 4$
Patch size	2	2	2
Layers	12	24	28
Hidden dim.	768	1,024	1,152
Num. heads	12	16	16
CRAN settings			
Contrastive between internal representations			
Contrastive blocks	$8 \leftrightarrow 3$	$16 \leftrightarrow 6$	$20 \leftrightarrow 8$
Contrastive time interval	[0, 0.2)	[0, 0.2)	[0, 0.2)
λ	1.0	1.0	1.0
Contrastive between internal representations and DINO representations			
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 Scale	-	-	1.8

Computational resources. All training experiments are conducted on 8 NVIDIA H200 GPUs, with a global batch size of 256 and 32 samples per GPU.

G METRICS

We evaluate image generation quality from complementary perspectives using standard quantitative metrics.

Fréchet Inception Distance (FID) (Heusel et al., 2017) measures the distributional similarity between real and generated images in the feature space of an Inception-v3 network. Lower FID indicates that generated samples are closer to the real data distribution.

Structural FID (sFID) (Nash et al., 2021) extends FID by computing the distance on spatial feature maps, making it more sensitive to structural and spatial coherence.

Inception Score (IS) (Salimans et al., 2016) evaluates both image quality and class-conditional diversity by measuring the discrepancy between confident per-image class predictions and the marginal class distribution over generated samples. Higher IS is better.

Precision and recall (Kynkäänniemi et al., 2019) separate sample fidelity from distribution coverage. Precision reflects how realistic generated samples are relative to real images, while recall measures how well the generated samples cover the target data manifold.

To investigate how CRAN affects learned representations, in addition to linear probing accuracy, we follow iREPA (Singh et al., 2025) and use spatial structure metrics to measure patch-level organization.

Spatial self-similarity metrics. These metrics analyze whether intermediate representations preserve spatial structure. Given patch features $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^{T}$ arranged on an $H \times W$ lattice, we define the cosine self-similarity kernel as $K_{\mathbf{X}}(t, t') = \langle \mathbf{x}_t, \mathbf{x}_{t'} \rangle / (\|\mathbf{x}_t\|_2 \|\mathbf{x}_{t'}\|_2)$. The Manhattan distance between two patch locations is denoted as $d(t, t')$.

Local vs. Distant Similarity (LDS) measures the contrast between local and distant patch similarities:

$$\mathrm{LDS}(\mathbf{X}) = \mathbb{E}[K_{\mathbf{X}}(t,t') \mid d(t,t') < r_{\mathrm{near}}] - \mathbb{E}[K_{\mathbf{X}}(t,t') \mid d(t,t') \geq r_{\mathrm{far}}]\,. \tag{9}$$

A larger LDS indicates that nearby patches are more similar than far-away patches, suggesting stronger spatial organization.

Correlation Decay Slope (CDS) quantifies how fast patch similarity decays with spatial distance. We first compute the spatial correlogram $g_{\mathbf{X}}(\delta) = \mathbb{E}[K_{\mathbf{X}}(t,t') \mid d(t,t') = \delta]$ for $\delta \in \{1,\dots,\Delta\}$. After fitting a least-squares line $g_{\mathbf{X}}(\delta) \approx \alpha + \beta\delta$, CDS is defined as

$$\mathrm{CDS}(\mathbf{X}) = -\hat{\beta}. \tag{10}$$

Higher CDS corresponds to a sharper decay of similarity over distance and therefore stronger spatial structure.

RMS Spatial Contrast (RMSC) measures the spatial variation of normalized patch features. Let $\hat{\mathbf{x}}_t = \mathbf{x}_t / \|\mathbf{x}_t\|_2$ and $\bar{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{x}}_t$. RMSC is defined as

$$\mathrm{RMSC}(\mathbf{X}) = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \|\hat{\mathbf{x}}_t - \bar{\mathbf{x}}\|_2^2}. \tag{11}$$

Higher RMSC indicates stronger spatial diversity across patch tokens, while lower RMSC suggests that the representation becomes spatially uniform.

H Uncurated Samples

We show uncurated samples generated by SiT-XL with `CRAN` on ImageNet 256×256, using classifier-free guidance with $w = 4.0$.



Figure 8: Generated images from SiT-XL/2 + `CRAN` conditioned on the class label “macaw” (88).



Figure 9: Generated images from SiT-XL/2 + CRAN conditioned on the class label “loggerhead” (33).



Figure 10: Generated images from SiT-XL/2 + CRAN conditioned on the class label “cheeseburger” (933).



Figure 11: Generated images from SiT-XL/2 + CRAN conditioned on the class label “great grey owl” (24).



Figure 12: Generated images from SiT-XL/2 + CRAN conditioned on the class label “castle” (483).



Figure 13: Generated images from SiT-XL/2 + CRAN conditioned on the class label “ice cream” (928).



Figure 14: Generated images from SiT-XL/2 + CRAN conditioned on the class label “volcano” (980).



Figure 15: Generated images from SiT-XL/2 + CRAN conditioned on the class label “lakeside” (975).



Figure 16: Generated images from SiT-XL/2 + CRAN conditioned on the class label “sulphur-crested cockatoo” (89).



Figure 17: Generated images from SiT-XL/2 + CRAN conditioned on the class label “koala” (105).

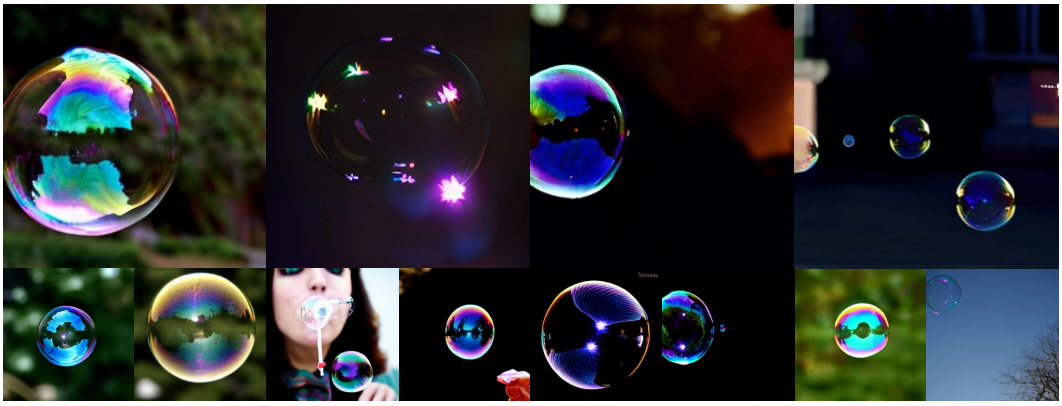


Figure 18: Generated images from SiT-XL/2 + CRAN conditioned on the class label “bubble” (971).



Figure 19: Generated images from SiT-XL/2 + CRAN conditioned on the class label “polecat” (358).



Figure 20: Generated images from SiT-XL/2 + CRAN conditioned on the class label “Blenheim spaniel” (156).

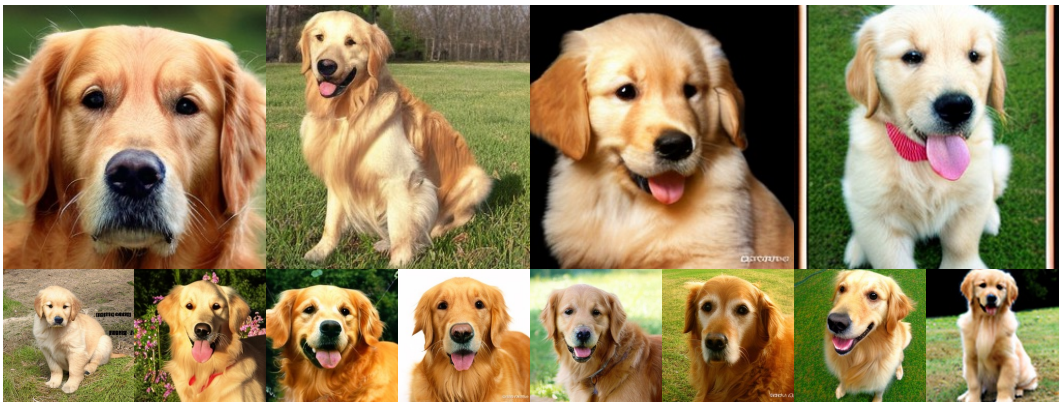


Figure 21: Generated images from SiT-XL/2 + CRAN conditioned on the class label “golden retriever” (207).



Figure 22: Generated images from SiT-XL/2 + CRAN conditioned on the class label “Border collie” (232).



Figure 23: Generated images from SiT-XL/2 + CRAN conditioned on the class label “coral reef” (973).

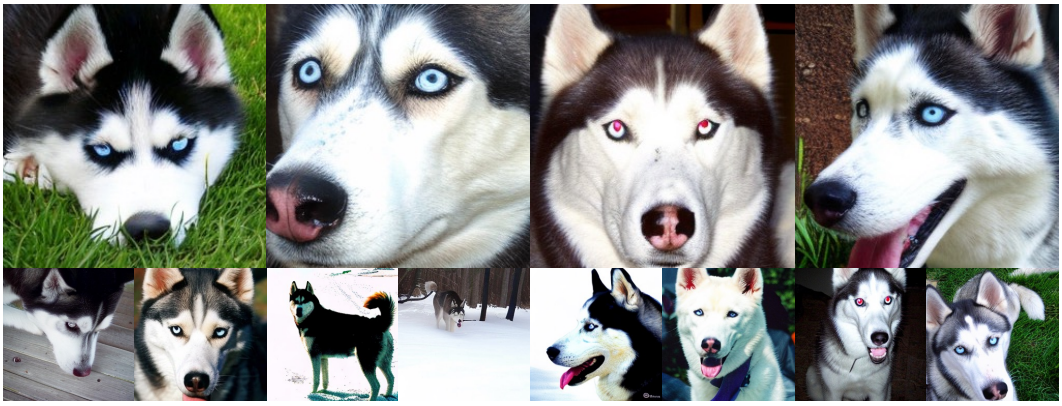


Figure 24: Generated images from SiT-XL/2 + CRAN conditioned on the class label “Siberian husky” (250).



Figure 25: Generated images from SiT-XL/2 + CRAN conditioned on the class label “balloon” (417).



Figure 26: Generated images from SiT-XL/2 + CRAN conditioned on the class label “lion” (291).



Figure 27: Generated images from SiT-XL/2 + CRAN conditioned on the class label “china cabinet” (495).