ENHANCING DIFFUSION MODELS TOWARDS UNIFIED GENERATIVE AND DISCRIMINATIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

While diffusion models excel at image synthesis, their generative pre-training has been shown to yield useful representations, paving the way towards unified generative and discriminative learning. However, their potential is hindered by an architectural limitation: the model's intrinsic semantic information flow is potentially sub-optimal. The features encoding the richest high-level semantics are often underutilized and diluted in decoding layers, impeding the formation of a strong representation bottleneck. To address this, we introduce *self-conditioning*, a lightweight mechanism that reshapes the model's layer-wise semantic hierarchy *without external guidance*. By aggregating and rerouting the richest intermediate features to guide its own decoding layers, our method concentrates more high-level semantics, concurrently strengthening generative guidance and forming a more discriminative representation. Results are compelling: this approach demonstrates a consistent dual-improvement trend across models and architectures with minimal overhead. Crucially, it creates an architectural semantic bridge that enables an effective integration of other discriminative techniques, such as contrastive *self-distillation*, to further amplify gains. Extensive experiments show that our enhanced models, particularly pixel-space UViT and latent-space DiT, become powerful unified learners, surpassing various self-supervised models in linear evaluation while also improving or maintaining high generation quality.

1 INTRODUCTION

Diffusion models (Ho et al., 2020) have recently emerged as one of the most powerful and popular techniques in generative AI, renowned for their ability to synthesize photorealistic visual data. These models have demonstrated remarkable versatility and high performance across a spectrum of tasks, such as class-conditional generation (Peebles & Xie, 2023; Karras et al., 2024), text-to-image synthesis (Rombach et al., 2022; Esser et al., 2024) and image editing (Mokady et al., 2023; Brooks et al., 2023), along with flexible customization options (Ruiz et al., 2023; Zhang & Agrawala, 2023).

Meanwhile, interest has grown in repurposing pre-trained diffusion models for discriminative tasks. Studies have shown that the intermediate representations (Baranchuk et al., 2022; Xiang et al., 2023) are effective for downstream tasks, particularly dense prediction like segmentation (Xu et al., 2023), depth estimation (Zhao et al., 2023) and keypoint detection (Xu et al., 2024). These findings highlight diffusion's potential as **unified** generative-and-discriminative learners, which gain a deep understanding of data through generative pre-training, as in language models (Chen et al., 2020a).

Despite these advances, diffusion models still face challenges in representation learning: achieving optimal performance often necessitates specialized methods for feature extraction (Meng et al., 2024), distillation (Li et al., 2023a), or design of dedicated decoders (Zhao et al., 2023). Moreover, while adept at capturing local semantics crucial for dense prediction, their global feature quality often underperforms modern self-supervised learners, especially in image-level tasks such as linear classification (Hudson et al., 2023; Chen et al., 2024). This discrepancy underscores a limitation in their ability to *condense high-level semantics into a compact, discriminative feature*.

Unlike paradigms imposing an explicit information bottleneck, such as view alignment in contrastive methods (Wang & Isola, 2020) or asymmetric encoder-decoder design in masked autoencoders (He et al., 2022), diffusion models distribute semantic information across all layers, with each offering representations of varying granularity (Baranchuk et al., 2022). While this dispersion is an emergent

and natural outcome of generative pre-training, it poses a fundamental challenge for representation learning, as no single layer is explicitly designed as a semantic bottleneck (Hudson et al., 2023).

Prior works such as DDAE (Xiang et al., 2023) have revealed that these emergent features form a layer-wise hierarchy, with richest ones residing in intermediate layers. However, these diagnostic works only evaluated pre-trained models without enhancing them. Moving from diagnosis to intervention, we posit that *the most semantically rich features are sub-optimally routed and consequently diluted in decoding*. This flawed information flow not only weakens guidance for synthesizing global structures but also impedes the formation of a stronger representation bottleneck. To resolve this, we introduce **self-conditioning**, a mechanism to reshape model's intrinsic semantic hierarchy by rerouting its information flow, resulting in a tighter bottleneck (Fig. 5).

Table 1: **Trade-offs and gains in generation and representation.** We enhance standard diffusion models in both domains, without requiring major framework overhauls or external knowledge.

	Generative Modeling		Representation Learning	
	Standard, generalizable	Sample quality++	Self-supervised, no extra encoder	Feature quality++
l-DAE (Chen et al., 2024)	×	×	✓	✓
SODA (Hudson et al., 2023)	×	✓	✓	✓
RCG (Li et al., 2023b)	✓	✓	×	×
REPA (Yu et al., 2025)	✓	✓	×	✓
DDAE (Xiang et al., 2023)	✓	×	✓	×
DDAE++ (Ours)	✓	✓	✓	✓

Our high-level principle is simple: aggregate a rich semantic feature from an intermediate layer, and use it to condition subsequent decoding layers. To implement this efficiently, our strategy is to *reuse the inherent conditioning pathways already present in diffusion backbones*, allowing for tailored yet minimal modifications: for UNet (Dhariwal & Nichol, 2021) and DiT (Peebles & Xie, 2023), we leverage their adaptive normalizations by injecting a global pooled feature vector after a time-adaptive transformation (Fig. 1); for in-context conditioning UViT (Bao et al., 2023a), we introduce an additional token to automatically aggregate features and condition patch tokens via self-attention (Fig. 2). This flexible reuse strategy makes our approach a plug-and-play enhancement with minimal computational overhead, and demonstrates a broad applicability across diverse architectures.

The immediate outcome of self-conditioning is a dual benefit: improved sample and feature quality. More importantly, it forges an **architectural semantic bridge** that explicitly connects discriminative bottleneck to its generative decoding path. This bridge unlocks further potentials: dedicated representation learning methods, such as contrastive learning (Chen et al., 2021), can now be integrated to directly refine the bottleneck feature. In turn, this enhanced feature is fed back through the bridge, providing stronger semantic guidance for generation. Our full framework, DDAE++, combines these synergistic components with self-conditioning to further amplify gains in both domains.

Extensive experiments demonstrate a consistent dual-improvement trend, where accuracy is significantly boosted while FID is either improved or maintained at state-of-the-art levels, a contrast to prior works that often sacrifice generative abilities (Hudson et al., 2023; Chen et al., 2024). Particularly noteworthy are UViT and DiT, which, with our enhancements, exhibit exceptional potential for representation learning, surpassing various self-supervised models. Crucially, the most substantial gains from discriminative techniques are observed when paired with self-conditioning, highlighting the synergy our approach facilitates between generative and discriminative paradigms.

In summary, our key contributions are:

- **Conceptual**: We identify and address a core limitation in diffusion models, *i.e.*, their sub-optimal semantic information flow, thereby intrinsically and concurrently enhancing their dual capabilities.
- **Methodological**: We propose self-conditioning, a lightweight mechanism that repurposes conditioning pathways to tighten semantic bottleneck and amplify synergistic discriminative methods.
- **Empirical**: Evaluation across models, backbones and datasets reveals a clear dual-improvement trend, and shows our enhanced UViT and DiT are powerful unified learners with good scalability.

2 Related work

Self-supervised learning (SSL) has established two dominant paradigms with distinct properties. Contrastive learning (CL) distills image-level semantics into a compact feature, pulling augmented views of an image closer. This instance discrimination (Wu et al., 2018) makes it highly effective for linear evaluation (Grill et al., 2020; Chen et al., 2021). Masked image modeling (MIM), conversely, learns by reconstructing corrupted inputs, akin to denoising autoencoding (Vincent et al., 2008). While this generative process preserves rich, transferable information, the resulting features are less

compact and discriminative (Bao et al., 2021; He et al., 2022). As a form of denoising autoencoders (Xiang et al., 2023; Chen et al., 2024), diffusion naturally inherits a similar trade-off. Our work operates at the intersection of these paradigms to address it. We first enhance semantic aggregation and utilization via an architectural modification, and then integrate an explicit contrastive loss, aligning with the trend of hybrid methods that unify CL and MIM (Zhou et al., 2022; Huang et al., 2023).

Semantic-enhanced diffusion models. Unconditional models often lag behind those conditioned on semantic cues in terms of generation quality (Bao et al., 2022). To bridge this gap, some studies leverage external vision foundation models to provide semantic guidance, which can be in the form of pseudo-labels derived from clustering (Hu et al., 2023a;b) or direct feature injection (Li et al., 2023b). However, reliance on external signals can compromise the flexibility of native unconditional applications (*e.g.*, image translation (Su et al., 2023) and domain adaptation (Liu et al., 2023)). Recent methods like REPA (Yu et al., 2025) use external features as a distillation target to regularize the output, rather than as an input condition. In contrast, our work is entirely self-contained. We posit that powerful semantic cues for guidance already exist within model's feature hierarchy, but remain sub-optimally utilized. By focusing on improving the *internal* information flow, our approach fully preserves original model's flexibility, and is potentially orthogonal to *external* methods like REPA.

3 BACKGROUND

Diffusion models. Diffusion (Ho et al., 2020; Karras et al., 2022) and flow-based (Lipman et al., 2023; Liu et al., 2023) models are state-of-the-art generative models with strong theoretical connections (Esser et al., 2024). These models, hereinafter collectively referred to as *diffusion models*, construct paths that progressively transform data into noise via a time-forward process over $t \in [0, T]$:

$$x_t = \alpha_t x_0 + \sigma_t \epsilon \quad \text{where } \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

with schedules for α_t, σ_t such that $t = 0$ corresponds to p_{data} and $t = T$ approximates $\mathcal{N}(0, I)$. To reverse this, an ordinary differential equation (ODE) is typically formulated (Song et al., 2021b):

$$\mathrm{d}x_t = v_\theta(x_t, t)\mathrm{d}t, \tag{2}$$

where velocity $\mathrm{d}x_t/\mathrm{d}t$ is parameterized by a time-conditioned network v_θ, which, once trained, enables ODE solvers to generate data. The estimation of v_θ is closely related to denoising autoencoding and score matching (Song et al., 2021b), allowing the training objective to be reparameterized into flexible forms, typically including training the network to predict the added noise ϵ, the clean data x_0, or the velocity. For simplicity, we refer to any such objective as *diffusion loss*, denoted by $\mathcal{L}_{\text{diff}}$.

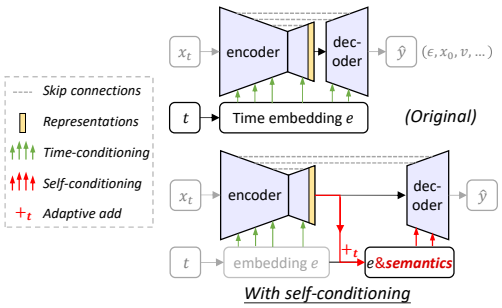
Different formulations of diffusion models may vary in noise schedule, training objective and ODE solver. In this paper, we examine three representative ones: DDPM (Ho et al., 2020), EDM (Karras et al., 2022) and Rectified Flow (RF) (Liu et al., 2023), with detailed overview provided in Appx. A.

Backbones for diffusion models. Early works like DDPM adapt a UNet (Ronneberger et al., 2015) architecture, where time t is specified through a global conditioning pathway, by injecting sinusoidal embeddings (Vaswani et al., 2017) of t into each block. We refer to this basic version as `ddpm`[1]. DDPM++ (Song et al., 2021b) further builds upon this, enhancing its capacity by doubling depth and employing BigGAN-style blocks (Brock et al., 2018). We refer to this scaled-up variant as `ddpmpp`.

Recently, ViT-based backbones have demonstrated better scalability (Bao et al., 2023b; Esser et al., 2024). We examine two designs with distinct conditioning mechanisms: UViT (Bao et al., 2023a), which processes all inputs (including time and other conditions) as tokens in a transformer; and DiT (Peebles & Xie, 2023), which injects conditions via AdaLN, analogous to UNet's AdaGN (Dhariwal & Nichol, 2021). The architectural similarity of these backbones to those used in visual recognition (Dosovitskiy et al., 2021) also motivates our investigation into their representation learning potential.

Motivation for a comprehensive baseline. Most related studies focus on extracting features from one specific model (Baranchuk et al., 2022; Yang & Wang, 2023), with few efforts to identify which models (and what factors) lead to better representations. While DDAE first suggested a link between generation and recognition by comparing DDPM and EDM, its analysis had limitations: the comparison lacked strict control over confounding factors (*e.g.*, backbone sizes, augmentations), and its scope did not cover modern advances like flow-based models or ViT-based backbones.

[1]We distinguish backbones (notations like `ddpm`) from models (DDPM), which can be combined flexibly.



(a) **UNet** with self-conditioning: an illustration.

```
# blockno    : the layer index between enc-dec
# t_embedder : a standalone timestep embedder
# proj       : a single nn.Linear layer

def dit_forward(x, t, c=None):
    x = x_embedder(x) + pos_embed          # (N, T, D)
    e = t_and_c_embedder(t, c)             # (N, D)
    for no, block in enumerate(dit_blocks):
        x = block(x, e)                    # (N, T, D)
        if no + 1 == blockno:
            global_feat = x.mean(dim=1)    # -->(N, D)
            e = e + t_embedder(t) * proj(global_feat)

    x = final_layer(x, e)
    return unpatchify(x)
```

(b) **DiT** with self-conditioning: PyTorch-like pseudocode.

Figure 1: **Self-conditioning applied to backbones based on adaptive normalization.** Originally, time t (and optional condition c) are specified via a global conditioning pathway, where their embedding e is injected into all layers. Here, we collect features from a specific layer by average pooling, and add them to the pathway of decoding layers, after being projected and time-adaptively scaled.

To better understand the properties of diffusion representations, and to validate our method's generalizability, we establish a more comprehensive and controlled baseline. It spans multiple model formulations, backbones (of different types and sizes) and datasets, providing a solid foundation to demonstrate that our method yields consistent benefits across diverse settings.

4 Approach

Our DDAE++ framework enhances diffusion models by systematically addressing their sub-optimal and under-utilized representations. It is built upon three necessary and complementary perspectives: 1) The **architectural** foundation is our core contribution, *self-conditioning*, which better produces and utilizes the semantics, and creates a robust bridge between representation and generation pathways. The bridge is a critical enabler for two other perspectives, translating further improved semantics into improved generation. 2) From the **data-space**, we employ *non-leaky augmentations* to generally improve representation robustness. 3) From the **objective-space**, we introduce *contrastive self-distillation* to directly refine the newly formed bottleneck. Together, these architectural, data-space and objective-space components form a complete and coherent system for dual enhancement.

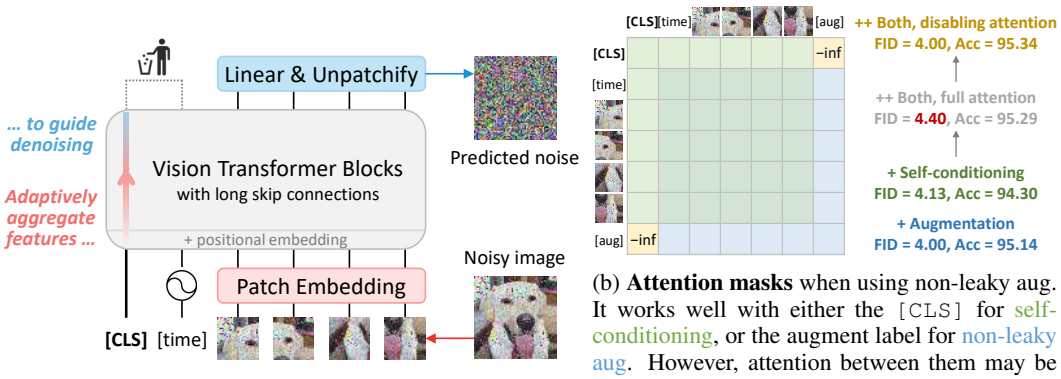
4.1 Self-conditioning

The essence of self-conditioning is to create a feedback loop: aggregating useful features from an intermediate layer and rerouting them to guide subsequent decoding layers. However, a conflict arises due to diffusion model's nature: effective representation often occurs in a narrow range of timesteps, while high-quality generation relies on the entire trajectory. The aggregated features, while valuable within a certain range, can be too noisy to provide beneficial guidance in others. Therefore, our implementations are designed to be **time-adaptive**, dynamically modulating the features. Specifically:

For UNet and DiT, our approach re-injects features from an intermediate layer back into the decoder's conditioning pathway. As illustrated in Fig. 1a, we first apply global average pooling to the encoded feature map of a designated layer to obtain a feature vector. This vector is then projected and modulated by a learned, time-dependent scaling factor before being added to the original conditioning embedding (details in Fig. 1b). This enriched embedding is propagated to all rest layers.

Identifying this bottleneck layer efficiently and without data leakage is crucial. We employ a search procedure based on training dynamics: guided by architectural heuristics, we first select a small set of candidates (*e.g.*, layer 8-11 in DiT). We then launch short, parallel training runs for each, and select the one yielding lowest training loss after a few epochs as the final bottleneck. As validated in ablation studies, this training-loss-based strategy serves as a reliable proxy for final performance.

For UViT, our approach automates the entire self-conditioning process, *i.e.*, semantic aggregation, time-adaptive modulation and decoding guidance, through an attention-native mechanism. Since all information, not only patch tokens but also the time token, is globally and continuously interacted, a



(a) **UViT** with [CLS] token, which automatically achieves time-adaptive semantic aggregation and decoding guidance.

(b) **Attention masks** when using non-leaky aug. It works well with either the [CLS] for self-conditioning, or the augment label for non-leaky aug. However, attention between them may be harmful when used together, while disabling this interaction resolves the issue.

Figure 2: **Self-conditioning applied to backbones based on in-context attention.** Based on the all-as-tokens design, we leverage an additional token to automatically interact with patch tokens and the time token, eliminating the need for manual feature selection, pooling, modulation or rerouting.

dedicated summary token is more natural than layer-wise pooling and vector scaling. We therefore introduce an additional learnable token, initialized randomly and conceptually empty, to dynamically aggregate and utilize global semantics as it propagates through the transformer (Fig. 2a).

We refer to this token as [CLS] due to its functional similarity to ViT's class token, and this design may also connect to register tokens (Darcet et al., 2024) and visual prompts (Jia et al., 2022). Note that despite the nomenclature, we *do not* apply any regularization to this token at this stage (learned solely through diffusion), in contrast to the class token used in supervised ViT training.

4.2 Adapting Discriminative Learning Techniques

Non-leaky augmentations, the data-space component of our framework, serve a dual purpose: generally enhancing features and creating positive views for contrastive learning. To avoid harmful artifacts introduced by aggressive SSL transformations (Chen et al., 2020b), we adopt the weaker, non-leaky geometric pipeline from EDM (Karras et al., 2022). A vector of transformation parameters is used to condition the model, further preventing augmentation effects from leaking into generated samples. While preserving generative quality, this choice presents a trade-off, as these transformations may yield less diverse views and potentially limit contrastive effectiveness (Tian et al., 2020).

We also find that [CLS] and the augmentation token can interfere with each other through attention, weakening the bottleneck. Disabling their direct interaction restores performance gains (Fig. 2b).

EMA rate	FID↓	Acc.↑
0.999	9.17	60.06
0.9993	8.99	60.18
0.9999	**8.81**	**60.47**

Contrastive self-distillation is employed to directly refine the bottleneck feature self-conditioning operates on. A key insight is that diffusion models already maintain a high-quality "teacher" model through exponential moving averaging (EMA) of weights, a standard practice to improve generation quality (Karras et al., 2024). Intriguingly, our preliminary study reveals that the EMA decay rate optimal for generation is also optimal for representation. This alignment suggests that the EMA model, originally intended for generation, serves as a powerful and readily available teacher for discriminative learning, obviating the need for a separate momentum encoder.

Leveraging this, we adapt MoCo v3 (Chen et al., 2021) for self-distillation. For a input view, features from the bottleneck layer of the online model (*i.e.*, the pooled feature or the [CLS] token) are passed through a time-dependent MLP projection head. These are trained to align with target features from EMA model, which are extracted from the same layer using an augmented view, but at a fixed, optimal timestep for classification. The final objective is a weighted sum of diffusion loss and contrastive loss: $\mathcal{L} = \mathcal{L}_{\text{diff}} + \gamma \mathcal{L}_{\text{MoCo}}$. Detailed formulation and illustration are provided in Appx. A.

Summary. Our full framework effectively unifies denoising autoencoding and instance discrimination within diffusion models based on an architectural enhancement. Studies with related concepts, such as SODA (Hudson et al., 2023) and REPA (Yu et al., 2025), are discussed in Appx. C.

Table 2: **A comprehensive comparison of generative and discriminative performances of self-supervised diffusion models. Baseline:** We extend to more models and backbones, clarifying the impact of each design factor. **+ Self-conditioning:** We introduce a simple method that leverages discriminative features within denoising network to guide generation by itself. As a generalizable enhancement, it improves both metrics in most cases, especially on the more diverse CIFAR-100. Colored values indicate gains (or degradation). Best results for each backbone are in bold.

Unconditional CIFAR-10 Generation & Linear Probing

Model	Backbone	Baseline		+ Self-conditioning	
		FID↓	Acc.↑	FID↓	Acc.↑
DDPM	ddpm	3.60	90.34	3.67 (+0.07)	90.94 (+0.60)
EDM	ddpm	3.39	**91.41**	**3.30** (-0.09)	91.07 (-0.34)
RF	ddpm	3.89	90.67	3.67 (-0.22)	90.91 (+0.24)
DDPM	ddpmpp	2.98	94.02	2.75 (-0.23)	94.34 (+0.32)
EDM	ddpmpp	2.23	94.83	**2.18** (-0.05)	**94.85** (+0.02)
RF	ddpmpp	2.54	93.97	2.42 (-0.12)	93.72 (-0.25)
DDPM	UViT-S	4.48	93.67	**4.13** (-0.35)	**94.30** (+0.63)

Unconditional CIFAR-100 Generation & Linear Probing

Model	Backbone	Baseline		+ Self-conditioning	
		FID↓	Acc.↑	FID↓	Acc.↑
DDPM	ddpm	5.97	62.06	**5.77** (-0.20)	63.55 (+1.49)
EDM	ddpm	6.21	63.68	6.01 (-0.20)	**65.35** (+1.67)
RF	ddpm	6.49	60.84	6.25 (-0.24)	62.83 (+1.99)
DDPM	ddpmpp	4.43	69.35	4.01 (-0.42)	71.11 (+1.76)
EDM	ddpmpp	3.46	71.09	**3.36** (-0.10)	**71.61** (+0.52)
RF	ddpmpp	4.07	67.46	4.29 (+0.22)	68.49 (+1.03)
DDPM	UViT-S	7.35	70.66	**7.14** (-0.21)	**72.04** (+1.38)

5 EXPERIMENTS

We present a series of experiments evaluating the efficacy, generalizability, scalability and the synergistic behavior of our method. In particular, we address the following research questions:

- How effective is our method in concurrently improving dual metrics on diverse baselines? (Tab. 2)
- How does each component interact and contribute to the overall improvements? (Tab. 3, Fig. 3)
- Does self-conditioning remain effective and scalable with DiT on challenging datasets? (Tab. 4, Fig. 4) How does it work? (Fig. 5, 6)

Implementation details. Our pixel-space experiments build upon DDAE (Xiang et al., 2023), after stabilizing some hyper-parameters in feature evaluation. Consequently, Tab. 2 represents our controlled and overhauled setup, so we do not report its original results. For latent-space models, we follow the state-of-the-art training recipe, Lightning-DiT (Yao et al., 2025), for fast convergence. FID (Heusel et al., 2017) and IS (Salimans et al., 2016) are used to measure sample quality. The standard linear probing protocol is used to measure feature quality, where the backbone is frozen and no fine-tuning is performed. Details on training, evaluation, layer/timestep hyper-parameters and other SSL methods in comparison, are specified in Appx. B, and qualitative results are in Appx. D.

5.1 MAIN PROPERTIES

Observations from baseline models. Tab. 2 shows the performance of our re-implemented, unconditional baselines, encompassing combinations[2] of models and backbones described in Sec. 3. Regarding sample quality, EDM generally achieves best FID, consistently outperforming RF, another continuous-time model, suggesting that designing a suitable noise schedule is crucial. Meanwhile, UViT still lags behind UNet in pixel-space generation. For classification, while larger models tend to yield higher results, the model formulation is also crucial. Notably, EDM consistently delivers the highest accuracy, even when it does not achieve the lowest FID (*e.g.*, EDM-`ddpm` on CIFAR-100), suggesting that generative and discriminative qualities do not naturally align in baseline models.

Self-conditioning improves both metrics. Tab. 2 shows a dual-improvement trend: it not only improves FID but also significantly boosts feature quality, particularly on the more diverse CIFAR-100 where accuracy increases by up to 1.99%, a substantial gain in the SSL context. Remarkably, this is achieved with negligible overhead: a mere 0.5M parameters for UNet and one more token for UViT.

We now extend our analysis to assess how self-conditioning interacts with discriminative techniques, namely data augmentations (Aug) and contrastive learning (CL), with results presented in Tab. 3.

Aug alone is not helpful enough. Using Aug alone provides mixed results: it slightly improves FID on CIFAR-10 but can degrade it notably on CIFAR-100. This suggests the geometric pipeline, tuned by EDM for CIFAR-10, may not generalize well, highlighting a key challenge for diffusion-based unified learning. Crucially, the fact that FID can worsen even when feature quality improves (*e.g.*, 7.35 to 7.96) indicates a disconnect: enhanced features are not effectively leveraged for generation.

[2]We find UViT to be unstable with EDM and RF, so these two are excluded.

Table 3: **Component-wise analysis and comparison with other SSL methods.** Self-conditioning, when using in conjunction with Aug and/or CL, brings significant improvements and can surpass other self-supervised or diffusion-based models. Best results for each backbone are in bold.

Model	Backbone	Aug	CL	CIFAR-10		CIFAR-100	
				FID↓	Acc.↑	FID↓	Acc.↑
EDM	ddpm			3.39	91.41	6.21	63.68
		✓		3.32	92.55	5.39	66.28
		✓	✓	3.42	92.61	5.42	67.63
	+ 0.46M (1.3%)			3.30	91.07	6.01	65.35
		✓		**3.08**	92.41	**5.13**	68.65
		✓	✓	3.14	**92.97**	5.25	**69.50**
EDM	ddpmpp			2.23	94.83	3.46	71.09
		✓		2.19	95.24	3.61	71.14
	+ 0.46M (0.8%)			2.18	94.85	3.36	71.61
		✓		2.17	95.33	3.38	**73.29**
		✓	✓	**2.14**	**95.35**	**3.35**	72.88
DDPM	UViT-S			4.48	93.67	7.35	70.66
		✓		4.00	95.14	7.96	72.85
	+ 1 token (0.4%)			4.13	94.30	7.14	72.04
		✓		**4.00**	**95.34**	7.05	73.58
		✓	✓	4.35	95.28	**6.88**	**74.48**
Contrastive	resnet18	✓	✓	89.17–93.10		64.88–70.90	
Contrastive	resnet50	✓	✓	90.88–93.89		66.15–72.51	
MIM-based	ViT-B	crop		61.70–70.20		—	
MDM	unet				94.80		—
SODA#	res18+unet	✓			80.00		54.90

#Official code unavailable; we build a simplified version based on core principles.

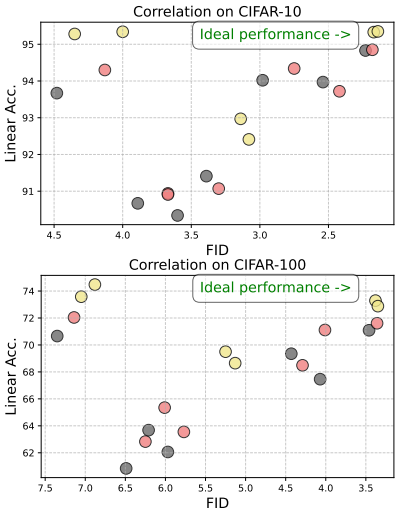


Figure 3: **Correlation between generation and discrimination.** Original diffusion baselines (in gray) only show relatively weak linear correlation. Our self-conditioning, aug and contrastive enhancements make it more significant by simultaneously and gradually boosting both metrics towards the upper-right ideal region.

Self-conditioning directly addresses this issue. When applied jointly with Aug, it not only counteracts the FID degradation on CIFAR-100 (*e.g.*, 7.35 to 7.05), but further amplifies FID gains when Aug alone yields marginal ones (*e.g.*, 3.32 *vs.* 3.08). Our architectural bridge ensures that improved representations from Aug are effectively utilized, leading to better outcomes in both domains.

CL and diffusion are complementary. While adding CL can sometimes cause a slight trade-off with FID, it is mitigated by self-conditioning, and it suggests the information generated by CL may differ from that needed in diffusion. This can actually be beneficial to building stronger representations: our fully combined method consistently achieves high accuracy (the highest in most cases).

Putting all together. Fig. 3 visualizes the evolving relationship between generation and discrimination. In baseline models, the correlation (discussed in Xiang et al. (2023); Yu et al. (2025)) is relatively weak, indicating that better generative models do not necessarily yield better representations (in line with Chen et al. (2024)). In contrast, our method establishes a much clearer positive correlation, demonstrating its ability to bridge the gap between the two domains. This effect is particularly pronounced for UViT. Although the accuracy is initially outperformed by UNet, our enhancements unlock its potential, enabling it to surpass `ddpmpp` with less compute and achieve highest results in Tab. 3. This suggests that, despite sub-optimal in pixel generation compared to UNet, UViT possesses unique potential to learn powerful representations, which we successfully unlock, even in this low-data regime where ViTs typically struggle (Dosovitskiy et al., 2021).

5.2 Performance and Scalability on ImageNet

To assess our method's effectiveness on more challenging, large-scale datasets, we conduct experiments on ImageNet-256x256 with latent-space DiTs. Our analysis demonstrates strong performance in a standard setting, alongside excellent scalability with respect to training duration and model size.

Fig. 4 reveals a more comprehensive picture: our models consistently outperform their respective baselines in FID, IS and linear probing accuracy throughout the training progress. Most importantly, the performance gap does not narrow; in fact, the accuracy gain widens as training proceeds, highlighting a desirable property similar to other SSL methods (Chen et al., 2021; He et al., 2022).

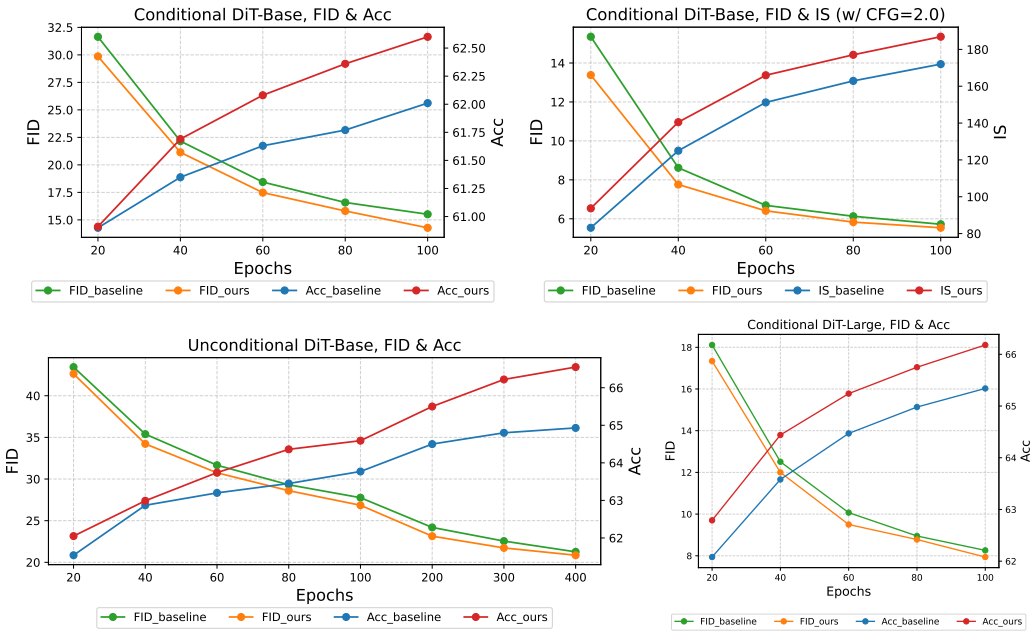


Figure 4: **Detailed performance evolution, showing scalability and consistency.** Across different settings and metrics, our method outperforms the respective baselines throughout the training.

Finally, we compare these results with other SSL and diffusion-based methods in Tab. 4. While self-conditioning models do not yet match leading contrastive methods like DINO (Caron et al., 2021), they outperform other generative learners such as l-DAE and MIM-based ViT-B/L, even with their longer training schedules. We hypothesize that a key factor limiting our recognition accuracy is the absence of *augmentation regularization within latent-space training* (at least cropping), a non-trivial challenge we defer to future work.

Table 4: **Performance comparison on ImageNet.**

Model	Backbone	Aug	CL	ImageNet-256x256		
				Ep.	FID↓	Accuracy↑
Flattened, clean VAE latent						40.04
RF	DiT-B			100	27.76	63.77
	+1.38M			100	26.85	64.59
RF	DiT-B			400	21.26	64.93
	+1.38M			400	20.85	66.55
RF (cond)	DiT-B			100	15.51	62.01
	+1.38M			100	14.28	62.60
RF (cond)	DiT-L			100	8.26	65.34
	+2.36M			100	7.94	66.18
l-DAE	DiT-L	crop		400	11.60	57.50
	DiT-L	crop		400	—	65.00
	DiT-Bx2	crop		400	—	60.30
CL	resnet50	✓	✓	100		66.50–69.30
	ViT-B	✓	✓	300		71.60–**78.20**
CL*	ViT-B	crop	✓	400		61.10–65.30
MIM	ViT-B	crop		400		61.40–62.90
	ViT-L	crop		200		62.20–65.80

*DINO with only flip+crop available (crop – multi-crop), reported by SODA.

Note that these DiT baselines are powered by the state-of-the-art Lightning-DiT (Yao et al., 2025), and operate at a high-performance frontier with extremely fast convergence. Against the backdrop of such already highly-optimized systems, the ability of our minimal, plug-and-play tweaks to yield further discernible dual improvements is particularly compelling and valuable.

How does self-conditioning work? Fig. 5 and Fig. 6 illuminate how it operates. Specially, we focus on how it reshapes the layer-wise discriminability: in baseline, while there's a shift of more discriminative features towards middle layers as training progresses (the model itself gradually learns how to adjust the landscape), the shift is slow and the variation is modest, in line with the distributed representation theory in Sec. 1. In contrast, we not only accelerate the convergence to middle layers, but also create a more pronounced performance hierarchy. This sharper focus suggests a more effective bottleneck. Intriguingly, while our designated layer is the 10th, peak accuracy emerges at an even earlier layer. We leave the deeper investigation to future work. Additionally, the observed reduction in denoising loss suggests that effective semantic guidance is indeed taking place.

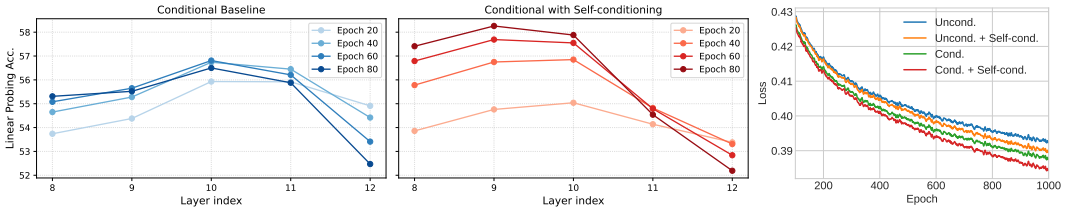


Figure 5: **Self-conditioning reshapes feature distribution.** While discriminative features gradually shift towards middle layers, they remain dispersed with modest differences. We accelerate this shift, forming a pronounced concentration, indicative of a condensed bottleneck.

Figure 6: **Self-conditioning facilitates the optimization** and narrows the loss gap between un- and class-cond.

Table 5: **Ablation studies** on CIFAR-10/100 for 1200 epochs. Our design choices are in gray.

(a) Self-conditioning in UNet.

Models, approaches	FID↓	Acc.↑
DDPM, baseline	3.02	94.02
DDPM, addition	2.80	94.18
DDPM, adaptive	**2.76**	**94.34**
EDM, baseline	2.25	94.83
EDM, addition	2.43	94.85
EDM, adaptive	**2.21**	**94.85**
RF, baseline	2.57	**93.97**
RF, addition	2.69	93.60
RF, adaptive	**2.42**	93.72

(b) MLP projection head.

MLP head	FID↓	Acc.↑
Original MoCo v3	5.87	69.01
Time-dependent	**5.88**	**69.50**

(c) Distillation timestep.

Target timestep	FID↓	Acc.↑
Minimal noise	5.81	68.76
Optimal for linear	**5.88**	**69.50**

(d) Contrastive loss weight.

γ	FID↓	Acc.↑
0.1	6.03	69.08
0.01	**5.88**	**69.50**
0.001	5.96	68.42

(e) Distillation target in UViT.

Feature to contrast	FID↓	Acc.↑
Pooling of tokens	8.51	73.15
`[CLS]`	**7.37**	**74.48**

5.3 Ablation studies

Below we show ablation experiments on design choices in self-conditioning and self-distillation. Tab. 5a: The **adaptive addition** of semantic features into the pathway is crucial, which consistently yields greater improvements, whereas a direct addition does not show much benefits. Tab. 5b: Using the original MoCo v3 projection head results in the same FID, but linear probing accuracy decreases by 0.5% compared to our **time-dependent** one. Tab. 5c: Using target features extracted with the **optimal timestep** that performs in linear probing, leads in accuracy by 0.7% when compared to SD-DiT's minimal-noise design (Zhu et al., 2024). Tab. 5d: Either excessive and insufficient **contrastive weight** leads to degradation in both metrics, suggesting that the contrastive method can contribute positively to dual aspects when appropriately tuned. Tab. 5e: We investigate two potential methods to extract features from UViT as **distillation target**: by average pooling (in line with linear probing) or by taking the `[CLS]` token. The `[CLS]` approach benefits both metrics, indicating that self-conditioning based on this token works well because it aggregates global semantics naturally.

Finally, we validate our training-loss-based bottleneck selection strategy, by evaluating four candidate layers in a conditional DiT-Base for self-conditioning on ImageNet. The layer (9th) yielding the lowest training loss after 20 epochs of training, will also achieve the optimal generation FID.

Bottleneck	Loss @ 20 ep.	FID @ 100 ep.
(Baseline)	0.4141	15.51
Layer 8	0.4135	14.79
Layer 9	**0.4133**	**14.28**
Layer 10	0.4136	14.52
Layer 11	0.4140	14.73

6 Conclusion

We show that established diffusion architectures can be enhanced, by conditioning the decoding process on features learned by themselves. The idea can be surprisingly simple to implement, yet concurrently improves both generation and representation quality at almost no cost. It also facilitates the integration of discriminative techniques for further recognition gains. We hope this straightforward principle inspires continued progress towards unified diffusion-based generation and understanding.

References

Fan Bao, Chongxuan Li, Jiacheng Sun, and Jun Zhu. Why are conditional generative models better than unconditional ones? *arXiv:2212.00362*, 2022.

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 *CVPR*, 2023a.

Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *ICML*, 2023b.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv:2106.08254*, 2021.

Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *ICLR*, 2022.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2018.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020b.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021.

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

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 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. *arXiv:2403.03206*, 2024.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

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 *NeurIPS*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Vincent Tao Hu, Yunlu Chen, Mathilde Caron, Yuki M Asano, Cees GM Snoek, and Bjorn Ommer. Guided diffusion from self-supervised diffusion features. *arXiv:2312.08825*, 2023a.

Vincent Tao Hu, David W Zhang, Yuki M Asano, Gertjan J Burghouts, and Cees GM Snoek. Self-guided diffusion models. In *CVPR*, 2023b.

Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *TPAMI*, 2023.

Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. *arXiv:2311.17901*, 2023.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, 2024.

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 *ICCV*, 2023a.

Tianhong Li, Dina Katabi, and Kaiming He. Self-conditioned image generation via generating representations. *arXiv:2312.03701*, 2023b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Benyuan Meng, Qianqian Xu, Zitai Wang, Xiaochun Cao, and Qingming Huang. Not all diffusion model activations have been evaluated as discriminative features. In *NeurIPS*, 2024.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021a.

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

Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. In *ICLR*, 2023.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *NeurIPS*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *ICCV*, 2023.

Guangkai Xu, Yongtao Ge, Mingyu Liu, Chengxiang Fan, Kangyang Xie, Zhiyue Zhao, Hao Chen, and Chunhua Shen. Diffusion models trained with large data are transferable visual models. *arXiv:2403.06090*, 2024.

Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2023.

Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *ICCV*, 2023.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025.

Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *ICCV*, 2023.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *ICLR*, 2022.

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:2403.17004*, 2024.

A Detailed description of objectives

DDPM (Ho et al., 2020) operates over a large number of discrete timesteps ($T = 1000$), based on a variance preserving schedule, *i.e.*, $x_t = \alpha_t x_0 + \sqrt{1-\alpha_t^2}\epsilon$. The network learns to predict the noise ϵ_θ, and is trained using the ϵ-prediction objective $\|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$. Euler's method is used for sampling, also known as DDIM (Song et al., 2021a).

EDM (Karras et al., 2022) is a continuous-time model based on the variance exploding schedule, *i.e.*, $x_t = x_0 + \sigma_t\epsilon$, where σ_t spans a continuous range of $[0.002, 80]$. The network predicts the denoised image D_θ, and is trained with the x_0-prediction objective $\|D_\theta(x_t, t) - x_0\|_2^2$. 2[nd] order Heun solver can be utilized for efficient sampling.

Table 6: **Details of diffusion model formulations.**

Pixel-space DDPM	
T	1000
training noise schedule	linear beta schedule $[10^{-4}, 0.02]$
training loss weighting	none
dropout rate	0.1
EMA decay rate	0.9999
ODE sampler	Euler (DDIM)
sampling NFE	100
Pixel-space RF	
training noise schedule	$t \in [0, 1]$
dropout rate	0.1
EMA decay rate	0.9999
ODE sampler	RK45
sampling NFE	140-160 adaptive

Pixel-space EDM	
training noise schedule	$P_{mean} = -1.2, P_{std} = 1.2$
training loss weighting	uncertainty (when using non-leaky)
dropout rate	0.13
EMA decay rate	0.9993
ODE sampler	2nd order Heun
sampling noise schedule	$\sigma_{min} = 0.002, \sigma_{max} = 80, \rho = 7$
sampling NFE	35 ($18 \times 2 - 1$)
Latent-space RF	
training noise schedule	$t \in [0, 1]$, lognorm sampling
EMA decay rate	0.9999
ODE sampler	Euler
sampling NFE	250
CFG parameters	scale=1.5, interval=$[0, 0.89]$
timestep shift	3.3

RF (Rectified Flow) (Liu et al., 2023) defines the forward process as a linear interpolation between data and noise, *i.e.*, $x_t = (1-t)x_0 + t\epsilon$, with t sampled from $[0, 1]$. The network directly estimates the velocity and is trained using a v-prediction objective $\|v_\theta(x_t, t) - (\epsilon - x_0)\|_2^2$. Euler's ODE solver is used for sampling.

MoCo v3 (Chen et al., 2021) is a contrastive self-distillation framework that learns representations by matching positive pairs. It uses an online model to compute the representation q_1 of an image, and a momentum-updated teacher for representation k_2 of another augmented view. Both online model and teacher consist of a backbone and a MLP projection head (Chen et al., 2020b), and the online encoder has an extra MLP prediction head (Grill et al., 2020). MoCo v3 employs a symmetrized contrastive loss $\mathcal{L}_{\text{MoCo}} = \mathcal{L}_{\text{InfoNCE}}(q_1, k_2) + \mathcal{L}_{\text{InfoNCE}}(q_2, k_1)$ to train the model.

B Implementation details

Diffusion models. Our pixel-space implementations are primarily based on the official codebases of DDAE[3] and UViT[4], including model and backbone definitions for DDPM, EDM, `ddpm`, `ddpmpp`, and UViT. The uncertainty loss weighting based on multi-task learning proposed in EDM2 (Karras et al., 2024) is also incorporated when non-leaky augmentation is applied to EDM, as it mitigates the slower convergence caused by augmentations. We also implement RF following core principles in Flow Matching (Lipman et al., 2023) and Rectified Flow (Liu et al., 2023). Our implementations of `ddpm` and UViT are equivalent to official versions, while `ddpmpp` is a simplified variant, adhering to main designs outlined in (Song et al., 2021b) but omitting certain details like skip connection rescaling, which don't find helpful. Our latent-space implementations are based on the official codebase of Lightning-DiT[5]. Pre-trained VA-VAE checkpoint is used, also downloaded from its official repository. We fix the "3-channel CFG" bug originated in DiT (Peebles & Xie, 2023) to enhance sample quality. More hyper-parameters are provided in Tab. 6 and Tab. 7.

Pre-training. On all pixel-space datasets, we train all models for 2000 epochs, saving checkpoints every 200 epochs for FID measurement. We observe that the model typically achieves lowest FID after 1400-2000 epochs. The training setups for DDPM and RF are identical, while EDM differs slightly in warmup, dropout, and EMA rate. These hyper-parameters are inherited and have not been heavily tuned. For UViT, the official implementation applies weight decay to all parameters. However, we find that excluding bias and positional embedding yields better results. We also apply weight decay to the `[CLS]` token, contrasting with

Table 7: **Details of backbone architectures.**

UNet-based architecture	**ddpm**	**ddpmpp**
base channels (multiplier)	128	128
channels per resolution	1-2-2-2	2-2-2
blocks per resolution	2	4
attention resolutions	$\{16\}$	$\{16\}$
attention heads	1	1
BigGAN block for up/down	no	yes
(param count)	35.7M	56.5M
ViT-based architecture	**UViT-S**	**DiT-B**
patch size	2	1
hidden size	512	768
ViT layers	13	12
attention heads	8	12
(param count)	44.3M	129.8M
Latent-space compressor	**VA-VAE**	
output latent size	16x16 (256/16)	
output dimension	32	

[3] https://github.com/FutureXiang/ddae

[4] https://github.com/baofff/U-ViT

[5] https://github.com/hustvl/LightningDiT

Table 8: **Time and layer for feature extraction.** Out denotes output layers in UNet and UViT.

Model	Backbone	*On CIFAR-10*		*On CIFAR-100*	
		Time	**Layer**	**Time**	**Layer**
DDPM	ddpm	$t = 11$	out_6/12	$t = 11$	out_6/12
EDM	ddpm	$\sigma_t = 0.06$*	out_7/12	$\sigma_t = 0.06$*	out_6/12
RF	ddpm	$t = 0.06$	out_7/12	$t = 0.06$	out_7/12
DDPM	ddpmpp	$t = 11$	out_7/15	$t = 11$	out_8/15
EDM	ddpmpp	$\sigma_t = 0.06$*	out_9/15	$\sigma_t = 0.06$*	out_8/15
RF	ddpmpp	$t = 0.06$	out_8/15	$t = 0.06$	out_8/15
DDPM	UViT-S	$t = 11$	out_2/6	$t = 11$	out_2/6

*Corresponds to $t = 4$ in 18 sampling steps.

Model	Backbone	*On Tiny-ImageNet and ImageNet100-64x64*	
		Time	**Layer**
EDM	ddpm	$\sigma_t = 0.2964$*	out_5/12
DDPM	UViT-S	$t = 66$	out_1/6

*Corresponds to $t = 6$ in 18 sampling steps.

Model	Backbone	*On IN100-256x256*		*On IN1k-256x256*	
		Time	**Layer**	**Time**	**Layer**
RF	DiT-B	$t = 0.25$	8/12	$t = 0.25$	9/12
(cond)	DiT-B	$t = 0.25$	8/12	$t = 0.25$	10/12

common practices. All these experiments are conducted on 4 NVIDIA 3080Ti or 4090 GPUs with automatic mixed precision enabled. On latent-space 256x256 datasets, we train DiT models for either 1400 epochs (unconditional, IN100), 1000 epochs (class-conditional, IN100), or 100 epochs (both, IN1k). These training durations are sufficient for conditional models, but unconditional ones may still benefit from longer training. Horizontal flip is applied as the only data augmentation method, unless specified otherwise (*i.e.*, previously using non-leaky on CIFAR). We cache two flipping versions of latents on IN100 or IN1k datasets during data pre-processing. All these experiments are conducted on 6 NVIDIA 4090 GPUs with automatic mixed precision.

Feature extraction. We determine the optimal timestep (noise scale) and layer index for feature extraction by grid searching over reasonable ranges, evaluated using linear probing on the validation set, following the practice in DDAE. The selected values are summarized in Tab. 8, and may differ from those of DDAE due to slight changes in linear probing settings. Notably, the optimal layer index is also utilized during the training of UNet and DiT models with self-conditioning. Similarly, the optimal noise scale is adopted for extracting the target feature in contrastive self-distillation.

Linear probing. We simplify the settings in DDAE by using identical training epochs and learning rates for all models. Additionally, we find that random cropping is unnecessary for linear probing, so we use only horizontal flipping as the augmentation method. The training and evaluation configurations are shown in Tab. 9. On CIFAR datasets, linear probing accuracy is reported as the highest among the checkpoints at 800, 1000, and 1200 epochs. On other datasets, we find that the linear probing accuracy does not saturated till the end of the training.

Table 9: **Details of pre-training and linear evaluations.** All our models (as well as other baselines in comparison) are trained and evaluated within same datasets, *without* transfer learning.

Pixel-space experiments: CIFAR, Tiny-IN, IN100-64x64		
	Pre-training	**Linear probing**
GPUs	4 * 3080Ti (CIFAR) or 4 * 4090 (others)	
batch size	512 (ddpm) or 256 (ddpmpp, uvit)	
optimizer	Adam (unet) or AdamW (uvit)	Adam
warmup epochs	13 (DDPM, RF) 200 (EDM)	—
epochs	2000	15 (CIFAR) 30 (others)
learning rate	4e-4	4e-3
lr schedule	constant	cosine
augmentations	flip (optional non-leaky)	flip

Latent-space experiments: IN100-256x256, IN1k-256x256		
	Pre-training	**Linear probing**
GPUs	6 * 4090	8 * 4090
batch size	1024	1440
optimizer	Adam	Adam
epochs	1400 (IN100) or 90 (IN1k)	30
learning rate	2e-4	2e-3
lr schedule	constant	cosine
augmentations	flip	flip

C METHODOLOGY COMPARISON

Our full approach with contrastive self-distillation culminates in a self-supervised model that learns semantically meaningful representations through **cross-view alignment** (discriminative, inherited from CL) and **intra-view reconstruction** (generative, inherent in diffusion denoising, analogous to MIM) Huang et al. (2023); Zhou et al. (2022). As for generative modeling, our design leads to a **self-supported semantic-conditional** framework that uses intermediate features as semantic cues

to guide generation, with the feature encoder absorbed into the diffusion network as its first few layers.

Relation to SODA. SODA Hudson et al. (2023) learns representations through cross-view reconstruction. Similar to our self-conditioning, it also employs a feature modulation mechanism to impose a tighter bottleneck between the encoder and decoder, thereby learning compact, linearly-separable features. However, SODA focuses on image-conditional tasks like novel view synthesis, so it uses a disentangled encoder separate from diffusion decoder. Additionally, SODA's features are learned through pure generative pre-training, without investigating the influence of contrastive methods.

Please note that a direct comparison with SODA is not so appropriate. First, since SODA *has not* released its official code, we made our best effort to re-implement one, and we found that it does not perform well on CIFAR and Tiny-ImageNet datasets. Second, SODA is optimized for representations and *cannot* function as a regular diffusion model in standard unconditional (or class-conditional) settings. Third, SODA may not achieve superior FID and Acc *simultaneously* with a same model, as it employs different augmentations for classification (stronger) and reconstruction (weaker).

Relation to "Guided Diffusion from Self-Supervised Diffusion Features". This study Hu et al. (2023a) also improves unconditional generation without relying on external encoders, by utilizing discriminative features within diffusion models. However, it only *uses* these features to generate pseudo-labels through Sinkhorn-Knopp cluster assignment, without *enhancing* them by comparing cluster assignments as in SwAV Caron et al. (2020). In contrast, our work presents a more fundamental combination of diffusion pre-training, self-condition guidance, and contrastive feature enhancement.

Relation to SD-DiT. SD-DiT Zhu et al. (2024) is a recent work that aims to accelerate the training convergence of DiT Peebles & Xie (2023) through self-distillation. It aligns features extracted from the visible patches of an image with those extracted from the entire image by an EMA teacher. While this joint optimization of generative and discriminative objectives is similar to our approach, it does not focus on enhancing representation quality. Furthermore, its key design, setting the distillation target to the minimal noise scale, differs from ours that using the linear probing timestep, proven more effective in ablation.

Relation to REPA. REPA Yu et al. (2025) is a concurrent work that accelerates DiT Peebles & Xie (2023) training through representation enhancement. It argues that aligning the intermediate features in diffusion models with powerful representations can significantly ease training. This "representation-for-better-generation" idea is similar to ours, but it uses an external large-scale pre-trained encoder as the teacher, rather than leveraging the diffusion models themselves. Additionally, REPA employs MLP heads and similarity functions, such as cross-entropy, to align features, similar to our MoCo v3-based method. However, it does not introduce two views for contrastive learning, as it is fundamentally a knowledge distillation process rather than a self-supervised one.

D Qualitative results

To visualize the generation quality, we present randomly generated samples in Fig. 7 (CIFAR-100) and Fig. 8 (ImageNet1k-256x256) using the same set of initial noise inputs.

E Additional information

E.1 Limitations and future research directions

Although we conduct extensive experiments to prove our claim that it is possible to enhance both generative and discriminative performance simultaneously through our approaches, our evaluations are limited to datasets up to the ImageNet1k-256x256 scale and models up to the DiT-Base scale. We did not investigate larger datasets with higher resolutions or larger models, such as DiT-XL, due to high computational costs inherently for generative models.

Figure 7: **Generate samples on CIFAR-100 using UViT-S.** Our proposed methods gradually improve the overall structure, semantics, and details (*e.g.*, see the motorbike, chair and bugs).

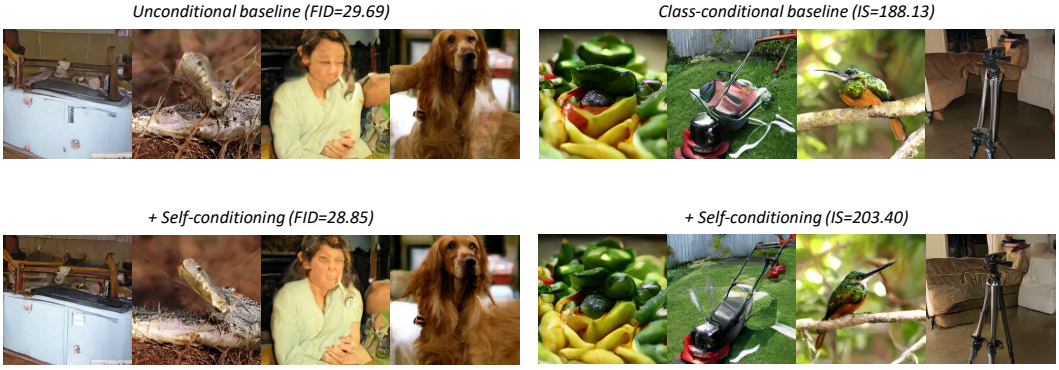


Figure 8: **Generate samples on ImageNet1k-256x256 using DiT-B.** Our method also improves the overall structure and details in latent-space unconditional (*e.g.*, see the human face and the dog) and class-conditional generation (*e.g.*, see the lawn mower and the tripod).

Moreover, some results indicate that our approaches may rely on careful choice of data augmentation strategies, which might require tuning when dataset changes. A key remaining challenge for integrating generative and discriminative learning, we believe, is the development of effective strategies to organize and identify multiple views of the same instance, meriting future research.

Finally, though our work provides initial insights into the formation, distribution, and enhancement of internal representations within diffusion models, the precise dynamics governing how these representations evolve throughout the training remain largely unexplored. Moreover, the consequences of potential misalignment between the layers designated for self-conditioning and those where optimal features naturally arise warrant further investigation.

E.2 Societal Impacts

Although we focus on exploring the frontiers in unconditional (or class-conditional) image generation and representation learning, on datasets that contain natural images (instead of human portraits or faces), our method still brings an improvement in the quality of generative models, which could be used to generate fake images for disinformation. However, the images generated by our models, restricted by the unconditional (or class-conditional) nature, are not as high-fidelity and photorealistic as those in the text-to-image synthesis. Therefore, the potential risk of negative impact is very low.