LEARNING REPRESENTATIONS TO GUIDE PIXEL DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review



Figure 1: **High-quality pixel-space generation with a plain Transformer.** RG guides generation with its learned representations, achieving strong performance in both class-conditional and text-to-image synthesis. We show samples at resolutions from 256×256 to 1024×1024.

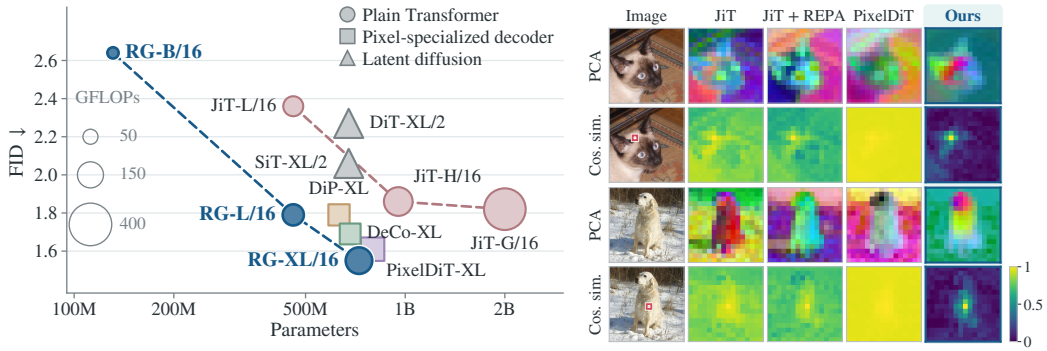
ABSTRACT

Visual representations are largely developed for understanding, yet what makes them useful for generation remains unclear. We study this question in pixel diffusion transformers and uncover two key insights. First, better representation quality does not necessarily translate into greater generative utility, as externally aligned semantics can be progressively lost during pixel synthesis. Second, architectural separation with specialized pixel decoders mainly redistributes semantic and spatial modeling, without necessarily yielding representations better suited to generation. These insights suggest that a useful generative representation must be both *properly formed* and *actively exploited*. Motivated by these insights, we introduce **Representation Guidance (RG)**, which directly forms and exploits internal representations while retaining a plain Transformer architecture. A low-dimensional bottleneck forces generation-relevant information into a compact representation, which is then explicitly used to guide decoder generation by modifying the flow-matching target according to its induced prediction offset. RG learns clearer semantic and spatial structure while scaling efficiently in both parameters and computation. On ImageNet 256×256, RG-L/16 with only 459M parameters outperforms substantially larger pixel diffusion transformers, while RG-XL/16 achieves an FID of 1.55. RG further reaches an FID of 1.74 at 512×512 and extends to text-to-image generation, achieving 0.89 on GenEval and 87.1 on DPG-Bench.

1 INTRODUCTION

Representation learning has been a central theme in computer vision (Chen et al., 2020; He et al., 2020; Caron et al., 2021; He et al., 2022). By transforming pixels into structured features, modern vision models have learned representations that transfer effectively across recognition tasks such as classification, detection, and segmentation (Oquab et al., 2023; Siméoni et al., 2025). Yet visual representations have been developed predominantly for understanding. Whether the same notion of a "good representation" applies to generation remains much less clear. Unlike recognition, which can discard much of the low-level image content once semantics are extracted, generation must organize global semantics and spatial structure while ultimately recovering every pixel. This raises a fundamental question: *what makes a representation useful for visual generation?*

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.



(a) (b)

Figure 2: **Our RG enables efficient scaling and better internal representations.** (a) RG scales consistently from B to XL. It achieves lower FID with fewer parameters and FLOPs than plain-Transformer JiT baselines, while also outperforming pixel-specialized decoders at lower complexity. (b) RG learns clearer spatial structure and more localized semantic correspondence.

Recent generative models have increasingly borrowed representations from pretrained vision encoders. One line of work improves the latent space in which generation operates, either by aligning autoencoder latents with pretrained features (Yao et al., 2025; Leng et al., 2025; Yue et al., 2026) or by directly using such features as generative latents (Zheng et al., 2026a; Shi et al., 2026; Singh et al., 2026). Another line aligns intermediate generative features with pretrained representations (Yu et al., 2024; Wu et al., 2025; Wang et al., 2025b; Singh et al., 2025). While these approaches show that semantic representations can facilitate generative learning, they largely focus on *what information a representation contains*. For generation, this is only part of the problem: a representation may encode rich semantic and spatial structure, yet provide limited benefit *if it is not properly formed for synthesis or if the generative process does not actually exploit it.*

We study this problem in pixel diffusion transformers (pDiTs) (Chen et al., 2025c; Yu et al., 2026), where representation learning and image synthesis take place end-to-end from raw pixels rather than through a representation space predefined by an external autoencoder (Rombach et al., 2022). More importantly, JiT (Li & He, 2026) shows that competitive pixel generation can be achieved with a plain Transformer, providing a simple setting for examining how internal representations emerge and interact with pixel synthesis. Our analysis of this setting reveals two key insights.

- **First, better representation quality does not necessarily translate into greater generative utility.** Aligning an intermediate JiT layer with DINOv2 (Oquab et al., 2023) substantially improves its semantic and spatial structure at the aligned layer. However, these gains rapidly diminish in subsequent layers as the network approaches pixel prediction. The same mismatch appears in generation: REPA (Yu et al., 2024) accelerates early training, but the baseline eventually overtakes it, while moving the alignment deeper further degrades the final FID. These results show that simply injecting stronger semantics is insufficient—a representation benefits generation only when its information is preserved and remains functionally relevant to subsequent synthesis.
- **Second, architectural separation does not necessarily improve representation for generation; it mainly redistributes semantic and spatial modeling across the network.** A natural response is to separate representation learning from pixel synthesis with an additional specialized pixel decoder (Yu et al., 2026; Ma et al., 2026a; Chen et al., 2026). However, the decoder itself reshapes the encoder representation: its spatial modeling capacity determines what the encoder must retain, producing different balances of global semantics and spatial structure without yielding a representation better suited to generation. These designs also incur substantial parameter and computation overhead. Thus, they mainly shift the modeling burden between encoder and decoder while departing from JiT's simple, scalable Transformer design.

Together, these findings suggest that a useful generative representation must be both *properly formed* and *actively exploited*. It should emerge through a compact interface that encourages the model to organize information into a concise, generation-relevant representation, while remaining tightly coupled to subsequent synthesis. Neither alone is sufficient: forcing the generator to rely on a poorly

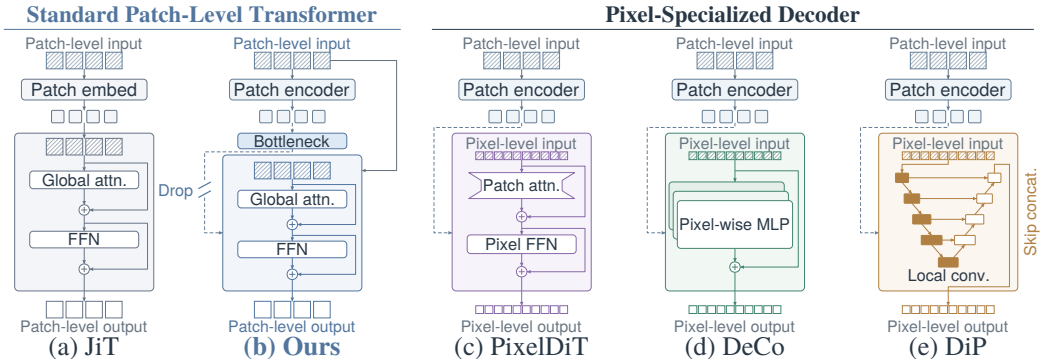


Figure 3: **Architecture comparison.** (a) JiT and (b) RG use patch-level input and standard Transformer blocks. JiT uses a single stack, while RG conditions its decoder on the bottleneck output, which is randomly dropped during training. (c)–(e) Pixel-specialized decoders take pixel-level input: patch attention with pixel-wise FFNs in PixelDiT, a pixel-wise MLP in DeCo, and a convolutional network with skip connections in DiP. Input tokens are noisy, output tokens are clean, and rounded squares denote the tokens output by the patch embedding or patch encoder.

formed representation can amplify unhelpful low-level information, whereas a well-structured representation provides little benefit if it can be ignored.

Based on these insights, we introduce **Representation Guidance (RG)**. RG preserves a plain Transformer architecture and operates on the representation interface rather than introducing a specialized pixel decoder. First, a low-dimensional bottleneck between the encoder and decoder forms a compact implicit latent space, encouraging generation-relevant information to be organized into a concise internal representation. Second, RG explicitly strengthens the generative utility of this representation. We compare decoder predictions with and without the representation and use the resulting representation-induced velocity offset to modify the flow-matching target (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023). This increases the model's dependence on the representation, while gradients through it simultaneously shape the encoder toward information useful for synthesis. The bottleneck and guidance thus play complementary roles: the former governs *how the representation is formed*, while the latter ensures that *generation actually uses it*. Pretrained representation alignment thus provides a semantic scaffold, while the representation consumed by the decoder is ultimately learned end-to-end for generation.

Figure 2 summarizes the consequences of this representation-centered design. Without introducing specialized pixel modules, RG achieves a substantially stronger parameter–quality and compute–quality trade-off, scales consistently with model size, and develops markedly clearer semantic and spatial organization in its internal representations. On ImageNet (Deng et al., 2009) 256×256, RG-XL/16 achieves an FID of 1.55, while RG-L/16 reaches 1.79 with only 459M parameters, outperforming substantially larger pixel-space models. RG further achieves an FID of 1.74 at 512×512 and extends naturally to text-to-image generation, reaching 0.89 on GenEval (Ghosh et al., 2023) and 87.1 on DPG-Bench (Hu et al., 2024). These results show that effective representation learning in pixel diffusion does not require increasingly elaborate pixel decoders or ever-stronger externally imposed features. Instead, a plain Transformer can learn representations that benefit generation when the representation is both properly formed and explicitly exploited.

2 What Makes Representations Useful in pDiTs?

We investigate how representations in pDiTs interact with generation. Our goal is not only to ask whether a representation becomes more semantic or spatially structured, but whether such improvements remain useful to the generative process. Unless otherwise specified, models in this section are base-sized and trained for 200 epochs on ImageNet 256×256 (Deng et al., 2009). We report FID-50K for generation quality, RMSC and LDS (Singh et al., 2025) for spatial structure, and linear probing accuracy (He et al., 2022) for global semantics. Appendix A.4 provides metric details.

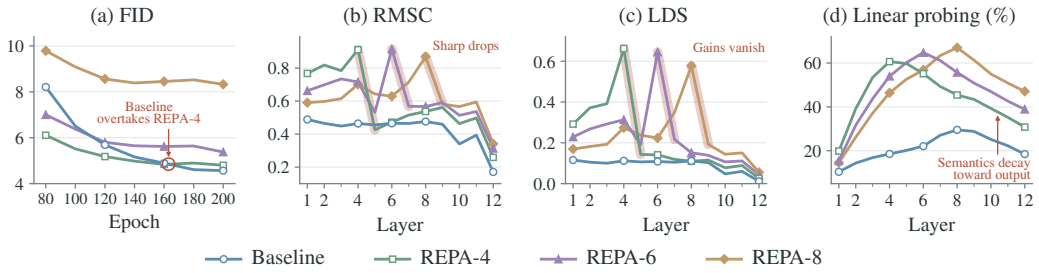


Figure 4: **Representation quality does not necessarily translate into generative utility.** REPA improves semantics and spatial structure at the aligned layer, but these gains quickly decay and do not improve final generation. REPA-k aligns layer k with DINOv2 features.

Does better representation quality translate into better generation? Pretrained representation alignment provides a direct way to strengthen the semantic and spatial structure of an intermediate representation. Rather than asking whether alignment can improve the aligned layer, we ask what happens afterwards: does this improvement survive subsequent pixel synthesis and benefit generation? We apply REPA (Yu et al., 2024) at different layers of JiT-B/16 and track both the representation across layers and generation quality during training.

- **Experimental Results.** Figure 4 reveals a clear mismatch. Aligning layer 4 substantially improves its RMSC, LDS, and linear probing accuracy, indicating stronger spatial structure and global semantics. However, these gains diminish immediately after the aligned layer as the network progresses toward pixel prediction. The same mismatch appears in generation: REPA-4 accelerates early training, but the baseline overtakes it after roughly 160 epochs and achieves a better final FID (4.56 vs. 4.80). Moreover, moving the alignment deeper further degrades generation, with FIDs of 5.38 and 8.33 for REPA-6 and REPA-8.
- **Analysis.** These results suggest that later layers progressively overwrite externally imposed semantics as they specialize toward dense pixel interactions and high-frequency synthesis. Alignment can therefore create a strong intermediate representation without making it useful to the final generative prediction; imposing it closer to the output can instead interfere more directly with pixel synthesis. **Our first insight is that better representation quality does not necessarily translate into greater generative utility.** A useful representation must remain functionally relevant throughout subsequent synthesis.

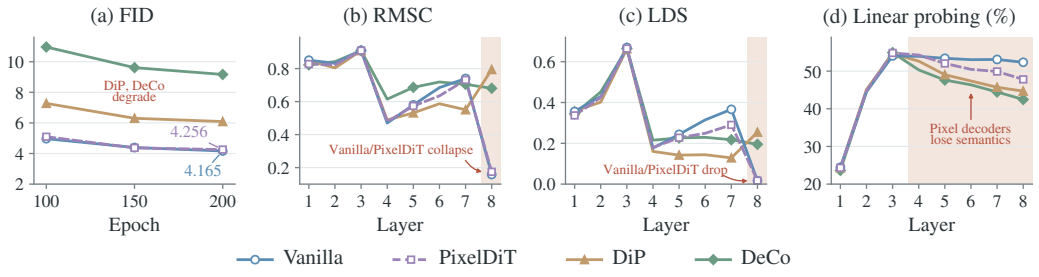


Figure 5: **Decoder design reshapes the encoder representation.** Different pixel decoders induce different balances of global semantics and spatial structure in encoder. Vanilla and PixelDiT retain stronger global semantics, whereas DiP and DeCo preserve more structures but weaker semantics.

Does architectural separation yield a better representation for generation? A natural response to the above mismatch is to separate high-level representation learning from low-level synthesis with an additional, separately designed pixel decoder. Recent pDiTs follow this route using pixel-wise MLPs (Ma et al., 2026a), convolutions (Chen et al., 2026), or pixel-level attention (Yu et al., 2026). But does such architectural separation actually produce a representation better suited to generation? We compare these designs under the same training budget and $\mathbf{x}$-prediction.

- **Experimental Results.** Figure 5 shows that pixel-specialized decoders substantially reshape the encoder representation. Their linear probing accuracy drops sharply after the aligned layer and falls below that of the vanilla Transformer decoder. At the last encoder layer, DeCo and

DiP retain stronger spatial variation but weaker global semantics, whereas vanilla and PixelDiT preserve stronger semantics but poor spatial structure. Consistently, Figure 2(b) shows less spatially structured representations in PixelDiT than in RG. Meanwhile, Table 1 shows that such specialization brings no consistent generation gain: PixelDiT reaches a similar FID to vanilla (4.26 vs. 4.17) with 55% more parameters and 67% more FLOPs, while DiP and DeCo degrade to FIDs of 6.09 and 9.17.

- **Analysis.** These differences reflect what each decoder can model. Vanilla and PixelDiT capture cross-patch spatial relations in the decoder, allowing the encoder to focus more on global semantics. In contrast, DeCo and DiP process patches largely independently and therefore rely on the encoder to retain spatial information. The decoder thus does not merely consume the representation; its design changes what information the encoder must retain. Together with Table 1, these results show that such redistribution does not consistently improve generation and often comes with higher parameter and computation costs. **Our second insight is that architectural separation mainly redistributes semantic and spatial modeling rather than yielding a representation better suited to generation.** Moreover, this specialization departs from JiT's simple, scalable Transformer design.

Table 1: **Pixel-specialized decoders offer limited gains under the same training budget.** PixelDiT achieves comparable FID with 55% more parameters and 67% more FLOPs, while DiP and DeCo perform worse. All models use REPA at encoder layer 3; for DiP, $4\downarrow +4\uparrow$ denotes four down/up-sampling stages.

Architecture	Encoder depth	Decoder depth	Params (M)	FLOPs (G)	FID ↓	IS ↑
Vanilla	8	4	131.32	28.42	4.165	298.62
PixelDiT	8	4	203.37	47.38	4.256	306.05
DiP	8	$4\downarrow +4\uparrow$	93.60	33.72	6.089	295.64
DeCo	8	4	99.52	25.80	9.167	249.23

Takeaways. Together, these findings motivate controlling both how the representation is formed during generation and how it is used by subsequent synthesis.

3 METHOD

3.1 REPRESENTATION GUIDANCE

RG addresses the two requirements in Section 2 with two components. For the first, an encoder maps the noisy image $\mathbf{z}_t$ to a compact representation through a bottleneck, and a decoder uses this representation as a condition to predict the clean image. For the second, RG guides training with the model's own learned representation. It shifts the training target along the velocity offset induced by this representation. Figure 3(b) illustrates the architecture.

Decoupled architecture. Following DDT (Wang et al., 2025a), the model consists of an encoder E_θ for representation learning and a decoder D_θ for pixel synthesis. All transformer blocks follow the JiT architecture (Li & He, 2026). The decoder has its own patch embedding and a few transformer blocks. It accesses the encoder only through the representation $\mathbf{r}$, which serves as a token-wise adaLN condition. Its $\mathbf{x}$-prediction gives the velocity $\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{r})$.

Implicit latent space. A linear bottleneck maps the encoder output tokens $\mathbf{h} \in \mathbb{R}^{N\times D}$ to a low-dimensional latent and back:

$$\mathbf{u} = \mathbf{h}\,\mathbf{W}_{\text{down}} \in \mathbb{R}^{N\times d}, \qquad \mathbf{r} = \mathbf{u}\,\mathbf{W}_{\text{up}} \in \mathbb{R}^{N\times D}, \qquad d \ll D. \tag{1}$$

We call the space of $\mathbf{u}$ an implicit latent space. It shares the compactness and spatial organization of a VAE latent space. However, it is computed from $\mathbf{z}_t$ at every step and optimized end-to-end for generation, while diffusion remains in pixel space.

Guidance objective. RG guides training with the velocity offset induced by the model's internal representation $\mathbf{r}$. To obtain a representation-free reference, we set $\mathbf{r}$ to zero with probability 0.1 during training. The decoder receives the class condition only through $\mathbf{r}$. With $\mathbf{r} = \mathbf{0}$, the same network therefore predicts an unconditional velocity $\mathbf{v}_\theta(\mathbf{z}_t, t, \varnothing)$. RG adds the offset between the two predictions to the flow matching target:

$$\mathbf{v}_{\text{RG}} = \mathbf{v} + (s-1)\ \text{sg}\left(\Delta\mathbf{v}_{\text{rep}}\right), \qquad \Delta\mathbf{v}_{\text{rep}} = \mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{r}) - \mathbf{v}_\theta(\mathbf{z}_t, t, \varnothing), \tag{2}$$

Table 2: **Class-conditional results on ImageNet** 256×256**.** RG-XL/16 reaches FID 1.55 at 600 epochs, narrowing the performance gap between pixel-space and latent-space diffusion models. The "Pre-training" columns list the external models. **Bold** = best in column.

Method	Epoch	Params	Pre-training	Generation@256			
				gFID↓	IS↑	Precision↑	Recall↑
Latent-space Generative Models							
DiT-XL (Peebles & Xie, 2023)	1400	675M	SD-VAE	2.27	278.2	**0.83**	0.57
SiT-XL (Ma et al., 2024)	1400	675M	SD-VAE	2.06	270.3	0.82	0.59
REPA (Yu et al., 2024)	800	675M	SD-VAE, DINOv2	1.42	305.7	0.80	0.65
Lightning DiT (Yao et al., 2025)	800	675M	VA-VAE	1.35	295.3	0.79	0.65
DDT-XL (Wang et al., 2025a)	400	675M	SD-VAE, DINOv2	1.26	**310.6**	0.79	0.65
RAE-XL (Zheng et al., 2026a)	800	839M	RAE	**1.13**	262.6	0.78	**0.67**
Pixel-space Generative Models							
RIN (Jabri et al., 2022)	480	410M	-	3.42	182.0	-	-
VDM++ (Kingma & Gao, 2023)	-	-	-	2.12	267.7	-	-
JetFormer (Tschannen et al., 2024)	-	2.8B	-	6.64	-	0.69	0.56
Simple Diffusion (Hoogeboom et al., 2023)	800	2.0B	-	2.44	256.3	-	-
FARMER (Zheng et al., 2026b)	320	1.9B	-	3.60	269.2	**0.81**	0.51
EPG-XXL/16 (Lei et al., 2026)	600	789M	-	1.81	294.6	0.80	0.61
PixelFlow-XL (Chen et al., 2025c)	320	677M	-	1.98	282.1	0.81	0.60
PixNerd-XL (Wang et al., 2026a)	320	700M	DINOv2	1.93	298.0	0.80	0.60
JiT-G/16 (Li & He, 2026)	600	2B	-	1.82	292.6	0.79	0.62
PixelGen-XL/16 (Ma et al., 2026b)	160	676M	VGG, DINOv2	1.83	293.6	0.79	0.63
PixelREPA-H/16 (Shin et al., 2026)	600	953M	DINOv2	1.81	317.2	-	-
DeCo-XL/16 (Ma et al., 2026a)	600	682M	DINOv2	1.69	304	0.79	0.63
DiP-XL/16 (Chen et al., 2026)	600	631M	DINOv2	1.79	281.9	0.80	0.63
PixelDiT-XL (Yu et al., 2026)	320	797M	DINOv2	1.61	292.7	0.78	0.64
PixelU-H/16 (Guo et al., 2026)	600	1168M	DINOv2	1.63	**305.9**	0.79	0.64
RG-L/16	600	459M	DINOv2	1.79	285.6	0.79	0.64
RG-XL/16	600	725M	DINOv2	**1.55**	293.8	0.80	**0.64**

where $s \geq 1$ is the RG scale and $\mathrm{sg}(\cdot)$ is the stop-gradient operator. The RG loss is $\mathcal{L}_{\mathrm{RG}} = \mathbb{E}_{\mathbf{x},\boldsymbol{\epsilon},t}\|\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{r}) - \mathbf{v}_{\mathrm{RG}}\|_2^2$, which reduces to $\mathcal{L}_{\mathrm{FM}}$ in Eq. (4) for $s = 1$. At convergence, RG amplifies the contribution of $\mathbf{r}$ to the predicted velocity by a factor of $1/(2-s)$ (Appendix A.3). The target distribution thus concentrates on a low-dimensional semantic manifold within the image data distribution (Chapelle et al., 2006; Li & He, 2026), as illustrated in Appendix Figure 7(b). The gradient of $\mathcal{L}_{\mathrm{RG}}$ also flows through $\mathbf{r}$ into the encoder, so the representation is optimized under the guided objective.

3.2 Training and Sampling

Guidance teacher. Following MG (Tang et al., 2025), we compute $\Delta\mathbf{v}_{\mathrm{rep}}$ with the EMA network, which provides a stable target. Its two predictions share $\mathbf{z}_t$, t, and $\mathbf{c}$, and differ only in $\mathbf{r}$. **Guidance interval and warm-up.** RG is applied for $t \in (t_{\min}, 1]$ and is enabled after a plain flow matching warm-up. The interval excludes highly noisy inputs, whose representations are unreliable (Section 4.3). Samples outside the interval or without the representation use the plain target $\mathbf{v}$. **Overall objective and cost.** We apply REPA (Yu et al., 2024) to an intermediate encoder layer. The overall loss is $\mathcal{L} = \mathcal{L}_{\mathrm{RG}} + \lambda\,\mathcal{L}_{\mathrm{REPA}}$. RG adds two gradient-free forward passes of the EMA network during training and no cost at inference.

4 Experiments

We evaluate RG on class-conditional ImageNet generation (Deng et al., 2009) and text-to-image generation. To study scalability, we instantiate RG with three model sizes: Base (B), Large (L), and Extra Large (XL). Table 9 summarizes their configurations. Unless otherwise specified, we use RG-B/16 trained for 200 epochs as the default model for all experiments on ImageNet.

4.1 Implementation Details

Class-conditional image generation. We follow the training setup of JiT (Li & He, 2026) on ImageNet-1K (Deng et al., 2009) at 256×256 and 512×512. We use AdamW with $\beta = (0.9, 0.95)$,

Table 3: **Class-conditional results on ImageNet 512×512.** Operating directly in pixel space, RG-XL/16 achieves a gFID of 1.74, outperforming the latent-space diffusion baselines.

Method	gFID↓	IS↑	Prec.↑	Rec.↑
Latent-space Generative Models				
SiT-XL	2.62	252.2	0.84	0.57
REPA	2.08	274.6	0.83	0.58
Pixel-space Generative Models				
JiT-H/32	1.94	309.1	-	-
DeCo-XL/16	2.22	290.0	0.80	0.60
PixelDiT-XL	1.81	278.6	0.78	0.67
PixelU-H/32	1.92	322.1	0.80	0.58
RG-XL/16	**1.74**	287.5	0.79	0.65

Table 4: **Text-to-image generation results at 512×512.** Reported parameter counts include both the generator and the text encoder.

Method	Params	GenEval↑	DPG↑
Latent-space Generative Models			
SD3 (Esser et al., 2024)	8B	0.68	84.1
FLUX.1-dev (Black Forest Labs, 2024)	12B	0.67	83.8
Pixel-space Generative Models			
PixelDiT-T2I (Yu et al., 2026)	1.3+2.6B	0.78	83.7
PixelGen-XXL/16 (Ma et al., 2026b)	1.1+1.7B	0.79	–
DeCo-XXL/16 (Ma et al., 2026a)	1.1+1.7B	0.86	81.4
MiniT2I-L/16 (Wang et al., 2026b)	912+341M	0.88	85.9
RG-T2I-L/16	912+341M	**0.89**	**87.1**

a constant learning rate of 2×10^{-4}, a batch size of 1024, EMA with a decay of 0.9999, and bfloat16 mixed precision on 8 A100 GPUs. Timesteps follow a logit-normal distribution (Esser et al., 2024) with $\mathrm{logit}(t) \sim \mathcal{N}(-0.8, 0.8^2)$. Unless otherwise stated, RG uses $s = 1.4$ and $t_{\min} = 0.3$ and is enabled after 20 epochs. Following ADM (Dhariwal & Nichol, 2021), we report FID, Inception Score, precision, and recall on 50K samples. We sample with a 50-step Heun solver (Heun, 1900). CFG is applied only within a timestep interval (Kynkäänniemi et al., 2024).

Text-to-image generation. RG-T2I-L/16 replaces the JiT blocks of RG with the MM-JiT blocks of MiniT2I (Wang et al., 2026b), which condition on text tokens from a frozen FLAN-T5-Large encoder (Chung et al., 2022). Following MiniT2I, the model is pretrained on CC12M (Changpinyo et al., 2021) with LLaVA-NeXT captions (Caption Emporium, 2024) and then fine-tuned on a 120K mixture of BLIP3o-60K (Chen et al., 2025a), DALL-E 3 (OpenDatasets, 2023), and ShareGPT-4o-Image (Chen et al., 2025b), both at 512×512. We evaluate on GenEval (Ghosh et al., 2023) and DPG-Bench (Hu et al., 2024), which contain 553 and 1,065 prompts. We generate 4 images per prompt with 100 sampling steps and a CFG scale of 5.0.

4.2 Main Results

Class-conditional image generation on ImageNet. As shown in Table 2, RG-XL/16 achieves an FID of 1.55 on ImageNet 256×256 with 725M parameters, the lowest among pixel-space methods. It outperforms mainstream methods with pixel-specialized decoders (PixelDiT-XL, FID 1.61; DiP-XL/16, FID 1.79; DeCo-XL/16, FID 1.69). Notably, RG-L/16 with only 459M parameters (FID 1.79) outperforms JiT-G/16 (2B, FID 1.82) and PixelREPA-H/16 (953M, FID 1.81). At 512×512, we train RG-XL/16 with patch size 16, yielding a 32×32 token grid. It achieves an FID of 1.74 (Table 3), also the lowest among pixel-space methods, which indicates that RG remains effective at higher resolution. Appendix A.7 shows qualitative samples.

Text-to-image generation. Table 4 reports text-to-image results at 512×512. RG-T2I-L/16 achieves 0.89 on GenEval and 87.1 on DPG-Bench, outperforming both large-scale latent-space models, such as SD3 (8B) and FLUX.1-dev (12B), and pixel-space methods with larger text encoders. PixelDiT-T2I and DeCo-XXL/16 use text encoders with 2.6B and 1.7B parameters, while RG-T2I-L/16 uses a 341M text encoder. With the same model size and training recipe as MiniT2I-L/16, RG-T2I-L/16 improves GenEval by 0.01 and DPG-Bench by 1.2, which indicates that RG remains effective for text-to-image generation. Appendix A.7 provides qualitative samples.

4.3 Ablations

On ImageNet 256×256, we ablate the bottleneck width, the aligned encoder layer, the encoder–decoder split, the components of RG, the RG scale and interval, and the teacher encoder for align-

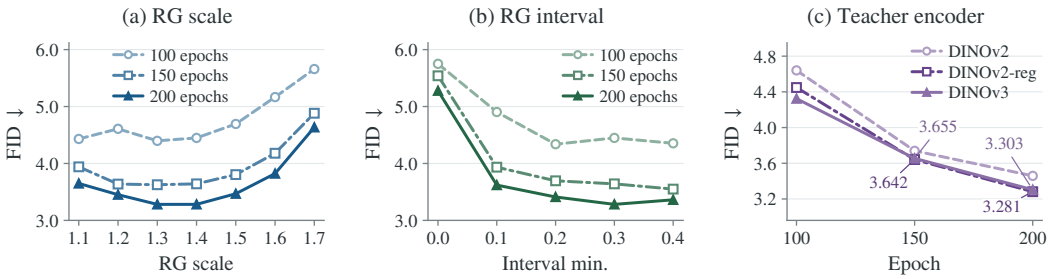


Figure 6: **Ablations on RG and the alignment target.** (a) A moderate RG scale ($s = 1.4$) yields the lowest FID at 200 epochs. (b) Excluding highly noisy inputs from guidance improves generation, with $t_{\min} = 0.3$ performing best. (c) DINOv2 with registers and DINOv3 yield comparable FIDs, both lower than standard DINOv2.

ment. Unless otherwise stated, all ablations use B/16 models trained for 200 epochs. For each model, we sweep the CFG scale with a fixed CFG interval of $(0.1, 1]$ and report the lowest FID.

Table 5: **Bottleneck width d.**

d	FID@200ep ↓	FID@600ep ↓
64	3.337	2.785
128	3.281	**2.644**
256	3.362	2.773
512	**3.181**	2.891

Table 6: **Alignment layer.**

REPA depth	FID ↓	IS ↑
3	**3.281**	264.1
6	3.327	**288.9**
8	8.973	164.4

Table 7: **Encoder/decoder blocks.**

E/D split	FID ↓	IS ↑
4 / 8	3.596	264.8
6 / 6	3.320	267.9
8 / 4	**3.281**	264.1
10 / 2	4.191	**270.1**

Bottleneck width. Table 5 varies the bottleneck width d. At 200 epochs, all widths yield comparable FIDs (3.18–3.36), with $d = 512$ performing best. At 600 epochs, however, $d = 128$ achieves the lowest FID (2.64), outperforming $d = 512$ (2.89). We therefore use $d = 128$ by default, combining a compact representation with better generation quality under longer training.

Alignment depth. Table 6 varies the encoder layer aligned by REPA. Aligning layer 3 or layer 6 yields similar FIDs of 3.28 and 3.33. Aligning the last encoder layer (layer 8) increases the FID to 8.97. In this setting, the encoder output $\mathbf{h}$ is directly aligned with the REPA target. This result indicates that pretrained visual representations such as DINOv2 do not suit pixel generation, which motivates optimizing the representation end-to-end for generation.

Encoder–decoder split. Table 7 varies the encoder and decoder depths under a fixed budget of 12 blocks. Increasing encoder depth reduces FID from 3.60 (4/8) to 3.28 (8/4), but a 2-block decoder (10/2) raises it to 4.19. A deeper encoder improves representations for generation, while sufficient decoder depth remains necessary for high-frequency synthesis. We use 8/4 by default.

Table 8: **Component ablation on B/16.**

REPA	BN	Guid.	Params (M)	FLOPs (G)	Epoch	FID↓	IS↑
Baseline: JiT-B/16			131.12	25.22	200	4.56	254.86
✗	✗	✓	131.32	28.42	200	8.08	230.81
✗	✓	✗	131.52	28.47	200	4.71	268.45
✗	✓	✓	131.52	28.47	200	8.99	226.20
✓	✗	✗	131.32	28.42	200	4.17	298.62
✓	✗	✓	131.32	28.42	200	7.85	220.40
✓	✓	✗	131.52	28.47	200	3.79	**302.43**
✓	✓	✓	131.52	28.47	200	3.28	264.12
					600	**2.64**	271.60

Components. Table 8 evaluates the three components on the decoupled B/16 model. REPA alone achieves an FID of 4.17, outperforming JiT-B (4.56) and JiT-B with REPA (4.80; Figure 4). Adding the bottleneck and then guidance reduces FID to 3.79 and 3.28, respectively; the full model reaches 2.64 at 600 epochs. Both REPA and the bottleneck are essential: without either, guidance raises FID to 7.85–8.99. REPA provides a semantic scaffold, while the bottleneck encourages compact representations and prevents collapse into low-level pixel features. The full model adds 0.4M parameters and 13% FLOPs over JiT-B.

RG scale and interval. Figure 6(a) varies the RG scale s. The lowest FID of 3.28 is achieved around $s = 1.4$. Figure 6(b) varies the lower bound $t_{\min}$ of the RG interval. The lowest FID is achieved at $t_{\min} = 0.3$. Applying RG at all noise levels ($t_{\min} = 0$) increases the FID to 5.28. Representations extracted from highly noisy inputs are unreliable and cannot effectively guide training.

Teacher encoder for alignment. Figure 6(c) compares three REPA targets: DINOv2-B, DINOv2-B with registers, and DINOv3-B/16 (Siméoni et al., 2025). At 200 epochs, they yield FIDs of 3.46, 3.28, and 3.30, respectively. All three pretrained representations thus provide an effective semantic scaffold for RG, and DINOv2 with registers and DINOv3 achieve slightly lower FIDs.

5 Related Work

Pixel-Space Generation. Most diffusion models generate images in a compressed latent space. Pixel-space generation directly models RGB images, avoiding the reconstruction bottleneck and preserving fine-grained details. Early work explored scalable pixel-space diffusion, flow-based generation, and autoregressive modeling (Jabri et al., 2022; Kingma & Gao, 2023; Hoogeboom et al., 2023; Tschannen et al., 2024). Recent transformer-based methods improve generation through patch-level modeling, pixel-level refinement, frequency decomposition, and encoder–decoder architectures (Li & He, 2026; Yu et al., 2026; Ma et al., 2026a; Chen et al., 2026). However, learning semantic representations and synthesizing dense pixels from the same raw input often slows convergence. Our method instead learns a compact internal representation to guide the generative objective during training, without requiring an external encoder at sampling time.

Representation Learning for Generation. Pretrained visual encoders provide semantic representations that transfer well to recognition tasks, but may discard the fine-grained information needed for image synthesis. Prior work uses these representations as diffusion latents (Yao et al., 2025; Zheng et al., 2026a) or aligns intermediate diffusion features with frozen encoder features (Yu et al., 2024; Shin et al., 2026). Dedicated encoder–decoder architectures also separate semantic modeling from pixel synthesis (Wang et al., 2025a; Ma et al., 2026a; Chen et al., 2026). While semantic representations can improve optimization, stronger alignment does not necessarily improve generation, and external representations may be mismatched with pixel-level synthesis. Our method learns representations end-to-end under the generative objective: a bottleneck encourages compact features, while representation guidance strengthens their role in generation, preserving global structure and fine-grained pixel information.

Guidance in Diffusion Models. Guidance is widely used to steer diffusion models toward conditional distributions. Classifier guidance uses auxiliary classifier gradients, while classifier-free guidance combines conditional and unconditional predictions during sampling (Dhariwal & Nichol, 2021; Ho & Salimans, 2022). Recent work leverages internal predictions: Internal Guidance introduces auxiliary intermediate-layer supervision and extrapolates between intermediate and final predictions during sampling (Zhou et al., 2026). RAEv2 shows that reparameterizing the REPA prediction head enables self-guidance without a separately trained guidance model or an additional classifier-free forward pass (Singh et al., 2026; Zu et al., 2026). While these methods primarily guide sampling, our representation guidance modifies the flow-matching target during training using the model's own representation. Gradients propagate through this representation, directly shaping both the learned vector field and the internal features used for generation.

6 Conclusion

We presented Representation Guidance (RG), which guides pixel diffusion transformers with their own learned representations. RG combines a low-dimensional bottleneck with a representation-guided flow-matching target to strengthen the generative utility of internal features while preserving a plain Transformer architecture. It achieves FIDs of 1.55 and 1.74 on ImageNet 256×256 and 512×512, respectively, and extends to text-to-image generation. Our findings highlight that useful generative representations must be both properly formed and actively exploited.

Limitations. Our experiments focus on image generation, and extending RG to video and other modalities is left for future work.

AI USE STATEMENT

We used large language model (LLM) assistants in two supporting roles. First, we used them to polish the language of author-written drafts, such as grammar, wording, and conciseness. Second, we used them for coding assistance on auxiliary scripts, including plotting, figure layout, and LaTeX formatting. We did not use generative AI tools to generate research ideas, design the method or experiments, or produce or analyze experimental results. All reported numbers come from our own training and evaluation runs, and all generated images shown in the paper are sampled from our RG models. The authors reviewed and edited all AI-assisted text, and checked all AI-assisted code and the resulting figures against the underlying data. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops a training method for pixel-space diffusion transformers. It does not involve human subjects or the collection of new data. We use publicly available datasets under their respective licenses: ImageNet-1K (Deng et al., 2009) for class-conditional generation, and CC12M (Changpinyo et al., 2021) with LLaVA-NeXT captions (Caption Emporium, 2024), BLIP3o-60K (Chen et al., 2025a), DALL-E 3 (OpenDatasets, 2023), and ShareGPT-4o-Image (Chen et al., 2025b) for text-to-image generation. These web-crawled or model-synthesized datasets may contain social biases, copyrighted content, or personal information, which models trained on them may reproduce. Like other image generators, RG could be misused to create misleading or harmful content, such as deepfakes or disinformation. RG does not introduce new capabilities beyond those of existing text-to-image models, but its improved training efficiency may lower the cost of building such models. Any release of trained models should therefore be accompanied by appropriate safeguards, such as usage licenses, content filters, and watermarking of generated images.

REPRODUCIBILITY STATEMENT

We describe RG in Section 3.1 and its training and sampling procedure in Section 3.2. Appendix A.3 derives the solution of the RG objective. Section 4.1 specifies the datasets, optimization hyperparameters, sampling settings, and evaluation protocols for both class-conditional and text-to-image generation, and Appendix A.2 lists the configurations of all models. Section 2 describes the analysis setting, and Appendix A.4 details the computation of RMSC, LDS, and linear probing accuracy. All datasets and pretrained models used in this work, including DINOv2 (Oquab et al., 2023) and FLAN-T5-Large (Chung et al., 2022), are publicly available, and our implementation builds on the public JiT (Li & He, 2026) and MiniT2I (Wang et al., 2026b) codebases. We will release our code and pretrained checkpoints upon publication.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

Black Forest Labs. Announcing black forest labs. `https://blackforestlabs.ai/announcing-black-forest-labs/`, 2024.

Caption Emporium. conceptual-captions-cc12m-llavanext. `https://huggingface.co/datasets/CaptionEmporium/conceptual-captions-cc12m-llavanext`, 2024.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (eds.). *Semi-Supervised Learning*. MIT Press, Cambridge, MA, USA, 2006.

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

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation, 2025b. URL https://arxiv.org/abs/2506.18095.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025c.

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

Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36136–36146, 2026.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

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. IEEE, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 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.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52132–52152, 2023.

Zipeng Guo, Lichen Ma, Yu He, Xiaolong Fu, Jingling Fu, Junshi Huang, and Yan Li. Pixelu: A u-shaped transformer for efficient end-to-end pixel diffusion. *arXiv preprint arXiv:2606.27760*, 2026.

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

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

Karl Heun. Neue methoden zur approximativen integration der differentialgleichungen einer unabhängigen veränderlichen. *Zeitschrift für Mathematik und Physik*, 45:23–38, 1900.

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

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. PMLR, 2023.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jiachen Lei, Keli Liu, Julius Berner, Haiming Yu, Hongkai Zheng, Jiahong Wu, and Xiangxiang Chu. There is no vae: End-to-end pixel-space generative modeling via self-supervised pre-training. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=HbUoKPIZmp.

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

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

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

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

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.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43600–43610, 2026a.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Improving pixel diffusion with perceptual supervision. *arXiv preprint arXiv:2602.02493*, 2026b.

OpenDatasets. Dall·e 3 dataset (laion discord set). https://huggingface.co/datasets/OpenDatasets/dalle-3-dataset, 2023.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Minglei Shi, Haolin Wang, Wenzhao Zheng, Ziyang Yuan, Xiaoshi Wu, Xintao Wang, Pengfei Wan, Jie Zhou, and Jiwen Lu. Latent diffusion model without variational autoencoder. In *International Conference on Learning Representations*, volume 2026, pp. 154506–154537, 2026.

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

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

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? In *The Fourteenth International Conference on Learning Representations*, 2025.

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

Zhicong Tang, Jianmin Bao, Dong Chen, and Baining Guo. Diffusion models without classifier-free guidance. *arXiv preprint arXiv:2502.12154*, 2025.

Michael Tschannen, André Susano Pinto, and Alexander Kolesnikov. Jetformer: An autoregressive generative model of raw images and text. *arXiv preprint arXiv:2411.19722*, 2024.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025a.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. In *International Conference on Learning Representations*, volume 2026, pp. 43559–43580, 2026a.

Xianbang Wang, Hanhong Zhao, Yiyang Lu, Kangyang Zhou, Linrui Ma, and Kaiming He. Minit2i: A minimalist baseline for text-to-image generation, 2026b. URL `https://peppaking8.github.io/#/post/minit2i`.

Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, Kai Wang, and Yang You. REPA works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. In *Advances in Neural Information Processing Systems*, 2025b.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Ming-Ming Cheng, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In *Advances in Neural Information Processing Systems*, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15703–15712. IEEE, 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. *arXiv preprint arXiv:2410.06940*, 2024.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14273–14282, 2026.

Zhengrong Yue, Taihang Hu, Mengting Chen, Haiyu Zhang, Zihao Pan, Tao Liu, Zikang Wang, Jinsong Lan, Xiaoyong Zhu, Bo Zheng, et al. What matters for diffusion-friendly latent manifold? prior-aligned autoencoders for latent diffusion. *arXiv preprint arXiv:2605.07915*, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026a.

Guangting Zheng, Qinyu Zhao, Tao Yang, Fei Xiao, Zhijie Lin, Jie Wu, Jiajun Deng, Yanyong Zhang, and Rui Zhu. Farmer: Flow autoregressive transformer over pixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25730–25741, 2026b.

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

Wenqiang Zu, Shenghao Xie, Bo Lei, and Lei Ma. Training-free representation guidance for diffusion models with a representation alignment projector. *arXiv preprint arXiv:2601.22468*, 2026.

A Appendix

A.1 Preliminaries

We adopt pixel-space flow matching (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023). Given a clean image $\mathbf{x}$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the noisy sample and the target velocity at time $t \in [0, 1]$ are

$$\mathbf{z}_t = t\,\mathbf{x} + (1-t)\,\boldsymbol{\epsilon}, \qquad \mathbf{v} = \mathbf{x} - \boldsymbol{\epsilon}. \tag{3}$$

Following JiT (Li & He, 2026), the network predicts the clean image. Given a condition $\mathbf{c}$, such as a class label or a text prompt, it outputs an estimate $\mathbf{x}_\theta(\mathbf{z}_t, t, \mathbf{c})$. This estimate is converted into a velocity $\mathbf{v}_\theta = (\mathbf{x}_\theta - \mathbf{z}_t)/(1-t)$ and trained with

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\mathbf{x},\boldsymbol{\epsilon},t} \big\| \mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{c}) - \mathbf{v} \big\|_2^2. \tag{4}$$

This objective amounts to an $\mathbf{x}$-space loss reweighted by $1/(1-t)^2$, where the denominator is clipped for numerical stability and t is drawn from a logit-normal distribution (Esser et al., 2024).

At inference, we apply classifier-free guidance (CFG) (Ho & Salimans, 2022). By setting $\mathbf{c}$ to a null token $\varnothing$ with probability 0.1 during training, a single network models both conditional and unconditional velocities, which are combined at sampling as

$$\mathbf{v}_\theta^w = \mathbf{v}_\theta(\mathbf{z}_t, t, \varnothing) + w\,\big(\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{c}) - \mathbf{v}_\theta(\mathbf{z}_t, t, \varnothing)\big), \tag{5}$$

where w is the guidance scale. This corresponds to sampling from a sharpened distribution $p(\mathbf{z}_t)\,p(\mathbf{c} \mid \mathbf{z}_t)^w$ at inference (Figure 7(a)). Unlike CFG, RG reshapes the target distribution during training by incorporating representation guidance into the flow matching target (Figure 7(b)).

A.2 Model Configurations

Table 9 summarizes the configurations of all RG models.

Table 9: **Model configurations.** D is the hidden dimension and d is the bottleneck width.

Model	Encoder depth	Decoder depth	D	d	Heads	Patch size	Params (M)
RG-B/16	8	4	768	128	12	16	131.5
RG-L/16	16	8	1024	128	16	16	459.4
RG-XL/16	20	10	1152	128	16	16	724.9

A.3 Analysis of Representation Guidance

This section derives the solution of the RG objective in Eq. (2) and discusses its relation to CFG and autoguidance.

Training objective. The RG loss replaces the flow matching target with $\mathbf{v}_{\text{RG}}$:

$$\mathcal{L}_{\text{RG}} = \mathbb{E}_{\mathbf{x},\boldsymbol{\epsilon},t} \big\| \mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{r}) - \mathbf{v}_{\text{RG}} \big\|_2^2. \tag{6}$$

An alternative is to extrapolate along $\Delta\mathbf{v}_{\text{rep}}$ at sampling time, as CFG does in Eq. (5). This doubles the decoder cost at inference and does not change the learned representation. RG instead moves the guidance into the training target.

Solution of the objective. Let $\bar{\mathbf{v}}_{\mathbf{r}} = \mathbb{E}[\mathbf{v} \mid \mathbf{z}_t, \mathbf{r}]$ and $\bar{\mathbf{v}}_\varnothing = \mathbb{E}[\mathbf{v} \mid \mathbf{z}_t]$. Samples with a dropped representation use the plain target, so the representation-free branch converges to $\bar{\mathbf{v}}_\varnothing$. Assume that the EMA teacher matches the online network at convergence. The minimizer of Eq. (6) then satisfies $\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{r}) = \bar{\mathbf{v}}_{\mathbf{r}} + (s-1)\big(\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{r}) - \bar{\mathbf{v}}_\varnothing\big)$, which gives

$$\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{r}) = \bar{\mathbf{v}}_\varnothing + \frac{1}{2-s}\big(\bar{\mathbf{v}}_{\mathbf{r}} - \bar{\mathbf{v}}_\varnothing\big), \qquad 1 \le s < 2. \tag{7}$$

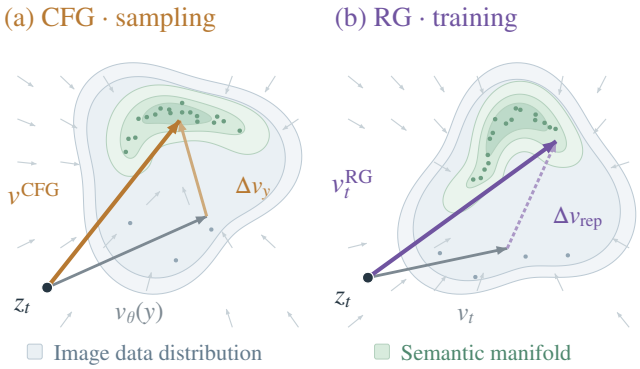


Figure 7: **CFG and RG.** (a) CFG guides sampling along $\Delta \mathbf{v}_y$. (b) RG guides training along $\Delta \mathbf{v}_{\text{rep}}$.

RG therefore trains the network to output a CFG-style velocity with an effective scale $w_{\text{rep}} = 1/(2-s)$, where the representation acts as the condition. For $s = 1.4$, $w_{\text{rep}} \approx 1.67$. By analogy with CFG, this informally corresponds to the sharpened distribution $p(\mathbf{z}_t)\, p(\mathbf{r} \mid \mathbf{z}_t)^{w_{\text{rep}}}$, which concentrates on samples consistent with the learned representation. Since part of the guidance is built into the model, a larger s reduces the CFG scale needed at sampling time (Section 4.3).

Relation to autoguidance. Since $\mathbf{r}$ is computed from $\mathbf{z}_t$, $\Delta \mathbf{v}_{\text{rep}}$ can also be viewed as the difference between the full model and its representation-free sub-network. This view relates RG to guiding a model with a weaker version of itself (Karras et al., 2024). Here, the weaker model is the decoder without access to the encoder.

Effect on the representation. CFG and MG use a fixed condition, whereas the condition of RG is learned. To fit the shifted target, the representation-conditioned branch must deviate further from the representation-free branch. This requires $\mathbf{r}$ to carry more information useful for denoising. The gradient of $\mathcal{L}_{\text{RG}}$ flows through $\mathbf{r}$ into the bottleneck and the encoder. RG therefore increases the dependence of the decoder on the representation and shapes the representation for generation.

A.4 Representation Metrics

All representation metrics are computed with the EMA weights at epoch 200. For each model, we extract the output tokens of every transformer block and exclude the in-context class tokens. For decoupled models, layers 1–8 refer to the encoder blocks and layers 9–12 refer to the decoder blocks. Each block output contains $N = 256$ patch tokens of dimension $D = 768$.

RMSC. We follow Singh et al. (2025). Let $\mathbf{h}_1, \ldots, \mathbf{h}_N$ be the patch tokens of one image, and let $\hat{\mathbf{h}}_i = \mathbf{h}_i / \|\mathbf{h}_i\|_2$. RMSC is defined as

$$\text{RMSC} = \Big(\frac{1}{N} \sum_{i=1}^{N} \big\| \hat{\mathbf{h}}_i - \bar{\mathbf{h}} \big\|_2^2 \Big)^{1/2}, \qquad \bar{\mathbf{h}} = \frac{1}{N} \sum_{i=1}^{N} \hat{\mathbf{h}}_i. \tag{8}$$

A higher RMSC means that the tokens of different spatial positions are more distinct. We compute RMSC on the 50K ImageNet validation images and average it over images.

LDS. Local-vs-distant similarity (LDS) (Singh et al., 2025) measures the contrast between nearby and distant patch tokens. Using the normalized tokens defined above, we compute

$$\text{LDS} = \frac{1}{|\mathcal{N}|} \sum_{(i,j) \in \mathcal{N}} \hat{\mathbf{h}}_i^\top \hat{\mathbf{h}}_j - \frac{1}{|\mathcal{F}|} \sum_{(i,j) \in \mathcal{F}} \hat{\mathbf{h}}_i^\top \hat{\mathbf{h}}_j. \tag{9}$$

On the 16×16 patch grid, $\mathcal{N}$ contains horizontal and vertical neighbors, and $\mathcal{F}$ contains pairs with Manhattan distance at least 6. A higher LDS indicates greater local similarity relative to distant positions. We compute LDS per image and average it over the 50K validation images.

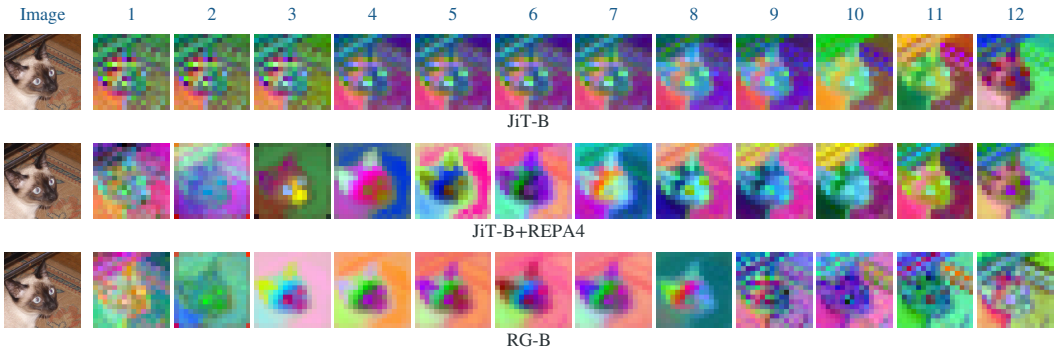


Figure 8: **PCA visualization of layer-wise features.** Features are visualized at $t = 0.5$. While REPA-induced semantic structure in JiT-B degrades in later layers, RG-B maintains clear semantic contours in encoder layers 3–8, with high-frequency patterns concentrated in decoder layers 9–12, indicating functional specialization.

For both RMSC and LDS, the input is the noisy image $\mathbf{z}_t = t\,\mathbf{x} + (1-t)\,\boldsymbol{\epsilon}$ with the ground-truth class label, and the representation is always kept. We compute the metrics at $t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and report their average. All models share the same images and noise.

Linear probing. We feed clean images ($t = 1$) with the null class label to the model, and average the patch tokens of each block to obtain image-level features. For each layer, we train a linear classifier on the features of the ImageNet training set, preceded by a BatchNorm layer without affine parameters. We use Adam with a learning rate of 10^{-3}, cosine decay, a batch size of 16384, and 90 epochs. We report the top-1 accuracy on the ImageNet validation set.

A.5 Representation Analysis

To understand how RG changes the internal representation, we analyze B/16 models at 200 epochs with PCA visualization and the representation metrics of Section 2.

Feature visualization. Figure 8 visualizes layer-wise features via PCA. JiT-B exhibits high-frequency patterns across layers, with unclear semantic contours and noisy within-object features. REPA clarifies object layout near the aligned layer 4, but later layers gradually revert to high-frequency patterns. In contrast, RG maintains clear semantic contours and foreground–background separation in encoder layers 3–8, while high-frequency patterns dominate decoder layers 9–12, indicating functional specialization.

Effect of each component. Figure 9 shows that the bottleneck substantially improves spatial structure at encoder layer 8 (higher RMSC and LDS), while guidance further strengthens functional specialization and global semantics. We conjecture that amplifying the representation-induced offset increases decoder reliance on the representation, encouraging the encoder to learn more semantic features.

A.6 Additional Results

CFG sweeps. Figure 10 shows the FID of B/16 models at epoch 200 under different CFG scales. For each RG setting, we report the FID over the evaluated range of CFG scales. The CFG interval is $(0.1, 1]$ in all cases. A larger RG scale shifts the optimal CFG scale to smaller values, which is consistent with Eq. (7). When RG is applied at all noise levels ($t_{\min} = 0$), the FID remains high for all CFG scales.

Convergence. Figure 11 compares the FID of RG-B, JiT-B, and JiT-B with REPA during training. REPA speeds up JiT-B in early training, but the advantage disappears after about 160 epochs. RG-B has a lower FID than both models from 100 epochs on, and the margin remains at 200 epochs.

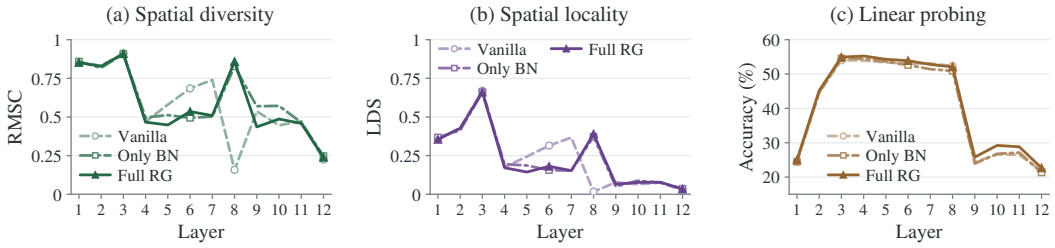


Figure 9: **Layer-wise representation metrics in RG-B.** The bottleneck (BN) improves spatial diversity and locality at the encoder output (layer 8), while representation guidance further strengthens global semantics, as measured by linear probing, and reinforces encoder–decoder specialization.

Layer-wise comparison with JiT. Figure 12 compares RG-B with JiT-B and JiT-B with REPA. In JiT-B, the linear probing accuracy increases slowly with depth and peaks at 29.5% at layer 8, and the LDS stays below 0.12 in all layers. REPA raises the accuracy of JiT-B to 60.7% at layer 4, but the accuracy then decreases to 30.8% at layer 12, and the LDS decreases from 0.662 to 0.025. The later layers of JiT convert the aligned features into pixels and gradually discard the semantic information. In RG-B, the linear probing accuracy stays between 52.1% and 55.2% from layer 3 to layer 8 and drops to between 22.7% and 29.2% in the decoder. The last encoder layer has an RMSC of 0.858 and an LDS of 0.390, compared with 0.476 and 0.111 in JiT-B. The encoder thus maintains a semantic and spatially diverse representation up to the interface, and the decoder performs pixel synthesis. JiT-B with REPA has a higher peak accuracy than RG-B, yet RG-B achieves a much lower FID (3.28 vs. 4.80). Generation quality therefore depends on whether the representation is maintained until the decoder uses it, rather than on the peak alignment quality.

Feature visualization. Figure 8 visualizes the features of all twelve layers of JiT-B, JiT-B with REPA, and RG-B. For each model and layer, we center the 256×768 token matrix, compute its first three principal components, and map them to RGB.

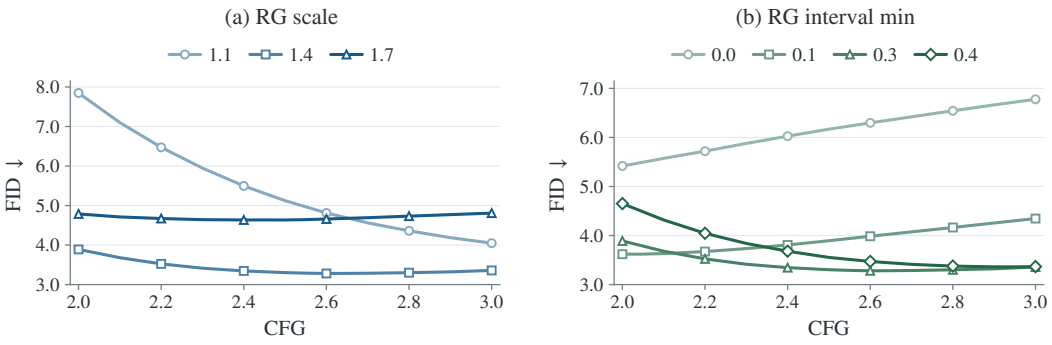


Figure 10: **FID versus CFG scale at epoch 200.** (a) Different RG scales s. (b) Different lower bounds $t_{\min}$ of the RG interval. Larger RG scales favor smaller CFG scales.

Our experiments focus on image generation, and extending RG to video and other modalities is left for future work.

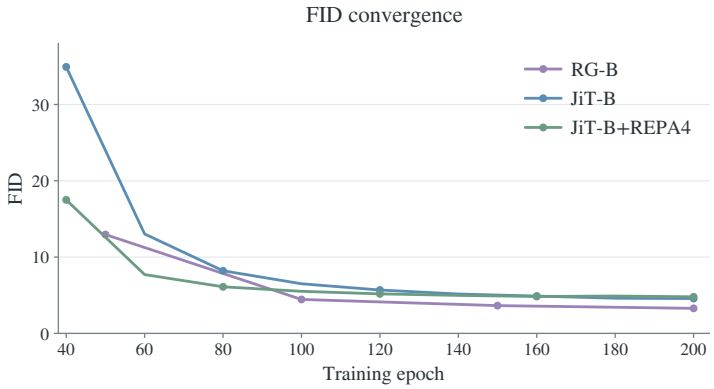


Figure 11: **FID during training.** RG-B reaches an FID of 4.45 at 100 epochs, lower than the FID of JiT-B at 200 epochs (4.56).

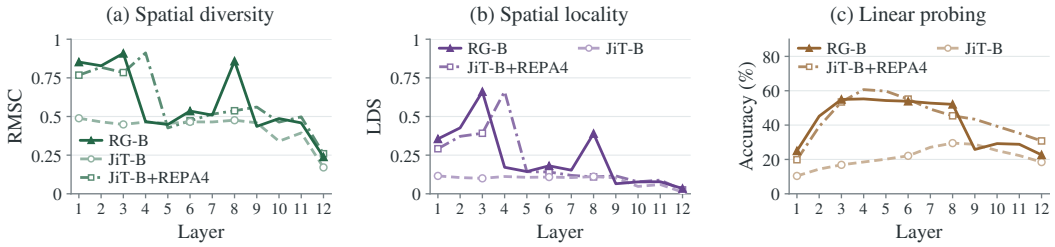


Figure 12: **Layer-wise representation of RG-B and JiT-B.** Layers 1–8 of RG-B form the encoder, and layers 9–12 form the decoder. RG-B keeps a semantic and spatially diverse representation up to the last encoder layer.

A.7 Qualitative Results



Figure 13: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 42.



Figure 14: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 88.



Figure 15: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 105.



Figure 16: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 263.



Figure 17: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 269.



Figure 18: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 282.

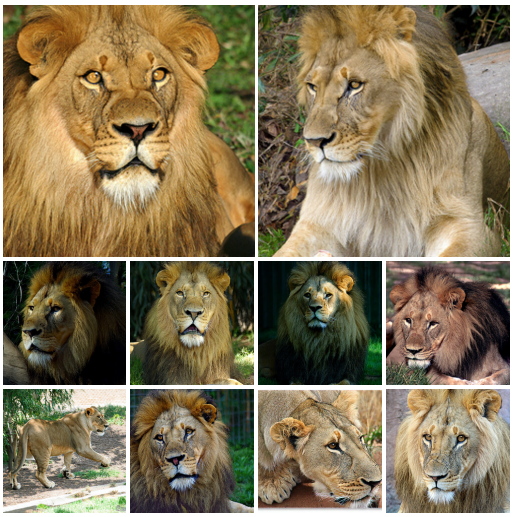


Figure 19: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 291.



Figure 20: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 483.



Figure 21: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 504.



Figure 22: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 511.



Figure 23: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 933.



Figure 24: Uncurated ImageNet 256 × 256 and 512 × 512 HF-XL samples. Class 949.