Semantics Lead the Way: Harmonizing Semantic and Texture Modeling with Asynchronous Latent Diffusion

Supplementary Material

Yueming Pan[1,2] Ruoyu Feng[3] Qi Dai[2] Yuqi Wang[3],
Wenfeng Lin[3] Mingyu Guo[3] Chong Luo[2] Nanning Zheng[1]

[1]State Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]Microsoft Research Asia [3]ByteDance

A. Additional Implementation Details

A.1. SemVAE Configuration

For semantic representation extraction, we employ DINOv2-B with registers [2, 19] on 256×256 images. The SemVAE architecture consists of 4 transformer blocks for both the encoder and decoder, with 29M parameters each (58M total). We train on ImageNet-1K [3] for 1M iterations with random cropping as data augmentation. Detailed hyperparameters are shown in Table 1.

A.2. Diffusion Model Configuration

We adopt LightningDiT [31] as our diffusion backbone, which is available in multiple scales (B/L/XL/XXL). For latent construction, SD-VAE [23, 31] (implemented in LightningDiT[1] [31]) encodes textures into 32 channels with 16× spatial compression, while SemVAE extracts 16-channel semantic representations. Their concatenation forms a unified 48-channel, 256-token latent for each 256×256 image.

Following the ADM [4] preprocessing pipeline, all images are center cropped and resized to 256×256 resolution. Training is conducted on ImageNet-1K [3] for 800 epochs with a batch size of 256, using AdamW optimizer with a learning rate of 1×10^{-4} and β values of (0.9, 0.999). We employ logit-normal timestep sampling following LightningDiT [31]. For sampling, the dopri5 solver [5] with adaptive step size is employed, with absolute and relative tolerances set to 10^{-6} and 10^{-3} respectively. Detailed hyperparameters across different model scales are shown in Table 2.

A.3. Dual Timestep Embedding

To support asynchronous denoising, SFD employs two independent timestep embedders corresponding to the semantic and texture timesteps t_s and t_z. Unlike LightningDiT [31], which uses a single MLP-based embedder of hidden dimension H, SFD constructs two smaller embedders whose hidden dimensions are reduced to $H/2$. Each embedder independently processes its respective timestep, the two embeddings are then concatenated along the channel dimension and injected into the backbone. This design allows the model to supply distinct timestep signals to the semantic and texture latents:

$$\mathbf{e} = [\,\tau_s(t_s),\ \tau_z(t_z)\,], \tag{1}$$

where $[\cdot,\cdot]$ denotes channel-wise concatenation, and $\tau_s(\cdot)$ and $\tau_z(\cdot)$ are the semantic and texture timestep embedders.

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

Table 1. **SemVAE training configuration.**

Component	Setting
Feature Extraction	
Feature Extractor	DINOv2-B-reg
Input Patch Size	256×256
Architecture	
Total Parameters	58M
Encoder Parameters	29M
Decoder Parameters	29M
Encoder Blocks	4
Decoder Blocks	4
Hidden Dimension	768
Attention Heads	6
Bottleneck Channels	16
KL Weight λ_{kl}	10^{-7}
Training	
Dataset	ImageNet-1K
Total Iterations	1,000,000
Batch Size	64
Data Augmentation	Random cropping
Optimization	
Optimizer	AdamW
Learning Rate	5×10^{-5}
(β_1, β_2)	(0.9, 0.999)
LR Schedule	
Warmup Steps	500
Constant Steps	800,000
Annealing	Cosine to 5×10^{-6}

Table 2. **Hyperparameter settings across different model scales.**

Backbone	LightningDiT-B	LightningDiT-L	LightningDiT-XL	LightningDiT-XXL
Architecture				
#Params	130M	458M	675M	1.0B
Input	$16 \times 16 \times 48$	$16 \times 16 \times 48$	$16 \times 16 \times 48$	$16 \times 16 \times 48$
Layers	12	24	28	32
Hidden dim.	768	1024	1152	1280
Num. heads	12	16	16	16
SFD settings				
β	2.0	2.0	2.0	2.0
Δt	0.3	0.3	0.3	0.3
REPA visual encoder	DINOv2-B-reg	DINOv2-B-reg	DINOv2-B-reg	DINOv2-B-reg
REPA weight λ	1.0	1.0	1.0	1.0
REPA alignment depth	2	2	2	2
REPA similarity function	cosine	cosine	cosine	cosine
Optimization				
Batch size	256	256	256	256
Optimizer	AdamW	AdamW	AdamW	AdamW
lr	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Sampling				
Sampler	dopri5	dopri5	dopri5	dopri5
Absolute tolerance	10^{-6}	10^{-6}	10^{-6}	10^{-6}
Relative tolerance	10^{-3}	10^{-3}	10^{-3}	10^{-3}

Table 3. **Configurations of degraded models used for guidance.**

Model	Epochs	Params	Degraded Model	Iterations	Guidance Scale
LightningDiT-XL	80	675M	LightningDiT-B	70K	1.6
LightningDiT-XL	800	675M	LightningDiT-B	70K	1.5
LightningDiT-XXL	80	1.0B	LightningDiT-B	60K	1.5
LightningDiT-XXL	800	1.0B	LightningDiT-B	120K	1.5

A.4. Evaluation Details

AutoGuidance. We employ AutoGuidance [12] as our primary guidance method. Unlike Classifier-Free Guidance (CFG) [9], which relies on an unconditional model, AutoGuidance guides the main diffusion model using a *weaker version* of itself—typically a model with smaller capacity or an earlier training snapshot. This self-guidance mechanism effectively suppresses out-of-manifold samples by aligning the denoising trajectory toward regions of higher data density, thereby improving image quality without sacrificing sample diversity. In practice, we use the degraded LightningDiT-B model as the guiding network. After searching, configurations of degraded models are illustrated in Tab. 3.

Class-balanced Sampling. RAE [33] shows that class-balanced sampling yields more reliable and lower FID estimates. To ensure fair comparison with prior work [17, 22, 25, 28, 33], we follow this protocol and adopt class-balanced sampling for FID-50K evaluation. Specifically, we generate 50 images per class (50,000 in total).

B. Additional Experimental Results

B.1. Complete Comparisons

Table 4 presents a system-level comparison of class-conditional generation on ImageNet 256×256. In the guidance setting, our SFD achieves state-of-the-art performance, surpassing existing methods in both FID and sFID. Notably, our SFD-XL (675M) outperforms the previous best model, RAE DiT$^{\text{DH}}$ (839M), with a lower FID (1.06 vs. 1.13), demonstrating superior generation quality with fewer parameters. Scaling up to SFD-XXL (1.0B) further pushes the performance boundary to a FID of 1.04. Notably, SFD achieves a superior sFID of 3.75, outperforming previous methods by a substantial margin. Since sFID serves as a metric for structural coherence and spatial alignment, this improvement validates the advantage of our explicit compression of semantic representations with spatial

layouts, which ensures robust global structure before texture refinement.

Regarding the unguided setting, SFD remains competitive but exhibits limitations in texture convergence. This is primarily attributed to the high complexity of the texture latents. Unlike methods such as ReDi [14] or REG [29] that utilize a standard f8d4 VAE, we employ the f16d32 variant (following LightningDiT), which results in a latent space with double the dimensionality. Consequently, modeling these high-dimensional texture latents is inherently more challenging and harder to converge.

B.2. Inference Strategies

Inference Steps. Table 5 illustrates FID scores without guidance of various sampling steps, showing that SFD maintains strong performance even with significantly fewer inference steps. While other baselines require 200–250 steps to approach their optimal FID, SFD already achieves a competitive score of 6.35 at only 100 steps, and further increasing the steps to 250 yields only marginal improvement (6.32). This observation suggests that the semantic-first design may facilitate more efficient sampling: by stabilizing global semantics early, the model requires fewer refinement steps to reach high-quality solutions.

Table 6 presents the FID scores with guidance across varying sampling steps. Notably, SFD (XL) achieves a superior FID of **1.045** at only 100 steps using the Euler sampler, surpassing the result yielded by the dopri5 sampler (1.064). Furthermore, in the few-step regime (25 steps), SFD (XL) maintains its advantage over SVG [24], recording an FID of 1.865 compared to 1.920 by SVG.

Class-balanced Sampling. To ensure a rigorous comparison, we re-evaluate prior state-of-the-art methods employing the same class-balanced protocol as discussed in RAE [33]. Specifically, results for SiT [18], REPA [32], and DDT [28] are adopted from RAE [33], while REPA-E [15] figures are sourced from its original publication. Additionally, we conduct independent evaluations for VA-VAE [31], ReDi [14], and REG [29]. The quantitative comparison results are presented in Table 7. As observed, our proposed SFD (XL) demonstrates consistent superiority across both protocols. Remarkably, whether using class-balanced or class-random sampling, SFD achieves the best performance in terms of FID and sFID metrics, surpassing all competing state-of-the-art methods.

B.3. Unconditional Generation

We further evaluate the proposed SFD on unconditional image generation on ImageNet 256×256. During both training and sampling, we set the class label to 1000 (the null label). As shown in Table 8, SFD demonstrates remarkable performance with high training efficiency. Even without AutoGuidance (AG) [12], SFD significantly surpasses ReDi (FID 25.10 → 10.24) after only 80 epochs and further improves to an FID of 8.46 after 200 epochs. With AG enabled, SFD achieves substantial gains, reaching FIDs of 3.77 and 2.90 at 80 and 200 epochs, respectively. We attribute these improvements to the asynchronous denoising mechanism of SFD, which becomes especially crucial in the unconditional setting. These results suggest that, without class labels as conditional guidance, smoother semantic representations are more easily modeled, thus providing accurate global structural cues for superior generation performance.

B.4. SFD for VA-VAE

Tab. 9 analyzes the impact of applying Semantic-First Diffusion (SFD) to VA-VAE [31]. For both VA-VAE and ReDi settings, the SFD implementations used for comparison are our reproduced versions. When equipped with SFD, VA-VAE improves performance from an FID of 4.52 to 4.14, indicating that SFD is also compatible with joint semantic-texture latent space. However, its overall performance still lags behind our SD-VAE-based SFD (FID 3.03). This is likely because VA-VAE's latent space inherently entangles semantics and textures, leaving limited flexibility for the asynchronous denoising mechanism to operate effectively. In contrast, disentangling semantic and texture representations (as done in SD-VAE) allows the semantic latents to stabilize early and provide clearer global guidance for texture refinement, ultimately yielding higher generative quality.

B.5. Computational Cost

We evaluate the computational overhead introduced by integrating SFD into LightningDiT-XL. SFD modifies the backbone in two ways. First, it augments the latent representation with a 16-channel semantic latent, which introduces a marginal increase in backbone FLOPs. Second, SFD replaces the single timestep embedder in LightningDiT with two independent embedders that operate on the semantic and texture timesteps t_s and t_z. As shown in equation 1, although two embedders are used, the total parameter count is actually smaller, since MLP parameters grow quadratically with hidden dimension. Consequently, two $(H/2)$-width MLPs contain only $0.5\times$ the parameters and FLOPs of a single H-width MLP.

Table 10 reports the computational cost comparison. SFD incurs only a negligible increase in FLOPs (less than 0.01%) while delivering a dramatic improvement in FID at 400K iterations. This indicates that SFD achieves an extremely favorable cost–performance tradeoff with virtually no additional computational burden.

Table 4. **System-level comparison of class-conditional generation on ImageNet 256×256.**

Method	Epochs	#Params	Generation@256 w/o guidance					Generation@256 w/ guidance				
			FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Autoregressive												
VAR [25]	350	2.0B	-	-	-	-	-	1.80	-	**365.4**	**0.83**	0.57
MAR [17]	800	943M	2.35	-	227.8	0.79	0.62	1.55	-	303.7	0.81	0.62
xAR [22]	800	1.1B	-	-	-	-	-	1.24	-	301.6	**0.83**	0.64
Pixel Diffusion												
ADM [4]	400	554M	10.94	6.02	101.0	0.69	0.63	3.94	6.14	215.8	**0.83**	0.53
RIN [11]	480	410M	3.42	-	182.0	-	-	-	-	-	-	-
PixelFlow [1]	320	677M	-	-	-	-	-	1.98	5.83	282.1	0.81	0.60
PixNerd [27]	160	700M	-	-	-	-	-	2.15	4.55	297.0	0.79	0.59
SiD2 [10]	1280	-	-	-	-	-	-	1.38	-	-	-	-
Latent Diffusion												
DiT [20]	1400	675M	9.62	6.85	121.5	0.67	0.67	2.27	4.60	278.2	**0.83**	0.57
MaskDiT [34]	1600	675M	5.69	10.34	177.9	0.74	0.60	2.28	5.67	276.6	0.80	0.61
SiT [18]	1400	675M	8.61	6.32	131.7	0.68	0.67	2.06	4.50	270.3	0.82	0.59
FasterDiT [30]	400	675M	7.91	5.45	131.3	0.67	0.69	2.03	4.63	264.0	0.81	0.60
MDT [6]	1300	675M	6.23	5.23	143.0	0.71	0.65	1.79	4.57	283.0	0.81	0.61
MDTv2 [7]	1080	675M	-		-	-	-	1.58	4.52	314.7	0.79	0.65
DDT [28]	400	675M	6.27	-	154.7	0.68	**0.69**	1.26	-	310.6	0.79	0.65
Leveraging Visual Representations												
VA-VAE [31]	800	675M	2.17	4.36	205.6	0.77	0.65	1.35	4.15	295.3	0.79	0.65
REPA [32]	800	675M	5.90	-	-	-	-	1.42	4.70	305.7	0.80	0.65
REPA-E [15]	800	675M	1.69	**4.17**	219.3	0.77	0.67	1.12	4.09	302.9	0.79	0.66
ReDi [14]	800	675M	3.30	4.80	188.9	0.74	0.68	1.61	4.66	295.1	0.78	0.64
REG [29]	800	677M	1.80	4.59	230.8	0.77	0.66	1.36	4.25	299.4	0.77	0.66
RAE [33] (DiT-XL)	800	676M	1.87	-	209.7	**0.80**	0.63	1.41	-	309.4	0.80	0.63
RAE [33] (DiT^{DH}-XL)	800	839M	**1.51**	-	**242.9**	0.79	0.63	1.13	-	262.6	0.78	**0.67**
SFD (XL)	80	675M	3.43	4.34	162.0	0.75	0.65	1.30	3.87	233.4	0.78	0.64
SFD (XL)	800	675M	2.54	4.38	191.7	0.75	0.67	1.06	3.89	267.0	0.78	**0.67**
SFD (XXL)	80	1.0B	2.84	4.25	172.6	0.75	0.65	1.19	4.00	240.4	0.78	0.65
SFD (XXL)	800	1.0B	2.38	4.37	197.9	0.75	0.67	**1.04**	**3.75**	264.2	0.78	0.66

Table 5. **FID (↓) comparison across inference steps.** All models are trained for 400K iterations and evaluated using the Euler sampler without guidance. Reported values are FID-10K scores computed at different inference step counts.

Method	Inference steps					
	250	200	150	100	80	60
LightningDiT	12.50	12.58	12.67	12.91	13.03	13.40
LightningDiT+REPA	10.00	10.10	10.23	10.50	10.67	10.94
LightningDiT+VA-VAE	7.66	7.66	7.68	7.70	7.76	7.83
LightningDiT+ReDi	8.58	8.63	8.72	8.86	9.02	9.32
LightningDiT+SFD (Ours)	6.32	6.26	6.41	6.35	6.81	6.77

C. Additional Ablation Studies

C.1. Semantic VAE Design

Our Semantic VAE (SemVAE) compresses pretrained vision foundation model features into compact semantic representations. To investigate its design choices, we conduct a series of ablation studies on four key aspects: the choice of pretrained vision encoder, model scaling within the encoder family, the number of output channels representing semantic capacity, and the choice of reconstruction objective.

Table 6. **FID (↓) comparison across inference steps** for SFD (XL) and SFD (XXL) models at 4M training iterations with guidance. Reported values are FID-50K scores.

Method	dopri5	250	200	150	100	80	60	50	40	30	25
SVG [24]	-	-	-	-	-	-	-	-	-	-	1.920
SFD (XL)	1.064	1.051	1.050	1.048	1.045	1.086	1.102	1.206	1.510	1.447	1.865
SFD (XXL)	1.040	1.035	1.040	1.041	1.058	1.080	1.106	1.190	1.429	1.456	1.844

Table 7. **Comparison of class-random sampling and class-balanced sampling.**

Method	Random sampling					Balanced sampling				
	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT [18]	2.06	4.50	270.3	**0.82**	0.59	1.95	-	259.5	-	-
REPA [32]	1.42	4.70	305.7	0.80	0.65	1.29	-	306.3	**0.79**	0.64
REPA-E [15]	1.26	4.11	**314.9**	0.79	0.66	1.12	4.09	302.9	**0.79**	0.66
DDT [28]	1.40	-	303.6	-	-	1.26	-	**310.6**	**0.79**	0.65
VA-VAE [31]	1.35	4.15	295.3	0.79	0.65	1.23	4.20	296.0	**0.79**	0.65
ReDi [14]	1.61	4.66	295.1	0.78	0.64	1.60	5.99	294.7	0.78	0.64
REG [29]	1.36	4.25	299.4	0.77	0.66	1.19	4.44	305.4	0.78	0.66
RAE [33] (DiT^{DH}-XL)	1.28	-	262.9	-	-	1.13	-	262.6	0.78	**0.67**
SFD (XL)	**1.18**	**3.89**	266.8	0.78	**0.67**	**1.06**	**3.89**	267.0	0.78	**0.67**

Table 8. **Comparison of unconditional generation on ImageNet 256×256.** RG and AG are short of Representation Guidance [14] and AutoGuidance [13].

Method	Epochs	Params	FID↓	IS↑
DiT-XL [20]	400	675M	30.68	32.7
ReDi [14]	80	675M	25.10	–
ReDi [14] (w/ RG)	80	675M	22.60	–
RAE [33] (w/ AG)	200	839M	4.96	123.1
RCG [16] (DiT-XL/2)	400	675M	4.89	143.2
RCG [16] (MAGE-L)	800	502M	3.44	186.9
RCG-G [16] (MAGE-L)	800	502M	**2.15**	**253.4**
SFD (w/o AG)	80	675M	10.24	78.5
SFD (w/ AG)	80	675M	3.77	127.9
SFD (w/o AG)	200	675M	8.46	89.9
SFD (w/ AG)	200	675M	2.90	148.5

Table 9. **Effect of SFD for VA-VAE.**

TexEnc	SFD	FID↓
VA-VAE	✗	4.52
VA-VAE	✓	4.14
SD-VAE (ours)	✓	**3.03**

Table 10. **Computational cost and performance comparison** between LightningDiT and LightningDiT+SFD at 400K iterations on ImageNet 256×256. SFD adds negligible computational overhead while delivering substantially improved generation quality.

Method	#Params (M)↓	GFLOPs↓	FID↓
LightningDiT-XL	683.39	**116.479**	9.29
LightningDiT-XL + SFD	**682.77**	116.487	**3.53**

Different target representation and model scaling. Tab. 11 (a) compares several pretrained vision encoders used as target representations. Among all candidates, DINOv2-B achieves the lowest FID of 3.03, outperforming MAE [8], CLIP [21], and SigLip [26], indicating that DINOv2 provides the most effective supervision for compact semantic latent learning. Tab. 11 (b) studies different model scales within the DINOv2 family. Larger encoders yield better semantic guidance, with DINOv2-L achieving the best FID of 2.97. Notably, this finding stands in contrast to recent works like REG [29] and RAE [33], which identified DINOv2-B as the optimal choice and observed performance degradation when scaling to larger VFMs due to their increased dimensionality. Our results demonstrate the superiority of our explicit semantic compression strategy, which effectively handles high-dimensional features and unlocks the potential for further scaling with more powerful VFMs. Considering the trade-off between performance and efficiency, we adopt DINOv2-B as the default pretrained visual encoder.

Channel capacity. DINOv2-B outputs 768-dimensional features, which are compressed by the Semantic VAE into a lower-dimensional semantic latent. Tab. 11 (c) investigates the impact of varying the latent channel capacity. We observe a consistent performance improvement as the number of channels increases from 2 to 16. This trend indi-

Table 11. **Ablation on Semantic VAE design.** (a) compares different target representation models; (b) studies model scaling within DINOv2 family; (c) analyzes semantic channel capacity.

Target Repr.	FID↓
DINOv2-B	**3.03**
MAE-B	6.29
CLIP-B	4.89
SigLip-B	4.15

(a) Model comparison.

Target Repr.	FID↓
DINOv2-S	4.14
DINOv2-B	3.03
DINOv2-L	**2.97**

(b) Scaling comparison.

#Channels	FID↓
2	3.90
4	3.67
8	3.16
16	**3.03**

(c) Channel capacity.

Table 12. **Different reconstruction objectives for SemVAE.**

Recon Loss	FID ↓
MSE only	10.79
Cosine similarity only	10.71
MSE + Cosine similarity	**10.14**

Table 13. **Effect of semantic loss weight.**

Weight β	0.25	0.5	1.0	2.0	4.0	8.0
FID↓	3.46	3.26	3.08	**3.03**	3.28	3.96

cates that a higher channel capacity is essential for preserving the rich semantic information embedded in the original high-dimensional features. The 16-channel configuration achieves the best FID of 3.03, confirming that retaining more semantic details directly contributes to superior generation quality.

SemVAE reconstruction objective. Tab. 12 compares different reconstruction objectives for training the Semantic VAE. We consider MSE loss, cosine similarity loss, and their combination. MSE preserves feature magnitude fidelity, while cosine similarity encourages angular alignment in the high-dimensional feature space. Using either loss alone yields inferior performance, with FID scores of 10.79 and 10.71, respectively. Their combination achieves the best FID of 10.14, so we use the combined objective by default.

C.2. Effect of semantic loss weight

Tab. 13 analyzes the impact of the semantic loss weight β in the velocity prediction objective. As the weight increases from 0.25 to 2.0, the FID score consistently decreases, indicating that stronger semantic supervision enhances training stability and generation performance. However, when β becomes excessively large (e.g., 4.0 or 8.0), the performance degrades, suggesting that overemphasizing semantics suppresses texture learning and leads to loss of fine details. Overall, $\beta = 2.0$ achieves the best balance between semantic guidance and texture refinement, yielding the lowest FID of 3.03.

C.3. Effect of REPA configurations

Alignment depth. Tab. 14 presents a systematic study of REPA configurations. In our experiments, applying the REPA loss at shallow layers (specifically at depth 2) yields the best performance with an FID of 3.03, whereas the original REPA [32] reports optimal performance at depth 8. We attribute this discrepancy to the distinct role of the alignment loss in our framework. While the original REPA operates as a distillation process that forces the diffusion model to gradually *analyze and understand* the input latents, our approach utilizes the REPA loss to drive the model to *decode and reconstruct* high-level semantic representations from the noisy compressed latents. Since decoding from a semantic latent is inherently a more straightforward task than analyzing semantics from scratch, our model can achieve effective alignment at much shallower layers. Consequently, early-layer alignment suffices to recover the semantic guidance, avoiding the need for deeper intervention.

Table 14. **Ablation on REPA configurations.** Depth of conducting REPA loss, loss weight λ, and loss type are included.

Depth	Weight λ	Type	FID↓
–	–	–	4.15
2	0.5	cosine+MSE	**3.03**
4	0.5	cosine+MSE	3.07
6	0.5	cosine+MSE	3.24
8	0.5	cosine+MSE	3.16
10	0.5	cosine+MSE	3.19
12	0.5	cosine+MSE	3.28
2	0.25	cosine+MSE	3.30
2	0.5	cosine+MSE	**3.03**
2	1.0	cosine+MSE	3.18
2	2.0	cosine+MSE	3.25
2	4.0	cosine+MSE	3.20
2	0.5	cosine	3.16
2	0.5	MSE	3.13
2	0.5	cosine+MSE	**3.03**

Figure 1. **Semantic–Texture Separation.** (a) Both semantic and texture noise varied. (b) Semantic noise fixed, texture noise varied.

REPA loss weight λ. For the REPA loss weight λ, the model achieves the lowest FID at $\lambda = 0.5$. This indicates that a moderate alignment strength provides a good balance between semantic consistency and generative fidelity.

REPA similarity function. We also compare results of different REPA similarity functions. While conventional REPA employs cosine similarity for feature alignment, we additionally explore combining cosine and MSE losses inspired by our SemVAE training. The combined objective (cosine+MSE) achieves the best performance of 3.03 FID score, outperforming single-loss variants. This suggests that employing a similarity function consistent with the SemVAE training metric yields optimal results. Furthermore, it demonstrates the complementary nature of the two terms: MSE ensures distribution-level precision, whereas cosine similarity enhances directional alignment, leading to better semantic matching and visual realism.

It is worth noting that the optimal settings identified in this ablation study differ slightly from the final hyperparameters presented in Table 2. This discrepancy arises because the ablation experiments were evaluated at 400K iterations; however, over the full training duration (4M iterations), the configuration detailed in Table 2 yielded superior performance. Consequently, our final model adopts the settings from Table 2 rather than strictly following the ablation outcomes.

D. Limitation and Future Work

Currently, SFD employs a fixed temporal offset Δt to manage the asynchronous denoising process. However, a static offset may not be optimal across all noisy levels. Future work could explore dynamic or adaptive schedules for Δt to further enhance the synergy between semantic and texture generation. Furthermore, our framework presently relies on the REPA loss as an auxiliary objective to enforce feature alignment. A promising direction for future research is to investigate methods that eliminate the need for such auxiliary supervision, aiming for a cleaner and more streamlined optimization structure.

Beyond algorithmic refinements, extending and scaling SFD to more complex application scenarios represents a highly valuable research direction. Specifically, adapting SFD to text-to-image and text-to-video generation tasks could further validate its potential in handling intricate multimodal guidance and temporal consistency.

E. More Visualization Results

We further provide visualizations in Fig. 1 to demonstrate the separation between semantics and textures. Varying the semantic latent changes the global composition and layout, as shown in Fig. 1(a). In contrast, fixing the semantic noise while varying the texture noise preserves the overall structure but changes local appearance details (e.g., fur texture, scarf appearance), as shown in Fig. 1(b). These results qualitatively demonstrate a clear separation between semantic and texture latent.

We qualitatively compare the training progression in Figure 2, where all models are evaluated using the same initial noise. The baseline LightningDiT, REPA, and VA-VAE variants exhibit weaker structural consistency and struggle to form coherent details in the early training stages. In contrast, SFD produces clearer structures and more realistic details at much earlier iterations, demonstrating noticeably faster convergence.

We also present more visualization results of SFD in Figures 3 - 11.

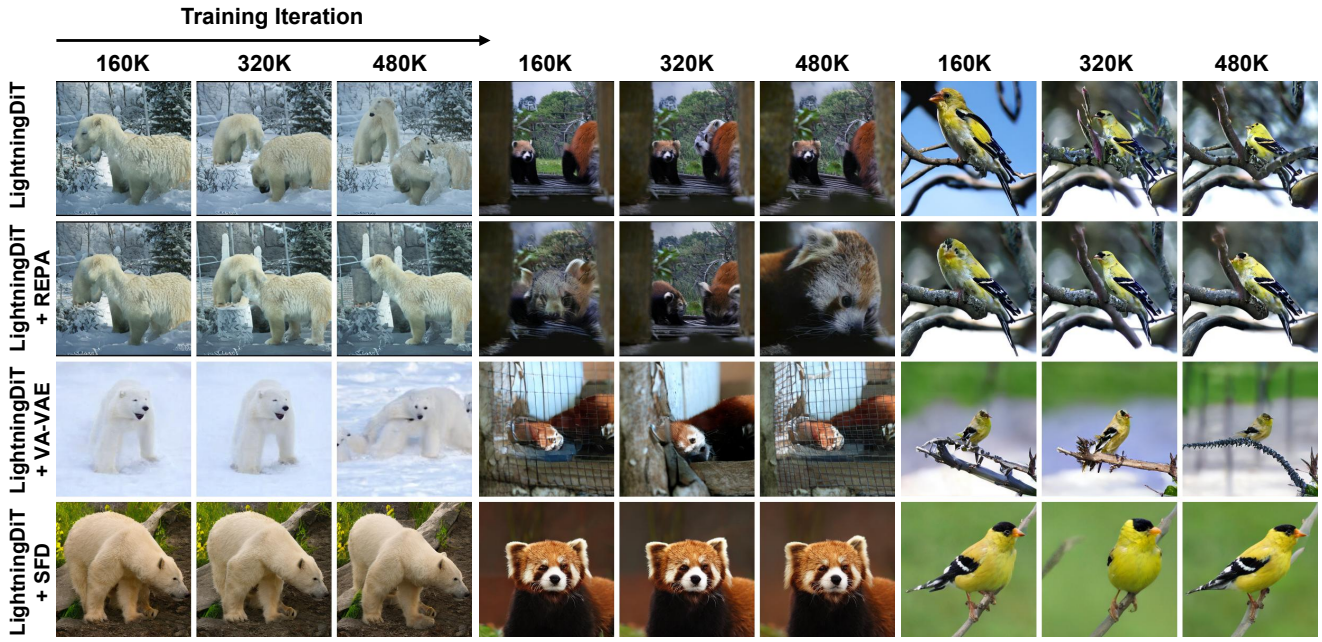

Figure 2. **Visualization of training results across different iterations (160K, 320K, and 480K).** Under a fixed random seed and identical initial noise, SFD produces clearer structures and more realistic details at early stages, demonstrating faster convergence compared with other variants.

Figure 3. Visualization results of LightningDiT-XL + SFD for the ImageNet class “Bald eagle” (22).

Figure 4. Visualization results of LightningDiT-XL + SFD for the ImageNet class “Sulphur-crested cockatoo” (89).

Figure 5. Visualization results of LightningDiT-XL + SFD for the ImageNet class “Giant panda” (388).

Figure 6. Visualization results of LightningDiT-XL + SFD for the ImageNet class “Teapot” (848).

Figure 7. Visualization results of LightningDiT-XL + SFD for the ImageNet class “Hamburger” (933).

Figure 8. Visualization results of LightningDiT-XL + SFD for the ImageNet class “Strawberry” (949).

Figure 9. Visualization results of LightningDiT-XL + SFD for the ImageNet class “Castle” (483).

Figure 10. Visualization results of LightningDiT-XL + SFD for the ImageNet class “Lakeside” (975).

Figure 11. Visualization results of LightningDiT-XL + SFD for the ImageNet class “Hot-air balloon” (417).

References

[1] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025. 4

[2] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 1

[3] 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*, pages 248–255. Ieee, 2009. 1

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 4

[5] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980. 1

[6] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 23164–23173, 2023. 4

[7] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023. 4

[8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 5

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[10] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. *arXiv preprint arXiv:2410.19324*, 2024. 4

[11] Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022. 4

[12] 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. 2, 3

[13] 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. 5

[14] Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. *arXiv preprint arXiv:2504.16064*, 2025. 3, 4, 5

[15] 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. *arXiv preprint arXiv:2504.10483*, 2025. 3, 4, 5

[16] Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. *Advances in Neural Information Processing Systems*, 37:125441–125468, 2024. 5

[17] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024. 2, 4

[18] 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*, pages 23–40. Springer, 2024. 3, 4, 5

[19] 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. 1

[20] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 4, 5

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 5

[22] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. *arXiv preprint arXiv:2502.20388*, 2025. 2, 4

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[24] 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. *arXiv preprint arXiv:2510.15301*, 2025. 3, 5

[25] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024. 2, 4

[26] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 5

[27] Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. *arXiv preprint arXiv:2507.23268*, 2025. 4

[28] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025. 2, 3, 4, 5

[29] Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *arXiv preprint arXiv:2507.01467*, 2025. 3, 4, 5

[30] Jingfeng Yao, Cheng Wang, Wenyu Liu, and Xinggang Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification. *Advances in Neural Information Processing Systems*, 37:56166–56189, 2024. 4

[31] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025. 1, 3, 4, 5

[32] 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 *The Thirteenth International Conference on Learning Representations*, 2025. 3, 4, 5, 6

[33] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025. 2, 3, 4, 5

[34] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv preprint arXiv:2306.09305*, 2023. 4