VUM: Visual Unified Model for Image Generation and Perception

Zidong Wang[1*✉] Yiyuan Zhang[1*✉] Xiaoyu Yue[2] Xiangyuan Xue[1]
Zhangquan Chen[3] Manyuan Zhang[†] Wanli Ouyang[1†]

[1]The Chinese University of Hong Kong [2]The University of Sydney [3]Tsinghua University

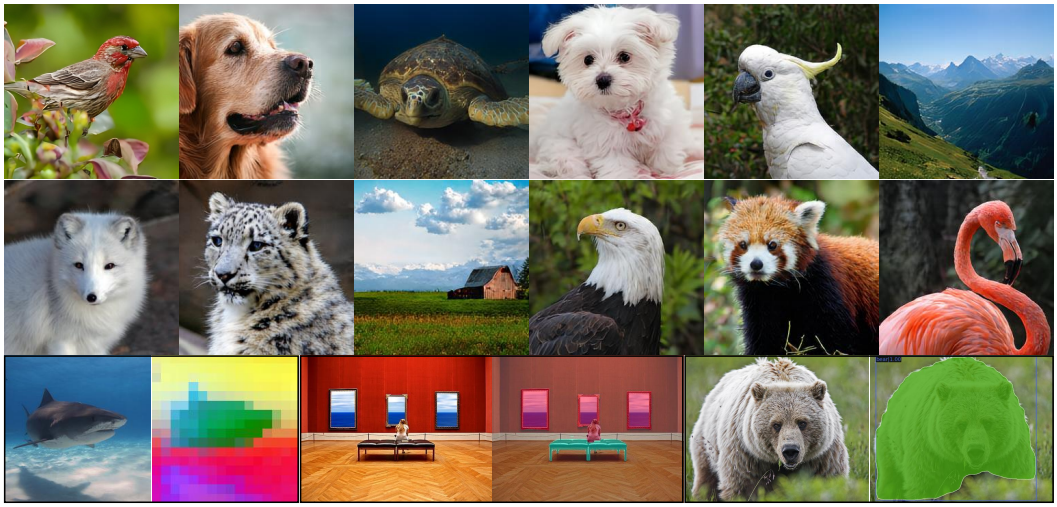


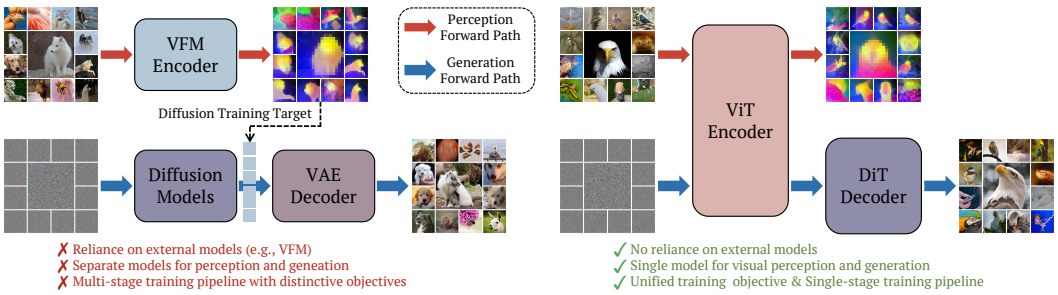
Figure 1: **One unified model, dual capabilities.** The Visual Unified Model (VUM) seamlessly integrates pixel synthesis and semantic perception within a single architecture. The top two rows showcase high-fidelity image generation on the ImageNet 256×256 benchmark, while the bottom row demonstrates its ability to extract highly effective representations for dense understanding tasks, including object detection, semantic segmentation, and instance segmentation.

Abstract

Unifying visual perception and generation poses a fundamental dichotomy: perception extracts semantics but discards details, while generation synthesizes fine-grained structures. Thus, existing works rely on decoupled architectures, as forcing these opposing information flows into one network inevitably triggers task interference and representation collapse. To break this bottleneck, we introduce the Visual Unified Model (VUM), a native framework that unifies both capabilities within a single architecture. Our key insight is that visual perception and generation can be unified by re-envisioning Masked Image Modeling and Diffusion Processes as complementary facets of data recovery. By establishing a shared degradation space that integrates both masking and noising, VUM optimizes a joint signal recovery objective: simultaneously predicting masked patches and denoising corrupted signals. This objective enforces the dual internalization of semantic abstraction and generative synthesis. Using one shared encoder, VUM

[*]Equal contribution. [†]Corresponding authors. [✉]Email: {wangzd2022, yiyuanzhang.ai}@gmail.com.

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



(a) Conventional Decoupled Models (b) Our Visual Unified Models

Figure 2: **Comparison of visual unification modeling paradigms.** (a) Conventional methods [83, 56] rely on an external Visual Foundation Model (VFM) as a feature extractor, employing multi-stage pipelines that require separate VAE and diffusion model training for synthesis. (b) Our VUM integrates perception and generation into a single backbone, enabling a single-stage training pipeline.

achieves competitive image generation and transfers to visual recognition and dense prediction within a single pixel-native model. Our implementation code is available at: https://github.com/WZDTHU/VUM.

1 Introduction

Unifying perception and generation within a single foundation model is a critical milestone toward Artificial General Intelligence (AGI) [28, 57, 67, 12]. Large Language Models (LLMs) [45, 76, 3, 64] achieve this through the autoregressive "next-token prediction" paradigm. The causal nature of text compels the model to comprehend the preceding context as a prerequisite for synthesizing future content. In computer vision, however, perception and generation remain bifurcated. Diffusion models [24, 50, 41, 69, 17, 51, 71] optimize low-level sensory fidelity without incentivizing semantic abstraction, whereas self-supervised learning [49, 21, 8] extracts invariant semantics via lossy compression, discarding the spatial granularity crucial for generation. Consequently, visual capabilities remain isolated across sensory and semantic spaces. Lacking the inherent causality of language, vision still misses an analogous mechanism to bridge this gap under a singular optimization signal.

Recent latent-space approaches [83, 56, 63], attempt to circumvent this divide through a multi-stage pipeline anchored to pre-trained vision foundation models (VFM)[20, 49, 21, 52]. A frozen VFM first projects visual signals into latent tokens; a VAE decoder is then trained for pixel reconstruction on these tokens [53, 18]; finally, a generative backbone (e.g., DiT [50]) is independently trained for synthesis. As illustrated in Fig. 2, this design delegates *perception* entirely to the vision encoder and *generation* to the generative backbone. Consequently, these capabilities remain structurally decoupled; rather than being unified within a single learning objective, they are merely stitched together and forced to communicate through the artificial bottleneck of latent tokens.

A fundamental question remains unresolved: *What constitutes the true native paradigm for unifying visual perception and generation?* We draw our initial inspiration from the structural isomorphism between diffusion formulation [24, 39, 60] and masked image modeling (MIM) [20, 74, 2]. Despite driving distinct capabilities historically, both paradigms share a fundamental objective: recovering the original data manifold from corrupted observations, whether through continuous Gaussian noise or discrete spatial masking. This observation motivates our core insight: *the paradigm for visual unification is not a multi-task ensemble, but a singular, joint signal recovery process.* Rather than treating discrete masking and continuous noising as isolated tasks, we advocate for a joint visual signal recovery objective: the model must simultaneously deduce masked patches from visible context and reconstruct clean signals from Gaussian perturbations. This dual constraint fundamentally dictates the network's evolution: masked reconstruction forces the extrapolation of global semantics, while denoising compels the synthesis of high-fidelity structures. Consequently, a single network internalizes both perceptual abstraction and generative synthesis.

However, actualizing this joint paradigm is non-trivial. Naively superimposing discrete masking and continuous noising is highly ineffective. This approach not only induces severe information loss through compounded degradation but also triggers gradient interference within a shared architecture, ultimately resulting in catastrophic mutual degradation that severely compromises both capabilities.

To this end, we introduce the Visual Unified Model (VUM), a native framework that unifies visual generation and understanding within a singular architecture through three key innovations: **1**) *Unified Pixel-Level Pretraining via Information-Balanced Degradation*. By analyzing degradation through Signal-to-Noise Ratio (SNR) [48], we derive a Hybrid SNR (HSNR) formulation that mathematically bridges spatial masking and Gaussian noising. This metric precisely quantifies aggregate information loss, allowing us to dynamically couple the mask ratio with the noise level to ensure information-balanced degradation. By optimizing pixel-level signal recovery within this space, VUM captures both global semantic priors and fine-grained structural details. **2**) *Asymmetric Architecture Design*. To resolve the inherent gradient conflicts during joint training, VUM employs a shared ViT [16] encoder coupled with an asymmetric dual-decoder. By pairing a primary DiT [50] decoder for denoising with an auxiliary ViT decoder for masked reconstruction, this design effectively mitigates optimization interference between semantics and generation. **3**) *Dual-Task Inference under a Singular Framework*. During inference, the auxiliary ViT decoder is discarded, allowing VUM to achieve bimodal execution without extraneous overhead. The ViT encoder extracts representations independently for perception while coupling with the DiT decoder for pixel synthesis. Thus, VUM unifies bimodal execution within a shared parameter space, bypassing the need for task-specific routing or network modifications.

Extensive evaluations on image generation and downstream perception benchmarks validate the effectiveness of our design. To the best of our knowledge, VUM is the first framework to natively unify semantic perception and generative synthesis directly in pure pixel space. A single VUM achieves competitive pixel-space generative fidelity (an FID of 1.91 on the ImageNet 256×256 benchmark [14]), while its shared encoder transfers to recognition and dense prediction, attaining 51.3 mIoU on ADE20K [85], and 53.1 AP^{box} and 46.3 AP^{mask} on COCO [38]. These results demonstrate competitive unification of pixel synthesis and semantic perception within a single pixel-native model.

2 Related Work

Explorations of Visual Perception and Generation. Current explorations predominantly rely on multi-stage pipelines, where an autoencoder [1] is first optimized for feature extraction and reconstruction, followed by the training of an independent generative model within this latent space. To enhance perception, one line, such as VA-VAE [78], VILA-U [73], and MAETok [6], aligns the latent space with pre-trained vision models (e.g., DINO [4, 49]). Alternatively, frameworks like RAE [83] and SVG [56] directly employ frozen pre-trained encoders, optimizing only the decoder for reconstruction. Others (UniTok [42], VTP [77]) leverage cross-modal supervision; for instance, VTP incorporates textual modalities and the CLIP [52] objective to augment visual perception.

Despite these advances, such methods share two limitations: 1) reliance on external priors (e.g., pre-trained vision features or text guidance), preventing true unification from raw pixels; and 2) structural decoupling, where perception and generation are handled by separate networks. Even concurrent joint-modeling efforts such as DREAM [33] still necessitate an auxiliary CLIP encoder to improve semantics. In contrast, VUM discards all external dependencies, coupling perception and generation within a single network through pure pixel-level degradation recovery.

Self-Supervised Learning. Advances in visual SSL can be broadly categorized into discriminative learning and masked image modeling (MIM) paradigms. Discriminative models (e.g., MoCo [21], SimCLR [8], DINO [4]) learn robust representations by maximizing agreement across augmented views. While this yields strong semantic discriminability for recognition tasks, it inevitably discards the fine-grained spatial details necessary for pixel-level synthesis. Alternatively, MIM frameworks (e.g., MAE [20], SimMIM [74], BEiT [2]) capture rich contextual and perceptual information by reconstructing masked inputs to model the underlying data distribution. Although these representations trail behind discriminative methods in global semantic separability, their pixel-level reconstruction objective shares a critical structural isomorphism with diffusion denoising mechanisms. Motivated by this alignment, our VUM adopts the pixel-recovery paradigm to unify perception and synthesis.

[1] An autoencoder [23, 30] typically comprises an encoder and a decoder: the encoder compresses images into a latent space, while the decoder reconstructs the original pixels from these latent tokens.

Pixel-Space Diffusion Models. Despite the dominance of latent-space paradigms, direct modeling of pixels is resurging to preserve maximum signal fidelity. Early models (e.g., DDPM [24], ADM [15]) were anchored in convolutional U-Nets [54]; as the paradigm evolved to embrace Vision Transformers (ViTs), directly modeling raw pixels triggered a severe sequence length explosion at high resolutions. To mitigate this complexity, SiD2 [27] adopts a memory-efficient U-ViT [1] with level-wise skip connections, while PixelFlow [7] dynamically upscales samples as denoising progresses. Others utilize dual-level architectures to decouple global semantics from local refinement; for instance, PixNerd [66] conditions NeRF-like rendering [46] on transformer features for patch-level reconstruction, and PixelDiT [80] cascades a patch-level DiT with a pixel-level DiT. In contrast, JiT [35] demonstrates that a plain ViT can directly predict clean images from raw patches without specialized architectural biases. Transcending this, our VUM establishes a singular, unified Transformer architecture that internalizes both perceptual understanding and generative synthesis.

3 Method

3.1 Preliminaries

Masked Image Modeling. As a canonical instantiation of Masked Image Modeling [2, 20, 74], MAE [20] learns visual representations by reconstructing missing pixels from a set of visible tokens. Formally, given an image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$, it is partitioned into N non-overlapping patches. A binary mask $\boldsymbol{M} \in \{0,1\}^N$ is sampled to mask 75% of the patches, where $\boldsymbol{M}_i = 1$ indicates a masked patch. The encoder $\mathcal{E}$ is a Vision Transformer (ViT) that operates exclusively on the sequence of visible tokens $\boldsymbol{x}_{\text{vis}}$ (extracted via $\boldsymbol{x} \odot \boldsymbol{M}$). A generic learnable $[\texttt{CLS}]$ token $\boldsymbol{e}_{\text{cls}}$ is prepended to this sequence, yielding the latent representation $\boldsymbol{h} = \mathcal{E}([\boldsymbol{e}_{\text{cls}}, \boldsymbol{x}_{\text{vis}}])$. The decoder $\mathcal{D}$ processes the concatenated full sequence $\hat{\boldsymbol{x}} = \mathcal{D}([\boldsymbol{h}, \boldsymbol{s}])$, where $\boldsymbol{s}$ represents a set of shared and learned mask tokens placed at the missing positions ($\boldsymbol{M}_i = 1$). The loss is computed exclusively on the masked patches: $\mathcal{L}_{\text{MAE}} = \mathbb{E}\|(\texttt{Norm}(\boldsymbol{x}) - \hat{\boldsymbol{x}}) \odot \boldsymbol{M}\|_2^2$, where $\texttt{Norm}(\cdot)$ denotes the unit-variance normalization.

Diffusion Models. As a prevalent formulation in diffusion models [24, 60, 59, 40, 39, 58, 81, 68, 70], flow-matching [39] linearly interpolates between sampled noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I})$ and data $\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})$ via: $\boldsymbol{x}_t = (1-t)\boldsymbol{x} + t\boldsymbol{\varepsilon}$, where $t \in [0,1]$ is the sampled time step. In this scheduling, JiT [35] directly predicts the clean data manifold from noisy input: $\hat{\boldsymbol{x}} = \mathcal{D}_{\text{G}}(\boldsymbol{x}_t, t)$, where $\mathcal{D}_{\text{G}}$ is a Diffusion Transformer (DiT) architecture with a patch size of 16. The training objective is to minimize the reconstruction error in the pixel space: $\mathcal{L}_{\text{Diff}} = \mathbb{E}_t\|\boldsymbol{x} - \mathcal{D}_{\text{G}}(\boldsymbol{x}_t, t)\|_2^2$.

3.2 Motivation

To construct a truly unified visual model, an intuitive first step is to directly combine the objectives of MIM and diffusion within a single backbone. To empirically investigate the feasibility of this integration, we evaluate three baselines: (1) a standard MAE-B baseline; (2) a 24-layer JiT-B baseline [2]; and (3) a naively fused 24-layer JiT-B, trained concurrently with standard noising and 75% spatial masking. We extract features from the 12-th layer in all models for linear probing (IN1K-LP).

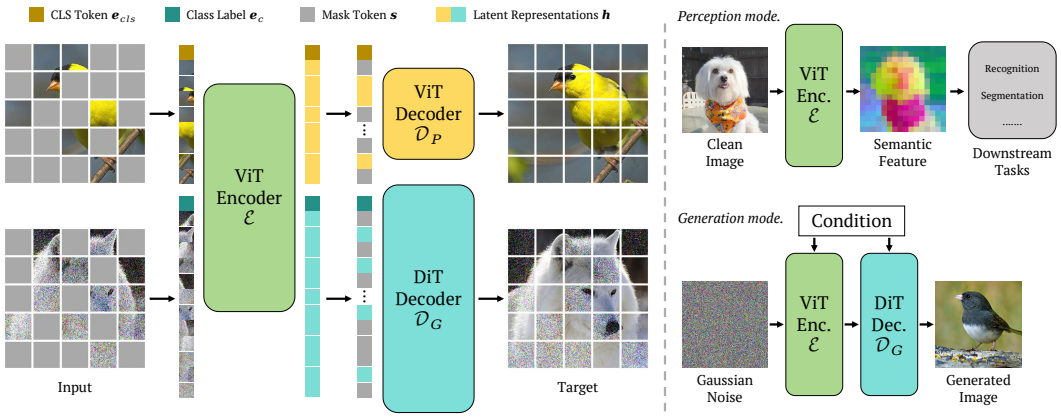
Tab. 1 reveals three key findings: 1) pure MAE yields strong perceptual representations ($\sim 45\%$ accuracy) its asymmetric design and discrete reconstruction objective preclude conditional image generation; 2) pure JiT fails at understanding tasks ($< 10\%$ IN1K-LP accuracy), demonstrating generative objectives alone lack perceptual abstraction; and 3) naively fusing both causes severe mutual degradation, deteriorating generation quality (FID 316.9) while keeping probing accuracy negligible.

Table 1: Empirical baselines: Naïve integration of MIM and diffusion leads to catastrophic mutual degradation.

Config	FID↓	IN1K-LP↑
MAE-B	-	44.85
JiT-B (24-Layer)	9.16	6.97
JiT-B (75% Mask)	316.9	6.57

These empirical failures indicate that perception and generation cannot be seamlessly unified by simply forcing a network to naively blend divergent training objectives. To establish a truly singular architecture, we must first dissect the deep-rooted incompatibilities that drive this mutual interference. Specifically, we identify four intrinsic challenges: *C1* **Compounding Information Loss.** Superimposing aggressive spatial masking (e.g., 75%) with diffusion's Gaussian noise severely depletes the input signal. Predicting pixels from such an information-starved state is fundamentally ill-posed,

[2]We scale the depth of JiT-B from the standard 12 layers to 24 layers to precisely match the total capacity of our VUM (detailed in Eq. (3)) for a fair comparison. The three models are trained for 160 epochs on ImageNet-1K [14].



(a) Overview of Visual Unified Training Pipeline. (b) Dual-Task Inference.

Figure 3: **Overview of Visual Unified Models (VUM).** (a) Visual Unified Training. Inputs undergo joint degradation where the mask ratio dynamically adjusts with the noise level. A shared ViT encoder $\mathcal{E}$ extracts latent features, routed to task-specific decoders ($\mathcal{D}_P$, $\mathcal{D}_G$) for joint pixel-level recovery. (b) Dual-Task Inference. For perception, standalone $\mathcal{E}$ extracts semantics directly from clean images. For generation, $\mathcal{E}$ pairs with the DiT decoder $\mathcal{D}_G$ for conditional pixel synthesis from pure noise.

inevitably triggering optimization collapse. *C2* **Structural Asymmetry.** MAE favors an asymmetric, encoder-heavy architecture to abstract semantics, whereas generative models necessitate substantial decoder capacity for iterative spatial refinement. *C3* **Input Discrepancy.** During inference, perception operates on clean images, whereas diffusion synthesizes content from pure Gaussian noise under a specific condition. This creates a severe input manifold gap that a naive shared backbone struggles to bridge. *C4* **Misaligned Optimization Gradients.** The two objectives inherently conflict: MAE abstracts invariant semantics by discarding local details, while diffusion strictly retains high-frequency pixels. Consequently, a naive linear combination of these losses triggers severe gradient interference.

The structural and input discrepancies (*C2* and *C3*) are evident. Regarding Challenges 1 and 4, we provide further empirical evidence for the information degradation and optimization conflicts in the experiment Sec. 4.3. Together, these limitations necessitate a rethinking of unified visual models.

3.3 Visual Unified Models

Overcoming the aforementioned barriers requires far more than a simple additive loss; it demands a paradigm shift from the degradation mechanism to the architectural design. As illustrated in Fig. 3, we present the Visual Unified Model (VUM), which dismantles the inherent perception-generation conflicts through three core designs: *1)* **Information-Balanced Degradation.** Directly addressing the issue of signal depletion (*C1*), we introduce a joint scheduler governed by a Hybrid SNR formulation. By dynamically scaling the mask ratio inversely with the noise level, this formulation prevents excessive signal depletion. *2)* **Asymmetric Architecture with Semantic Bridging.** Addressing both structural mismatches (*C2*) and input discrepancies (*C3*), we design an asymmetric ViT-DiT topology. To bridge clean images and pure noise during inference, we dynamically repurpose the [CLS] token as a unified conditional anchor, seamlessly connecting the two modalities. *3)* **Dual-Decoder Design.** To eliminate the gradient interference typical of coupled architectures (*C4*), VUM routes the latent features into decoupled task-specific heads. The generative synthesis is entrusted to a heavy DiT decoder, while a lightweight auxiliary ViT decoder handles pixel-level masked recovery.

3.3.1 Information-Balanced Degradation

Hybrid Signal-to-Noise Ratio (HSNR). To address the compounding information loss (*C1*), we must quantify the aggregate signal destruction caused by concurrent masking and noising. To this end, we introduce the Hybrid Signal-to-Noise Ratio (HSNR), a metric that explicitly quantifies the ratio of preserved signal power to total visual uncertainty. We defer the rigorous mathematical derivations to Appendix A and summarize the core formulations here. First, the independent SNRs for continuous

diffusion and discrete masking are formulated as:

$$SNR_{\text{Diff}}(t) = \frac{(1-t)^2}{t^2}, \quad SNR_{\text{MAE}}(r) = \frac{1-r}{r}. \tag{1}$$

Building upon these isolated states, we deduce the joint degradation. When superimposing a spatial mask r onto a noisy continuous manifold $\mathbf{x}_t$, the preserved signal power becomes $(1-r)(1-t)^2$, while the total uncertainty aggregates the continuous noise $(1-r)t^2$ and the discrete spatial loss r. This mathematically yields the exact Hybrid SNR:

$$SNR_{\text{Hybrid}}(r,t) = \frac{(1-r)(1-t)^2}{(1-r)t^2 + r}. \tag{2}$$

Diagnosis of Information Death and Dynamic Balance. Eq. (2) formally exposes the root cause of the catastrophic collapse observed during naive fusion. A high mask ratio (e.g., $r = 0.75$) inherently induces a massive baseline loss of visual information, which is then severely exacerbated as the noise level t escalates. Specifically, as t increases, the signal term $(1-t)^2$ in the numerator decays quadratically to zero, whereas the degradation term $(1-r)t^2 + r$ in the denominator strictly increases. This simultaneous signal decay and noise expansion causes the overall SNR_{Hybrid} to non-linearly collapse toward zero (as shown in Fig. 4). Ultimately, this information death obliterates the visual signal, leaving the network with insufficient context to drive both generative and perceptual objectives.

To maintain a tractable information bottleneck throughout the forward process, we propose a dynamic *Information-Balanced Scheduling* strategy. Rather than applying a static mask, we inversely couple the mask ratio r with the continuous noise level t:

$$r(t) = r_{\text{h}} - t \cdot (r_{\text{h}} - r_{\text{l}}), \tag{3}$$

where $r_{\text{h}} = 0.75$ and $r_{\text{l}} = 0.0$. By substituting this dynamic schedule back into Eq. (2), we derive the Balanced HSNR, which is governed seamlessly by the temporal state t:

$$SNR_{\text{Balanced}}(t) = \frac{(1-r(t))(1-t)^2}{(1-r(t))t^2 + r(t)}. \tag{4}$$

Crucially, this dynamic formulation transcends naive multi-task learning by orchestrating an elegant curriculum of signal recovery. Under severe continuous noise ($t \to 1$), the schedule naturally lifts the spatial mask ($r(t) \to 0$), equipping the network with the dense spatial context required for structural denoising. Conversely, as the signal clarifies ($t \to 0$), severe spatial masking is introduced ($r(t) \to 0.75$), forcing the network to extract high-level semantic abstractions rather than memorizing local pixels. Ultimately, perception and generation are no longer antagonistically coupled; instead, they operate as a unified, complementary continuum of signal recovery. We further provide empirical explorations of alternative scheduling functions for $r(t)$ in Tab. 5.

3.3.2 Architecture and Dual-Task Execution

To resolve the remaining conflicts (*C2–C4*), we introduce the architectural designs of Visual Unified Model (VUM). Addressing structural asymmetry (*C2*), VUM pairs a shared ViT encoder with a heavy DiT decoder, directly satisfying perception's need for deep semantic abstraction and diffusion's demand for iterative spatial refinement. Building on this foundation, we further introduce specialized mechanisms to resolve input discrepancies (*C3*) and mitigate gradient conflicts (*C4*).

Trajectory-Adaptive Context Modulation and Efficient Encoding. To resolve the conditional input discrepancy (*C3*) without fracturing the unified degradation continuum, we introduce a trajectory-adaptive context modulation mechanism. Rather than resorting to modality-specific branches, we formalize the standard [CLS] token slot as a *noise-aware semantic pivot* governed by the trajectory $t \in [0, 1]$. As in Fig. 3 (a), towards the clean-state limit ($t \to 0$), the slot is instantiated with a learnable embedding $\mathbf{e}_{\text{cls}}$ to aggregate global semantic abstraction. Across the noisy spectrum ($t > 0$), it incorporates the class condition $\mathbf{e}_c$ to guide the spatial synthesis. This context modulation mechanism seamlessly unifies feature extraction and conditional generation within the same encoder.

Furthermore, inspired by MAE [20], the shared encoder $\mathcal{E}$ operates exclusively on unmasked patches $\boldsymbol{x}_{\text{vis}}$ to maximize efficiency. Because our dynamic schedule $r(t)$ inherently creates highly variable

sequence lengths within the same batch, we integrate a `varlen_flash_attn` backend [3]. This bypasses inefficient zero-padding and attention masking, allowing the encoder to project the modulated input sequences into a shared latent space $\boldsymbol{h} = \mathcal{E}([\mathbf{e}_{\text{cls/c}}, \boldsymbol{x}_{\text{vis}}])$ in a highly optimized forward pass.

Decoupled Decoding as Gradient Buffers. While the shared encoder unifies feature extraction, directly backpropagating from all temporal states causes severe gradient conflicts (further analyzed in the Appendix D.2). To resolve this (*C4*), we decouple the decoding phase into state-conditioned gradient buffers. After the encoder extracts the compressed latents $\boldsymbol{h}$ and pads them with mask tokens $\boldsymbol{s}$, the sequence is handled by two specialized decoders based on the trajectory state t:

Auxiliary ViT Decoder ($\mathcal{D}_P$): In the clean boundary state ($t = 0$), a lightweight ViT (an 8-layer ViT with a width of 512) decoder forces the encoder to abstract semantics by recovering the missing pixels. It is supervised by a standard MAE loss computed exclusively on the masked regions:

$$\mathcal{L}_{\text{P}} = \mathbb{E}\left[\|(\texttt{Norm}(\boldsymbol{x}) - \mathcal{D}_{\text{P}}([\boldsymbol{h}, \boldsymbol{s}])) \odot \boldsymbol{M}\|_2^2\right]. \tag{5}$$

DiT Decoder ($\mathcal{D}_G$): A heavy generative decoder processes the continuous noisy stream ($t > 0$). To handle noisy states with masking, $\mathcal{D}_{\text{G}}$ employs a region-specific objective. It applies standard denoising to visible regions and reconstructs the noisy structure for masked regions:

$$\mathcal{L}_{\text{G}} = \mathbb{E}_t\left[\|(\boldsymbol{x} - \mathcal{D}_{\text{G}}([\boldsymbol{h}, \boldsymbol{s}], t)) \odot (1 - \boldsymbol{M})\|_2^2 + \lambda\|(\texttt{Norm}(\boldsymbol{x}_t) - \mathcal{D}_{\text{G}}([\boldsymbol{h}, \boldsymbol{s}], t)) \odot \boldsymbol{M}\|_2^2\right], \tag{6}$$

where λ balances the gradient scales between denoising and structure reconstruction, following [84]. The full network is optimized end-to-end via a joint objective:

$$\mathcal{L}_{\text{Total}} = \beta\mathcal{L}_{\text{P}} + \mathcal{L}_{\text{G}}. \tag{7}$$

Ultimately, driven by the joint objective $\mathcal{L}_{\text{Total}}$, state-conditioned decoders act as independent gradient buffers: $\mathcal{D}_{\text{P}}$ absorbs stable semantic gradients, while $\mathcal{D}_{\text{G}}$ digests high-variance spatial dynamics. By resolving this divergence late in the network, VUM achieves a powerful synergy: it fully insulates the shared encoder to learn a universal visual abstraction, while simultaneously unlocking the DiT decoder to deliver high-fidelity generative synthesis without being bottlenecked by perceptual constraints.

Dual-Task Execution. As illustrated in Fig. 3 (b), VUM executes perception and generation through an adaptive inference strategy. For perception, the encoder $\mathcal{E}$ standalone extracts semantics for downstream adaptations. For generation, $\mathcal{E}$ pairs with the DiT decoder $\mathcal{D}_{\text{G}}$ to iteratively synthesize pixels from Gaussian noise, conditioned on $\mathbf{e}_c$. By discarding the auxiliary ViT decoder during inference, VUM maintains computational efficiency for both tasks without structural redundancy.

4 Experiments

4.1 Setup

Implementation Setup. The VUM architecture comprises a standard ViT encoder (patch size 16) coupled with a DiT decoder and a ViT decoder. The flagship VUM-L configuration instantiates ViT-L and DiT-L backbones, yielding 763M parameters. Training is conducted on ImageNet at 256×256 resolution (256 patches per image) with standard center-crop and horizontal flip augmentations. The network is trained with Muon for 800 epochs (40-epoch warmup), using a batch size of 2048 and a learning rate of 2×10^{-4}. We set $\lambda = 0.1$ in Eq. (6) and $\beta = 1.0$ in Eq. (7) by default. Following generative modeling conventions, an exponential moving average (EMA) with a 0.9999 decay rate is maintained. More details are provided in the Appendix C.

Evaluation Protocol. Generative synthesis is evaluated using FID [22], sFID [47], Inception Score (IS) [55], Precision and Recall [31], standardizing the computation via the ADM suite [15]. Perception performance is evaluated through ImageNet-1K linear probing and end-to-end fine-tuning. We also evaluate VUM on downstream tasks: object detection and instance segmentation on COCO [38] as well as semantic segmentation on ADE20K [85].

4.2 Unified Visual Capabilities on Generation and Perception

Image Generation Performance. We evaluate the generative capabilities of VUM on the standard ImageNet-256 benchmark against state-of-the-art autoregressive, latent diffusion, and pixel-space

[3] Specifically, the `varlen_flash_attn` interface [13] packs disparate sequences into a 1D array, utilizing an index of cumulative lengths to enforce strict attention boundaries, achieving isolated self-attention without explicit masking overhead.

Table 2: **Model comparisons on ImageNet-256 Benchmark.** All models utilize Classifier-Free Guidance (CFG) for generation. Latent-space models incur an overhead of 86M parameters for the VAE [53] or VQ-VAE [18] tokenizer.

Type	Model	#Params	Epochs	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Autoregressive Models	VAR [65]	600M+86M	-	2.57	-	302.6	0.83	0.56
	LlamaGen [62]	775M+86M	300	2.63	5.59	244.09	0.81	0.58
	ST-AR [82]	775M	300	2.37	6.05	270.59	0.82	0.58
	FractalMAR [36]	848M	600	6.15	-	348.9	0.81	0.46
Latent Diffusion Models	LDM [53]	400M+86M	170	3.60	5.12	247.67	0.87	0.48
	DiT [50]	675M+86M	1400	2.27	4.60	278.24	0.83	0.57
	MaskDiT [84]	700M+86M	1600	2.28	5.67	276.60	0.80	0.61
	SiT [43]	675M+86M	1400	2.15	4.50	258.09	0.81	0.60
	REPA [79]	675M+86M	800	1.42	4.70	305.7	0.80	0.64
Pixel Diffusion Models	CDM [25]	-	2160	4.88	-	158.7	-	-
	ADM[15]	554M	400	3.94	6.14	215.8	0.83	0.53
	SiD [26]	2.0B	800	2.44	-	256.30	-	-
	PixelFlow [7]	677M	320	1.98	-	282.1	0.81	0.60
	DiP [11]	631M	320	1.98	4.57	282.9	0.80	0.60
	PixNerd [66]	700M	320	1.95	-	298.0	0.80	0.60
	DeCo [44]	682M	320	1.90	4.47	303.0	0.80	0.61
	JiT [35]	953M	600	1.86	-	303.4	0.78	0.62
	VUM (Ours)	763M	800	1.91	4.92	288.3	0.80	0.62

Table 3: **Model comparisons on downstream perception tasks.** All evaluated models employ a ViT-L backbone. For our method, we isolate the ViT encoder of VUM for downstream evaluation. We report mIoU for semantic segmentation on ADE20K, alongside AP^{box} and AP^{mask} for object detection and instance segmentation on COCO, respectively.

Model	Pretrain Data	Epoch.	IN1K		ADE20K	COCO	
			FT↑	LP↑	mIOU↑	AP^{box} ↑	AP^{mask} ↑
MoCo v3 [10]	IN1K	600	84.1	77.6	49.1	49.3	44.0
BEiT [2]	IN1K+DALLE	1600	85.2	-	53.3	53.3	47.1
MAE [20]	IN1K	1600	86.0	76.0	53.6	53.3	47.2
VUM (Ours)	IN1K	800	84.4	68.4	51.3	53.1	46.3

diffusion models. As shown in Tab. 2, VUM achieves a competitive FID of 1.91 and an Inception Score (IS) of 288.3. Within the pixel diffusion paradigm, VUM achieves an FID close to JiT and DeCo (1.91 vs. 1.86 and 1.90, respectively), while improving over ADM (3.94) and DiP (1.98). Compared to latent diffusion models, VUM achieves lower FID and higher Recall than DiT (2.27) and MaskDiT (2.28) without relying on external VAEs. Together with the shared encoder's transfer performance, these results support competitive unification of generation and perception within a single pixel-native model.

Perceptual Representation. We evaluate the shared VUM encoder on downstream tasks in Tab. 3. For dense prediction, VUM achieves 51.3 mIoU on ADE20K and 53.1/46.3 AP on COCO, outperforming MoCo v3 and approaching MAE on COCO. VUM is pretrained for 800 epochs, compared with 1600 for MAE, while also learning image generation. On ImageNet-1K, VUM yields 84.4 FT accuracy and 68.4 LP accuracy, compared with 86.0 and 76.0 for MAE. The 7.6-percentage-point LP gap highlights a trade-off between generation and perceptual representation quality. Overall, the same encoder supports competitive transfer to recognition and dense prediction alongside image generation, with lower perceptual scores than MAE on these benchmarks.

Table 5: Ablations on the information-balanced scheduling of Balanced HSNR. We compare various scheduling functions that dictate the trajectory of the SNR across different noise levels.

Scheduling	FID↓	IS↑	IN1K-FT↑	IN1K-LP↑
Sqrt: $r(t) = r_h - t \cdot \sqrt{r_h - r_l}$	10.07	193.43	81.65	48.49
Square: $r(t) = r_h - t \cdot (r_h - r_l)^2$	23.18	76.63	81.59	41.66
Cosine: $r(t) = r_l + \frac{r_h - r_l}{2} \cdot (1 + \cos(\pi t))$	10.30	185.47	81.77	48.51
Linear: $r(t) = r_h - t \cdot (r_h - r_l)$	9.77	193.51	81.79	48.70

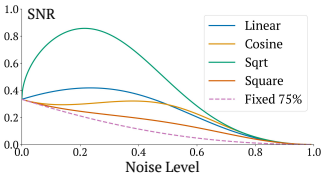


Figure 4: The SNR Curve of different scheduling functions.

Table 4: **Ablations on the components of VUM.** Architectural configurations are denoted as *Model-Layers/Width* (e.g., ViT-12/768 indicates a 12-layer ViT with a hidden dimension of 768). †: for the DiT-24/768 architectures, we use the 12-th layer feature for the end-to-end finetuning (IN1K-FT) and linear probing (IN1K-LP) evaluations on ImageNet1K.

Model	Params.	Encoder	Perceptual Decoder	Generative Decoder	Corruption	FID↓	IS↑	IN1K-FT↑	IN1K-LP↑
Perception and Generation Baselines									
MAE	148.7M	ViT-12/768	ViT-8/512	-	75% Mask	-	-	83.02	44.85
JiT†	258.7M	-	-	DiT-24/768	Noise	9.16	241.67	78.64	6.97
The evolution of VUM model									
Config1†	258.7M	-	-	DiT-24/768	75% Mask+Noise	316.9	2.07	74.41	6.57
Config2	217.3M	ViT-12/768	-	DiT-12/768	75% Mask+Noise	159.6	5.66	80.63	21.25
Config3	217.3M	ViT-12/768	-	DiT-12/768	Balanced HSNR	22.07	80.50	81.69	43.25
VUM	243.3M	ViT-12/768	ViT-8/512	DiT-12/768	Balanced HSNR	9.77	193.51	81.79	48.70

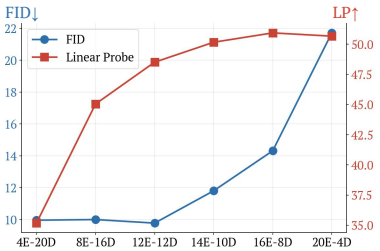


Figure 5: Ablations on the encoder-decoder layer ratio.

Table 6: Ablations on the spatial decoupling weight λ in Eq. (6). The task-balancing weight is fixed at $\beta = 1.0$.

λ	FID↓	IN1K-LP↑
0.0	9.52	22.79
0.1	9.77	48.70
0.5	12.42	46.36
1.0	13.41	48.08

Table 7: Ablations on the task-balancing weight β in Eq. (7). The spatial decoupling weight is fixed at the default $\lambda = 0.1$.

β	FID↓	IN1K-LP↑
0.5	9.75	37.53
1.0	9.77	48.70
1.5	12.23	46.01
2.0	12.52	33.44

4.3 Ablation Studies

We conduct comprehensive ablations to validate the design choices of VUM. All model variants are trained on ImageNet-1K at 256×256 resolution for 160 epochs, and generative metrics are reported using a classifier-free guidance (CFG) scale of 2.5. We ablate our framework by evaluating its architectural evolution from standard baselines (Tab. 4), HSNR scheduling strategies (Tab. 5), encoder-decoder depth ratios (Fig. 5), and loss-balancing coefficients (λ, β).

Evolution of VUM. Tab. 4 traces our structural trajectory. Task-specific baselines expose inherent divergence: MAE excels at perception (IN1K-LP 44.85) but cannot generate, whereas JiT synthesizes well (FID 9.16) but suffers representational collapse (LP 6.97). Naive joint training (*Config1*) fails catastrophically. Introducing an encoder-decoder (*Config2*) only marginally recovers perception while struggling with synthesis (FID 159.6). Crucially, integrating the information-balanced HSNR (*Config3*) bridges this gap, dramatically improving FID to 22.07 and LP to 43.25. Finally, the full VUM uses dual decoders to achieve a competitive balance: an FID of 9.77 compared with JiT's 9.16 and an IN1K-LP of 48.70 compared with the 160-epoch MAE baseline's 44.85, demonstrating generation and perception within one shared encoder.

HSNR Scheduling Strategies. We compare four $r(t)$ decay functions (Tab. 5, Fig. 4): our default Linear ($r_\text{h} - t(r_\text{h} - r_\text{l})$), Sqrt ($r_\text{h} - t\sqrt{r_\text{h} - r_\text{l}}$), Square ($r_\text{h} - t(r_\text{h} - r_\text{l})^2$), and Cosine ($r_\text{l} + \frac{r_\text{h} - r_\text{l}}{2}(1 + \cos(\pi t))$). The Square schedule drops SNR too rapidly, severely degrading both FID (23.18) and IN1K-LP (41.66). Conversely, the Sqrt schedule maintains an excessively high SNR, oversimplifying reconstruction, and hurting fidelity (FID 10.07). While Cosine transitions smoothly, the Linear schedule strictly dominates, yielding peak generative (FID 9.77) and semantic (LP 48.70) performance.

Capacity Allocation. Under a fixed 24-layer budget (Fig. 5), increasing encoder depth improves perception (LP), but overly compressing the decoder drastically degrades generation (FID). We find that a symmetric 12E-12D configuration yields the optimal trade-off between semantic abstraction and generative fidelity.

Objective Balancing. We ablate the spatial weight λ (Tab. 6) and task weight β (Tab. 7). Increasing λ from 0 to 0.1 improves LP from 22.79 to 48.70, with a small FID increase from 9.52 to 9.77. Similarly, $\beta = 1.0$ achieves the highest LP among the tested task weights while maintaining an FID of 9.77, illustrating the selected balance between generation and perception.

5 Conclusion and Limitation

We present the Visual Unified Model (VUM), a pixel-native model that jointly learns visual perception and image generation through a shared encoder. By conceptualizing generative noise and perceptual masking within a unified degradation continuum, VUM bridges both paradigms via an Information-Balanced (HSNR) schedule, task-adaptive routing, and an asymmetric dual-decoder topology. Without relying on external VAEs, VUM achieves competitive pixel-space generation, with ImageNet-256 FID close to JiT and DeCo (1.91 vs. 1.86 and 1.90), while the shared encoder transfers competitively to recognition and dense prediction. The ablations further show that coupling masking with the diffusion timestep and using an auxiliary perception decoder help balance generative fidelity and representation quality. These findings support further exploration of unified visual learning from raw pixels under a shared encoder.

Limitations and Future Work. VUM's linear-probing gap relative to MAE motivates further study of noise-conditioned normalization and constraints on clean-state features. Scaling to larger models and higher resolutions also requires addressing the computational costs of pixel-space modeling. Our generation experiments are limited to class-conditional ImageNet; extending VUM to text-conditioned, web-scale multimodal learning remains future work. On the perception side, we will also extend evaluation to depth and surface normal estimation.

Acknowledgments and Disclosure of Funding

This work was supported by the JC STEM Lab of AI for Science and Engineering, funded by The Hong Kong Jockey Club Charities Trust and the Research Grants Council of Hong Kong (Project No. CUHK14213224). No competing interests are declared for this work.

References

[1] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In CVPR, 2023.

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

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. NeurIPS, 2020.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pages 9650–9660, 2021.

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

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

[7] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. ArXiv, abs/2504.07963, 2025.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. ArXiv, abs/2002.05709, 2020.

[9] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. International Journal of Computer Vision, 132(1):208–223, 2024.

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057, 2021.

[11] 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. arXiv preprint arXiv:2511.18822, 2025.

[12] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261, 2025.

[13] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691, 2023.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009.

[15] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. NeurIPS, 2021.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[17] 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 ICML, 2024.

[18] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In CVPR, 2021.

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

[20] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. Masked autoencoders are scalable vision learners. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 15979–15988, 2021.

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9726–9735, 2019.

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS, 2017.

[23] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. NeurIPS, 1993.

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. NeurIPS, 2020.

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

[26] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In ICML, 2023.

[27] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 18062–18071, 2025.

[28] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. arXiv preprint arXiv:2410.21276, 2024.

[29] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In CVPR, pages 7482–7491, 2018.

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[31] T. Kynkäänniemi, T. Karras, S. Laine, and T Lehtinen, J.and Aila. Improved precision and recall metric for assessing generative models. NeurIPS, 2019.

[32] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 18262–18272, 2025.

[33] Chao Li, Tianhong Li, Sai Vidyaranya Nuthalapati, Hong-You Chen, Satya Narayan Shukla, Yonghuan Yang, Jun Xiao, Xiangjun Fan, Aashu Singh, Dina Katabi, et al. Dream: Where visual understanding meets text-to-image generation. arXiv preprint arXiv:2603.02667, 2026.

[34] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2142–2152, 2023.

[35] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. arXiv preprint arXiv:2511.13720, 2025.

[36] Tianhong Li, Qinyi Sun, Lijie Fan, and Kaiming He. Fractal generative models. arXiv preprint arXiv:2502.17437, 2025.

[37] 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.

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In ECCV, 2014.

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

[40] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In ICLR, 2023.

[41] Zeyu Lu, ZiDong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and LEI BAI. Fit: Flexible vision transformer for diffusion model. In ICML, 2024.

[42] Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding. arXiv preprint arXiv:2502.20321, 2025.

[43] 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. arXiv preprint arXiv:2401.08740, 2024.

[44] Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. DeCo: Frequency-decoupled pixel diffusion for end-to-end image generation. arXiv preprint arXiv:2511.19365, 2025.

[45] AI Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, april 2025, 2025.

[46] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 65(1):99–106, 2021.

[47] C. Nash, J. Menick, S. Dieleman, and P. W Battaglia. Generating images with sparse representations. arXiv preprint arXiv:2103.03841, 2021.

[48] Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. Signals & systems. Pearson Educación, 1997.

[49] 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.

[50] William Peebles and Saining Xie. Scalable diffusion models with transformers. In ICCV, 2023.

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

[52] 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 ICML, 2021.

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

[54] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.

[55] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X Chen. Improved techniques for training gans. NeurIPS, 2016.

[56] 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.

[57] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. arXiv preprint arXiv:2601.03267, 2025.

[58] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In International Conference on Machine Learning, pages 32211–32252. PMLR, 2023.

[59] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. NeurIPS, 2019.

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

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

[62] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. arXiv preprint arXiv:2406.06525, 2024.

[63] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. arXiv preprint arXiv:2307.05222, 2023.

[64] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. arXiv preprint arXiv:2503.19786, 2025.

[65] 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.

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

[67] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. arXiv preprint arXiv:2409.18869, 2024.

[68] Zidong Wang, Lei Bai, Xiangyu Yue, Wanli Ouyang, and Yiyuan Zhang. Native-resolution image synthesis. Advances in Neural Information Processing Systems, 38:14812–14848, 2026.

[69] ZiDong Wang, Zeyu Lu, Di Huang, Cai Zhou, Wanli Ouyang, and LEI BAI. Fitv2: Scalable and improved flexible vision transformer for diffusion model. arXiv preprint arXiv:2410.13925, 2024.

[70] Zidong Wang, Yiyuan Zhang, Xiaoyu Yue, Xiangyu Yue, Yangguang Li, Wanli Ouyang, and Lei Bai. Transition models: Rethinking the generative learning objective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 29178–29189, 2026.

[71] Hongyang Wei, Hongbo Liu, Zidong Wang, Yi Peng, Baixin Xu, Size Wu, Xuying Zhang, Xianglong He, Zexiang Liu, Peiyu Wang, et al. Skywork unipic 3.0: Unified multi-image composition via sequence modeling. arXiv preprint arXiv:2601.15664, 2026.

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

[73] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. arXiv preprint arXiv:2409.04429, 2024.

[74] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: a simple framework for masked image modeling. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9643–9653, 2021.

[75] Wanghan Xu, Xiaoyu Yue, Zidong Wang, Yao Teng, Wenlong Zhang, Xihui Liu, Luping Zhou, Wanli Ouyang, and Lei Bai. Exploring representation-aligned latent space for better generation. arXiv preprint arXiv:2502.00359, 2025.

[76] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115, 2024.

[77] Jingfeng Yao, Yuda Song, Yucong Zhou, and Xinggang Wang. Towards scalable pre-training of visual tokenizers for generation. arXiv preprint arXiv:2512.13687, 2025.

[78] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. arXiv preprint arXiv:2501.01423, 2025.

[79] 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.

[80] Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqi Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. ArXiv, abs/2511.20645, 2025.

[81] Xiaoyu Yue, Zidong Wang, Zeyu Lu, Shuyang Sun, Meng Wei, Wanli Ouyang, Lei Bai, and Luping Zhou. Diffusion models need visual priors for image generation. arXiv preprint arXiv:2410.08531, 2024.

[82] Xiaoyu Yue, Zidong Wang, Yuqing Wang, Wenlong Zhang, Xihui Liu, Wanli Ouyang, Lei Bai, and Luping Zhou. Understand before you generate: Self-guided training for autoregressive image generation. arXiv preprint arXiv:2509.15185, 2025.

[83] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. arXiv preprint arXiv:2510.11690, 2025.

[84] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. Transactions on Machine Learning Research, 2023.

[85] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. International journal of computer vision, 127(3):302–321, 2019.

Appendix

We include additional derivations, experimental details, and results in the appendix.

- In Sec. A, we provide a detailed derivation of Hybrid Signal-to-Noise Ratio (HSNR).
- In Sec. B, we we discuss more related works, including generative models with external representation and masked image modeling for image generation.
- In Sec. C, we provide more implementation details of experiments, including model architecture, training procedure, and downstream perception evaluations.
- In Sec. D, we provide additional empirical results, including HSNR comparisons, the gradient conflict analysis and structure target analysis.
- In Sec. E, we compare the feature map of VUM with MAE and JiT.
- In Sec. F, we provide more qualitative results of VUM.

A Mathematical Derivation of Hybrid SNR

In this section, we provide the rigorous mathematical derivations for the independent Signal-to-Noise Ratios (SNR) of continuous diffusion and discrete masking, and formally deduce our proposed Hybrid SNR (HSNR) formulation.

A.1 Preliminary Definitions

Let the original clean image be a random variable $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ drawn from the natural image distribution. For simplicity, we assume the data is pre-normalized to have zero mean and unit variance, such that the expected signal power is $\mathbb{E}[\mathbf{x}^2] = 1$. Let the continuous degradation be driven by Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with $\mathbb{E}[\boldsymbol{\epsilon}^2] = 1$. Let the spatial masking be defined by an independent Bernoulli random variable $\mathbf{m} \in \{0, 1\}$, where the probability of a patch being masked is r, and the probability of remaining visible is $1 - r$. Thus, the expectation of the mask indicator is $\mathbb{E}[\mathbf{m}] = 1 - r$, and the expectation of the masked region indicator is $\mathbb{E}[1 - \mathbf{m}] = r$.

A.2 Independent SNRs

Continuous Diffusion SNR. For a standard continuous noise schedule, the degraded state at trajectory step $t \in [0, 1]$ is defined as:

$$\mathbf{x}_t = (1 - t)\mathbf{x} + t\boldsymbol{\epsilon} \tag{8}$$

The preserved signal power $P_{\text{sig}}^{\text{diff}}$ and the additive noise power $P_{\text{noise}}^{\text{diff}}$ are calculated as:

$$P_{\text{sig}}^{\text{diff}} = \mathbb{E}[((1 - t)\mathbf{x})^2] = (1 - t)^2, \quad P_{\text{noise}}^{\text{diff}} = \mathbb{E}[(t\boldsymbol{\epsilon})^2] = t^2 \tag{9}$$

Thus, the diffusion SNR is strictly formulated as:

$$SNR_{\text{Diff}}(t) = \frac{P_{\text{sig}}^{\text{diff}}}{P_{\text{noise}}^{\text{diff}}} = \frac{(1 - t)^2}{t^2} \tag{10}$$

Discrete Masking SNR. For pure spatial masking, the observed state is $\mathbf{x}_{\text{mask}} = \mathbf{m} \odot \mathbf{x}$. The SNR is defined by the ratio of the visible signal power to the spatial information loss (which acts as the reconstruction uncertainty or "noise" from the perspective of the decoder):

$$P_{\text{sig}}^{\text{mask}} = \mathbb{E}[\mathbf{m}\mathbf{x}^2] = \mathbb{E}[\mathbf{m}]\mathbb{E}[\mathbf{x}^2] = 1 - r \tag{11}$$

$$P_{\text{loss}}^{\text{mask}} = \mathbb{E}[(1 - \mathbf{m})\mathbf{x}^2] = \mathbb{E}[1 - \mathbf{m}]\mathbb{E}[\mathbf{x}^2] = r \tag{12}$$

Consequently, the masking SNR is:

$$SNR_{\text{MAE}}(r) = \frac{P_{\text{sig}}^{\text{mask}}}{P_{\text{loss}}^{\text{mask}}} = \frac{1 - r}{r} \tag{13}$$

A.3 Hybrid SNR (Joint Degradation)

To model the unified continuum, we superimpose the spatial mask r onto the noisy continuous manifold $\mathbf{x}_t$. The joint degraded state is formulated as:

$$\mathbf{x}_{r,t} = \mathbf{m} \odot \mathbf{x}_t = \mathbf{m} \odot ((1-t)\mathbf{x} + t\boldsymbol{\epsilon}) \tag{14}$$

To derive the Hybrid SNR, we must calculate the joint preserved signal power and the aggregate uncertainty.

Joint Preserved Signal Power. The original signal $(1-t)\mathbf{x}$ is only preserved in the visible patches ($\mathbf{m} = 1$). Since the mask $\mathbf{m}$ and the signal $\mathbf{x}$ are independent, the expected preserved signal power is:

$$P_{\text{sig}}^{\text{hybrid}} = \mathbb{E}\left[\mathbf{m}((1-t)\mathbf{x})^2\right] = \mathbb{E}[\mathbf{m}] \cdot (1-t)^2 \cdot \mathbb{E}[\mathbf{x}^2] = (1-r)(1-t)^2 \tag{15}$$

Aggregate Uncertainty. The total uncertainty to the reconstruction objective stems from two mutually exclusive sources: the additive Gaussian noise present in the visible regions, and the complete structural signal loss in the masked regions.

- *Visible Continuous Noise:* The variance of the Gaussian noise within the unmasked patches is:
$$P_{\text{noise}}^{\text{visible}} = \mathbb{E}\left[\mathbf{m}(t\boldsymbol{\epsilon})^2\right] = \mathbb{E}[\mathbf{m}] \cdot t^2 \cdot \mathbb{E}[\boldsymbol{\epsilon}^2] = (1-r)t^2 \tag{16}$$
- *Discrete Spatial Loss:* The missing signal in the masked patches acts as absolute uncertainty. The variance of this completely lost signal is:
$$P_{\text{loss}}^{\text{spatial}} = \mathbb{E}\left[(1-\mathbf{m})\mathbf{x}^2\right] = \mathbb{E}[1-\mathbf{m}] \cdot \mathbb{E}[\mathbf{x}^2] = r \tag{17}$$

Because spatial masking creates disjoint sets of pixels, these two uncertainties aggregate linearly to form the total reconstruction noise power:

$$P_{\text{uncertainty}}^{\text{total}} = P_{\text{noise}}^{\text{visible}} + P_{\text{loss}}^{\text{spatial}} = (1-r)t^2 + r \tag{18}$$

Final Formulation. By dividing the joint preserved signal power by the aggregate uncertainty, we obtain the mathematically exact Hybrid SNR:

$$SNR_{\text{Hybrid}}(r,t) = \frac{P_{\text{sig}}^{\text{hybrid}}}{P_{\text{uncertainty}}^{\text{total}}} = \frac{(1-r)(1-t)^2}{(1-r)t^2 + r} \tag{19}$$

This formulation smoothly degenerates to $SNR_{\text{Diff}}(t)$ when $r = 0$, and to $SNR_{\text{MAE}}(r)$ when $t = 0$, mathematically proving that spatial masking and Gaussian noising exist on the exact same continuous degradation manifold.

B Extended Related Work

Generative Models with External Representations. Recent advancements have increasingly leveraged features from pre-trained Vision Foundation Models (VFMs) to augment diffusion training. For example, REPA [79] enhances semantic representation quality by enforcing the alignment of features between the early diffusion layers and the features of frozen vision. Based on this, REPA-E [32] extends the framework to support end-to-end VAE fine-tuning, further improving the alignment of features with VFM. REG [72] integrates spatial visual features with semantic class embeddings derived from VFMs, establishing a persistent semantic steering mechanism throughout the generative process. At the tokenizer level, ReaLS [75] and the VA-VAE used by LightningDiT [78] distill semantic priors from pre-trained VFMs into VAE encoders through representation alignment. Diffusion transformers are subsequently trained in the latent spaces produced by these aligned encoders. Despite their efficacy, these methodologies share a fundamental bottleneck: they strictly depend on external, off-the-shelf vision encoders to supply semantic priors. In contrast, our framework entirely discards the reliance on external representations; instead, by jointly optimizing for signal recovery and generation, we compel the model to intrinsically develop powerful semantic abstractions directly from raw visual signals.

Masked Image Modeling for Image Generation. Masked image modeling (MIM) [2, 74, 20] has been increasingly adapted for image synthesis. Early approaches operate in discrete token spaces; for instance, MaskGIT [5] employs a bidirectional transformer with an iterative parallel decoding strategy, while MAGE [34] utilizes variable masking ratios to train both generation and representation capabilities within a single framework. Moving beyond discrete tokenization, MAR [37] performs masked autoregressive modeling in continuous-valued spaces by integrating a diffusion loss. FractalMAR [36] further extends this to the pixel level through a recursive, divide-and-conquer fractal architecture. An orthogonal line of work integrates masking mechanisms into diffusion models. MaskDiT [84] applies an asymmetric encoder-decoder architecture to process only unmasked patches during diffusion training, and MDT [19] incorporates a mask latent modeling scheme to predict masked tokens from unmasked ones within the diffusion process. Unlike these approaches that either rely on discrete tokenizers or treat masking merely as an auxiliary mechanism, VUM fundamentally integrates masked image modeling and continuous diffusion into a joint signal recovery objective, achieving true perception-generation unification directly in the pure pixel space.

C More Implementation Details

Table 8: **Architectural and optimization details for VUM-B and VUM-L.** The framework employs a shared encoder paired with asymmetric decoupled decoders.

	VUM-B	**VUM-L**
Shared ViT Encoder ($\mathcal{E}$)		
Input resolution	$256 \times 256 \times 3$	$256 \times 256 \times 3$
Patch size	16×16	16×16
Num. layers	12	24
Hidden dim.	768	1024
Num. heads	12	16
Parameters	85.2M	302.5M
Heavy DiT Decoder ($\mathcal{D}_\mathbf{G}$)		
Num. layers	12	24
Hidden dim.	768	1024
Num. heads	12	16
Parameters	130.5M	458.1M
Auxiliary ViT Decoder ($\mathcal{D}_\mathbf{P}$)		
Num. layers	8	8
Hidden dim.	512	512
Num. heads	16	16
Parameters	26.1M	
Optimization		
Optimizer	Muon	
Learning rate	0.0002	
Learning rate Schedule	Constant with warmup	
Weight decay	0.01	
Batch size	2048	
Warmup epochs	40	
Training epochs	160	800

Model Architecture. We adopt the standard Vision Transformer (ViT) architecture for our shared encoder and auxiliary perceptual decoder, and the Diffusion Transformer (DiT) architecture for our heavy generative decoder. To incorporate conditional information during the generative phase, the DiT decoder utilizes adaptive layer normalization (adaLN) to condition the class embeddings. Instead of standard absolute positional embeddings, we apply 2D Rotary Position Embedding (2D RoPE) [61] across the network to better capture relative spatial relationships and enhance structural generalization.

Training. For training, we utilize the Muon optimizer. Data augmentation is kept strictly minimal; in contrast to the *RandomResizedCrop* typically employed by MAE, we uniformly apply a standard center-crop strategy alongside random horizontal flipping to process all input images. Crucially, to bridge the distribution gap with standard MAE inference, which operates on clean pixels without noise, we explicitly enforce that 25% of the images in every training batch are kept perfectly clean ($t = 0$) and subjected to a heavy 75% spatial mask. This guaranties that a consistent portion of the gradient updates is dedicated entirely to the perceptual abstraction objective.

Generation Sampling. For image synthesis, we employ an ODE-based sampler. We configure the generation process with 50 sampling steps and apply Classifier-Free Guidance (CFG) to enhance sample fidelity and class alignment. As CFG requires evaluating both the conditional and unconditional states at each step, this setup results in a total of 100 Number of Function Evaluations (NFE).

Linear Probing. Following the established protocols in MAE [20], we train a linear classifier on the frozen representations extracted from our shared ViT encoder. We optimize the linear head using the AdamW optimizer with a base learning rate of 0.1, a batch size of 16384, and a cosine decay schedule over 90 epochs (including 10 warmup epochs). We explicitly disable mixup, cutmix, drop path, and color jittering, and strictly set the weight decay to zero.

End-to-End Finetuning. For full fine-tuning, we update all model parameters following MAE practices [20]. We employ the AdamW optimizer with a base learning rate of 1e-3, a weight decay of 0.05, and a global batch size of 1024. The training spans 100 epochs for VUM-B and 50 epochs for VUM-L, utilizing a cosine learning rate schedule with 5 warmup epochs. To effectively adapt the pre-trained representations, we apply a layer-wise learning rate decay of 0.75 across the shared encoder. Standard augmentations and regularizations are adopted, including RandAug (9, 0.5), label smoothing (0.1), mixup (0.8), cutmix (1.0), and drop path (0.1).

Object Detection and Instance Segmentation on COCO. Following the adaptation strategy in MAE [20], we integrate our VUM ViT encoder as the backbone for Mask R-CNN. Strictly following the implementation in [9], we equally divide the Transformer blocks into four subsets and apply intermediate convolutions to extract multi-scale feature maps (strides 4, 8, 16, and 32) for the Feature Pyramid Network (FPN), fine-tuning the entire model for 12 epochs with a learning rate of 1.5×10^{-4}.

Semantic Segmentation on ADE20K. We evaluate dense semantic segmentation performance using the UperNet framework. Following the implementation in [9], the VUM ViT encoder serves as the backbone, and the entire network is fine-tuned end-to-end for 100 epochs with a batch size of 16 and a learning rate of 3.5×10^{-4}.

D Additional Results

D.1 HSNR versus Loss Balancing and Training Curricula

To examine whether adaptive loss weighting or generic training curricula can replace HSNR, we compare five strategies under a matched architecture and training budget. All strategies use a ViT-B encoder and a DiT-B decoder, without the auxiliary ViT decoder, and are trained on ImageNet for 160 epochs. The static baseline and HSNR correspond to *Config2* and *Config3* in Tab. 4, respectively.

- **Static mask and noise.** At each training step, we apply a fixed 75% random mask together with randomly sampled diffusion noise. This is the joint recovery baseline without information-balanced coupling.
- **Uncertainty weighting.** We retain the same static degradation as *Config2*, but replace the fixed task weighting in Eq. (7) with homoscedastic uncertainty weighting [29]. Specifically, we learn two scalars $s_{\mathrm{G}} = \log \sigma_{\mathrm{G}}^2$ and $s_{\mathrm{P}} = \log \sigma_{\mathrm{P}}^2$ and optimize

$$\mathcal{L}_{\mathrm{UW}} = \frac{1}{2} e^{-s_{\mathrm{G}}} \mathcal{L}_{\mathrm{G}} + \frac{1}{2} e^{-s_{\mathrm{P}}} \mathcal{L}_{\mathrm{P}} + \frac{1}{2} s_{\mathrm{G}} + \frac{1}{2} s_{\mathrm{P}}. \tag{20}$$

  The exponential terms adaptively weight the generative and perceptual losses, while the log-variance terms regularize these weights. This tests whether adaptive loss balancing alone can improve the static joint objective.

Table 9: **HSNR versus loss balancing and training curricula.** All strategies use a ViT-B encoder and a DiT-B decoder without an auxiliary ViT decoder, trained on ImageNet for 160 epochs. The static baseline and HSNR rows reuse *Config2* and *Config3* from Tab. 4.

Strategy	FID↓	IN1K-LP↑
Static 75% mask + noise	159.6	21.25
Uncertainty weighting	150.35	21.93
Progressive masking	52.47	23.37
Stage-wise (perception → generation)	27.36	12.63
HSNR (*Config3*)	**22.07**	**43.25**

- **Progressive masking.** The diffusion noise level t is sampled randomly, while the mask ratio is linearly annealed over training epochs from 0.75 to 0.25. This curriculum changes masking over the course of training without coupling it to the instantaneous noise level.
- **Stage-wise training.** We divide the same total epoch budget into two equal halves. The first half trains perception only, using a fixed 75% mask and no diffusion noise. The second half trains diffusion only, using randomly sampled noise and no masking. This tests a sequential perception-to-generation curriculum.
- **HSNR.** We couple the mask ratio to the instantaneous noise level using Eq. (3), guided by the Hybrid SNR formulation in Eq. (2). Unlike an epoch-based curriculum, this strategy adjusts spatial masking to the continuous corruption level throughout joint training.

Analysis. As shown in Tab. 9, uncertainty weighting provides limited improvements over the static baseline: FID changes from 159.6 to 150.35, and IN1K-LP from 21.25 to 21.93. Adaptive loss weighting therefore does not recover the performance of HSNR under this setting. Progressive masking improves FID to 52.47, but its LP remains 23.37. Stage-wise training achieves a lower FID of 27.36, while its LP falls to 12.63, indicating a weaker final perceptual representation. Among these five strategies, HSNR achieves both the lowest FID (22.07) and the highest LP (43.25). These results support coupling input corruption to the current noise level, beyond reweighting task losses or scheduling them across epochs. With the auxiliary ViT decoder added, the full VUM further achieves FID 9.77 and IN1K-LP 48.70, as reported in Tab. 4.

D.2 Gradient Conflict Analysis

To quantify the optimization interference between the $\mathcal{L}_{\text{P}}$ and $\mathcal{L}_{\text{G}}$ objectives in Eq. (7), we measure gradient conflict in the shared parameters. Given a mini-batch, we compute the MAE loss $\mathcal{L}_{\text{P}}$ and diffusion loss $\mathcal{L}_{\text{G}}$ separately, obtaining their respective gradients $\mathbf{g}_{\text{P}} = \nabla\theta\mathcal{L}_{\text{P}}$ and $\mathbf{g}_{\text{G}} = \nabla_\theta\mathcal{L}_{\text{G}}$ for each parameter group θ. We adopt two complementary metrics to evaluate the conflicts.

Gradient Cosine Similarity. We measure the cosine similarity between the two gradient vectors:

$$\text{CosSim}(\mathbf{g}_{\text{P}}, \mathbf{g}_{\text{G}}) = \frac{\mathbf{g}_{\text{P}} \cdot \mathbf{g}_{\text{G}}}{|\mathbf{g}_{\text{P}}| \cdot |\mathbf{g}_{\text{G}}|} \tag{21}$$

which measures the overall directional alignment between the two gradient vectors. A positive value indicates cooperative optimization, zero indicates orthogonality, and a negative value indicates conflicting objectives.

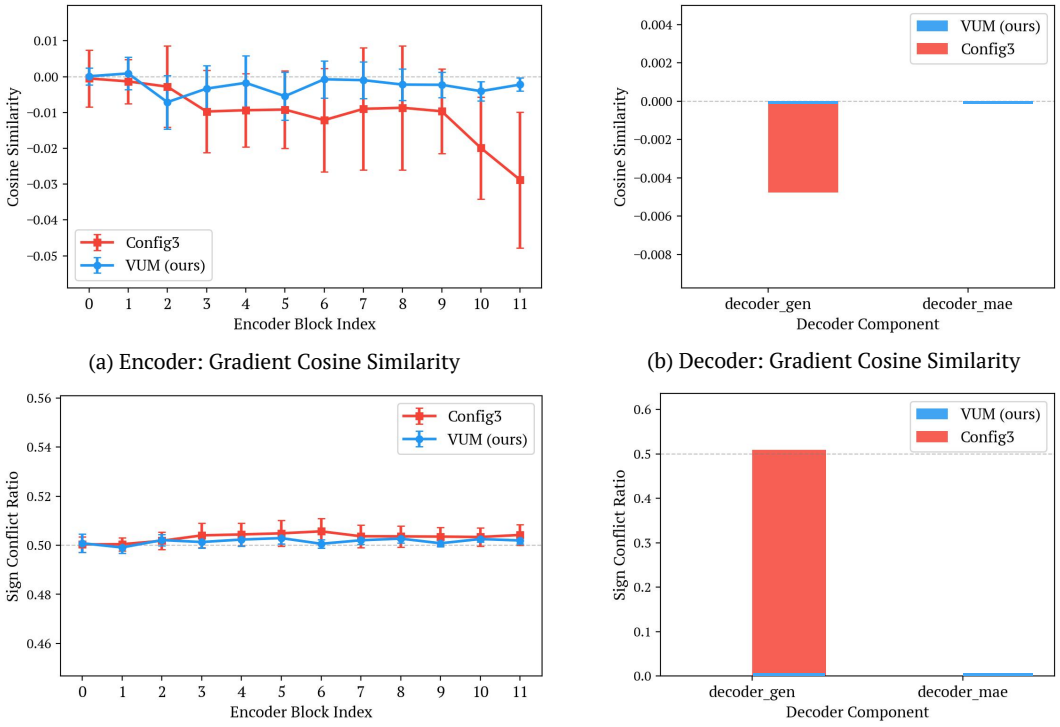
Sign Conflict Ratio. The sign conflict ratio is defined as the fraction of parameter elements (among those with magnitude above the median threshold for both gradients) that receive opposing gradient directions from the two tasks. To suppress noise from near-zero gradient elements, we define the significant set $\mathcal{S} = \{i : |g_{\text{P}}^{(i)}| > \text{med}(|\mathbf{g}_{\text{P}}|) \wedge |\mathbf{g}_{\text{G}}^{(i)}| > \text{med}(|\mathbf{g}_{\text{diff}}|)\}$, where $\text{med}(\cdot)$ denotes the element-wise median. The sign conflict ratio is then:

$$\text{SCR} = \frac{\left|\{i \in \mathcal{S} : \text{sign}(g_{\text{P}}^{(i)}) \neq \text{sign}(g_{\text{G}}^{(i)})\}\right|}{|\mathcal{S}|} \tag{22}$$

This metric captures the fraction of significant parameters that receive opposing update directions from the two tasks.

Analysis. We analyze our VUM-B with its single-decoder counterpart (i.e., *Config 3* in Tab. 4). As shown in Fig. 6 (a) and (c), the single-decoder baseline suffers from severe optimization interference within the shared encoder. Specifically, its cosine similarity progressively drops into the negative regime, and the SCR consistently exceeds 0.5, with the conflict peaking in the deeper encoder layers (blocks 8–11). This demonstrates that the perception and generation objectives actively pull the shared representations in opposing directions.

In stark contrast, VUM successfully neutralizes this interference, maintaining near-perfect gradient orthogonality (cosine similarity ≈ 0 and SCR ≈ 0.5) across all encoder blocks. The decoder-level analysis (b, d) exposes the mechanical root of this success: while the unified decoder in *Config 3* faces catastrophic gradient collisions (massive negative cosine similarity and high SCR), VUM's decoupled dual-decoders physically isolate the backward passes. Because $\mathcal{L}_{\text{P}}$ and $\mathcal{L}_{\text{G}}$ only route through their dedicated decoders ($\mathcal{D}_{\text{P}}$ and $\mathcal{D}_{\text{G}}$), the parameter-level gradient overlap is strictly zero. This confirms our architectural motivation: the decoupled decoders function as highly effective structural firewalls, completely eradicating task-level gradient conflicts before they can pollute the shared encoder.



(a) Encoder: Gradient Cosine Similarity

(b) Decoder: Gradient Cosine Similarity

(c) Encoder: Sign Conflict Ratio

(d) Decoder: Sign Conflict Ratio

Figure 6: **Gradient conflict analysis between perception and generation objectives.** We compare our decoupled VUM architecture against a single-decoder baseline (*Config 3* in Tab. 4) across the shared encoder layers (a, c) and the decoder components (b, d). By physically decoupling the decoding phase, VUM perfectly isolates task-specific gradients in the decoders, thereby shielding the shared encoder and maintaining conflict-free optimization.

D.3 Extra Ablations

Across the stochastic continuum ($t > 0$), a heavy DiT drives the spatial synthesis. To handle variable masking at arbitrary noise levels, $\mathcal{D}_{\text{G}}$ is governed by a state-conditioned regional objective (Eq. (6)): it applies standard denoising to the visible regions while actively reconstructing the *noisy* structure for the masked regions. Crucially, by targeting the noisy rather than the clean structure, the decoder natively preserves spatial coherence within the current temporal state t. The necessity of this trajectory-aligned target is explicitly ablated in Tab. 10.

Table 10: **Ablation on the masked region reconstruction target during the generative phase ($t > 0$).** Forcing the DiT decoder to predict the clean structure (Eq. 23) severely degrades generation fidelity and representation quality. In contrast, targeting the noisy structure (Eq. 6) intrinsically aligns the masked latents with the current stochastic trajectory, yielding drastically superior performance across all metrics.

Mask Region Target	FID↓	IS↑	IN1K-FT↑	IN1K-LP↑
Clean Structure Eq. (23)	16.18	131.54	80.57	41.47
Noisy Structure Eq. (6)	9.77	193.51	81.79	48.70

Compared to the default loss Eq. (6), its clean structure variant is defined as follows:

$$\mathcal{L}_{\mathrm{G}} = \mathbb{E}_t \left[\|(\boldsymbol{x} - \mathcal{D}_{\mathrm{G}}([\boldsymbol{h}, \boldsymbol{s}], t)) \odot (1 - \boldsymbol{M})\|_2^2 + \lambda \|(\texttt{Norm}(\boldsymbol{x}) - \mathcal{D}_{\mathrm{G}}([\boldsymbol{h}, \boldsymbol{s}], t)) \odot \boldsymbol{M}\|_2^2 \right], \quad (23)$$

Analysis As detailed in Tab. 10, the choice of reconstruction target for the masked regions during the generative phase ($t > 0$) profoundly impacts the network's capabilities. We observe a severe degradation in generative performance when $\mathcal{D}_{\mathrm{G}}$ is forced to predict the clean structure, with FID surging from 9.77 to 16.18 and IS plummeting from 193.51 to 131.54. Furthermore, the perceptual metrics (IN1K-FT and IN1K-LP) also suffer noticeable declines.

This stark performance gap is rooted in the physical consistency of the degradation trajectory. When targeting the clean structure (Eq. (23)), the network is compelled to execute two mathematically conflicting operations within the exact same feature map: standard incremental denoising for visible patches at timestep t, and a premature, full-trajectory projection to $t = 0$ for the masked patches. This severe spatial inconsistency essentially fractures the unified degradation continuum.

Conversely, targeting the noisy structure (Eq. (6)) guarantees that the reconstructed masked latents remain safely anchored to the exact same stochastic trajectory as the visible patches. By maintaining this strict spatial and temporal coherence at any given state t, VUM profoundly stabilizes the generative optimization. This trajectory-aligned strategy is the key to unlocking the model's full potential, delivering a substantial boost in both high-fidelity synthesis and discriminative abstraction.

E Feature Analysis

We present the PCA visualization of the feature maps in Figs. 7 to 10. We use MAE-B, JiT-B (24 layers) and our VUM-B (i.e., the three models in Tab. 4). MAE-B is evaluated exclusively on clean images ($t = 0$), whereas JiT-B and VUM-B are evaluated on varying noise levels $t \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$. For all models, the features are extracted from the layers $\{0, 1, 2, 3, 4, 5, 6, 7, 9, 11\}$. Compared to JiT-B, which produces noisy feature maps, VUM-B consistently produces cleaner and more structured representations. These discriminative hidden states directly benefit both generation quality and downstream perception performance.

F Qualitative Results

We present uncurated qualitative results generated by VUM-L in Figs. 11 to 14. All images are synthesized using 100 ODE sampling steps with a classifier-free guidance (CFG) scale of 2.5.

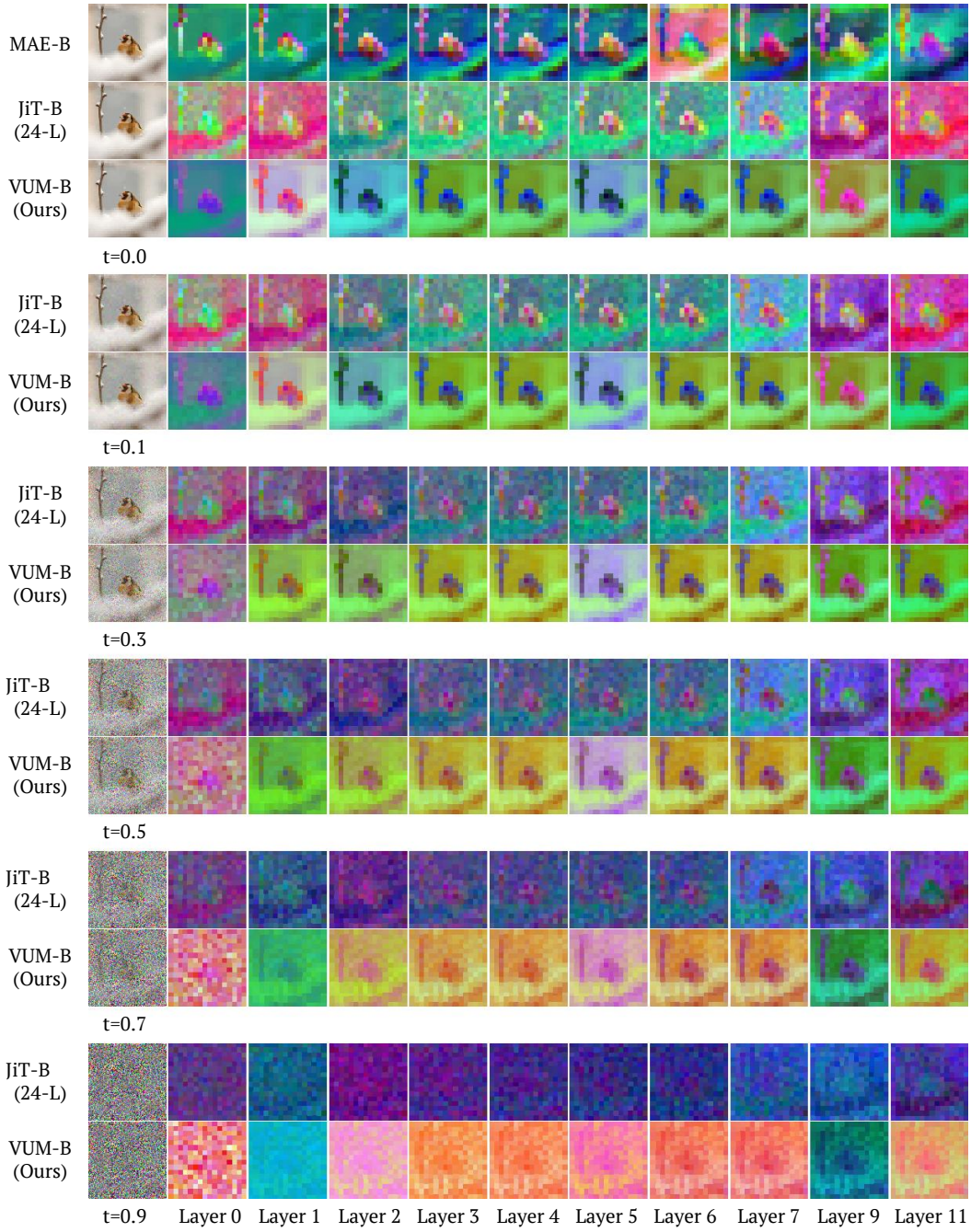


Figure 7: PCA feature map comparison among MAE-B, JiT-B (24 layers), and VUM-B. MAE-B is evaluated exclusively on clean images ($t = 0$), whereas JiT-B and VUM-B are evaluated across varying noise levels $t \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$. For all models, features are extracted from layers $\{0, 1, 2, 3, 4, 5, 6, 7, 9, 11\}$.

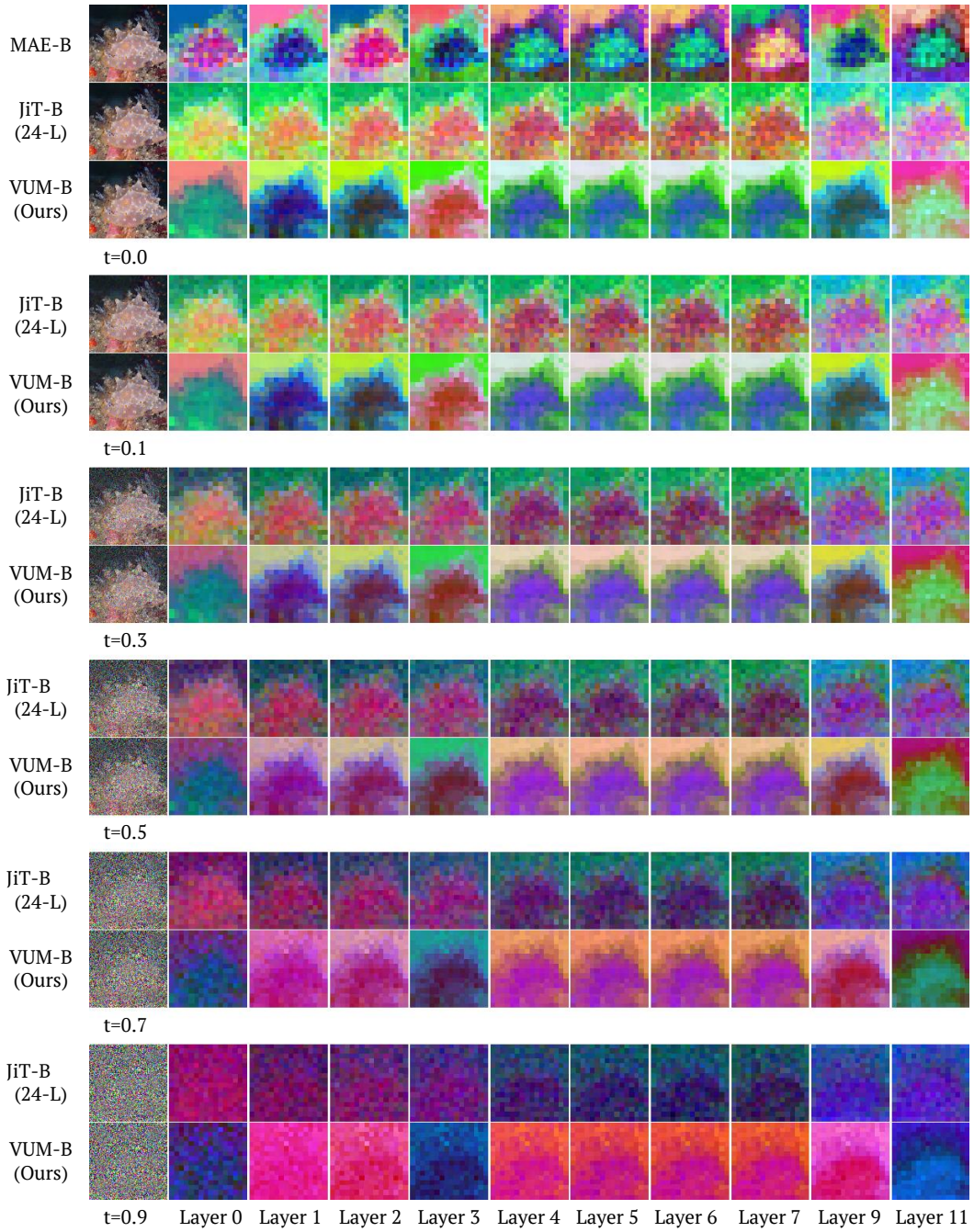


Figure 8: PCA feature map comparison among MAE-B, JiT-B (24 layers), and VUM-B. MAE-B is evaluated exclusively on clean images ($t = 0$), whereas JiT-B and VUM-B are evaluated across varying noise levels $t \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$. For all models, features are extracted from layers $\{0, 1, 2, 3, 4, 5, 6, 7, 9, 11\}$.

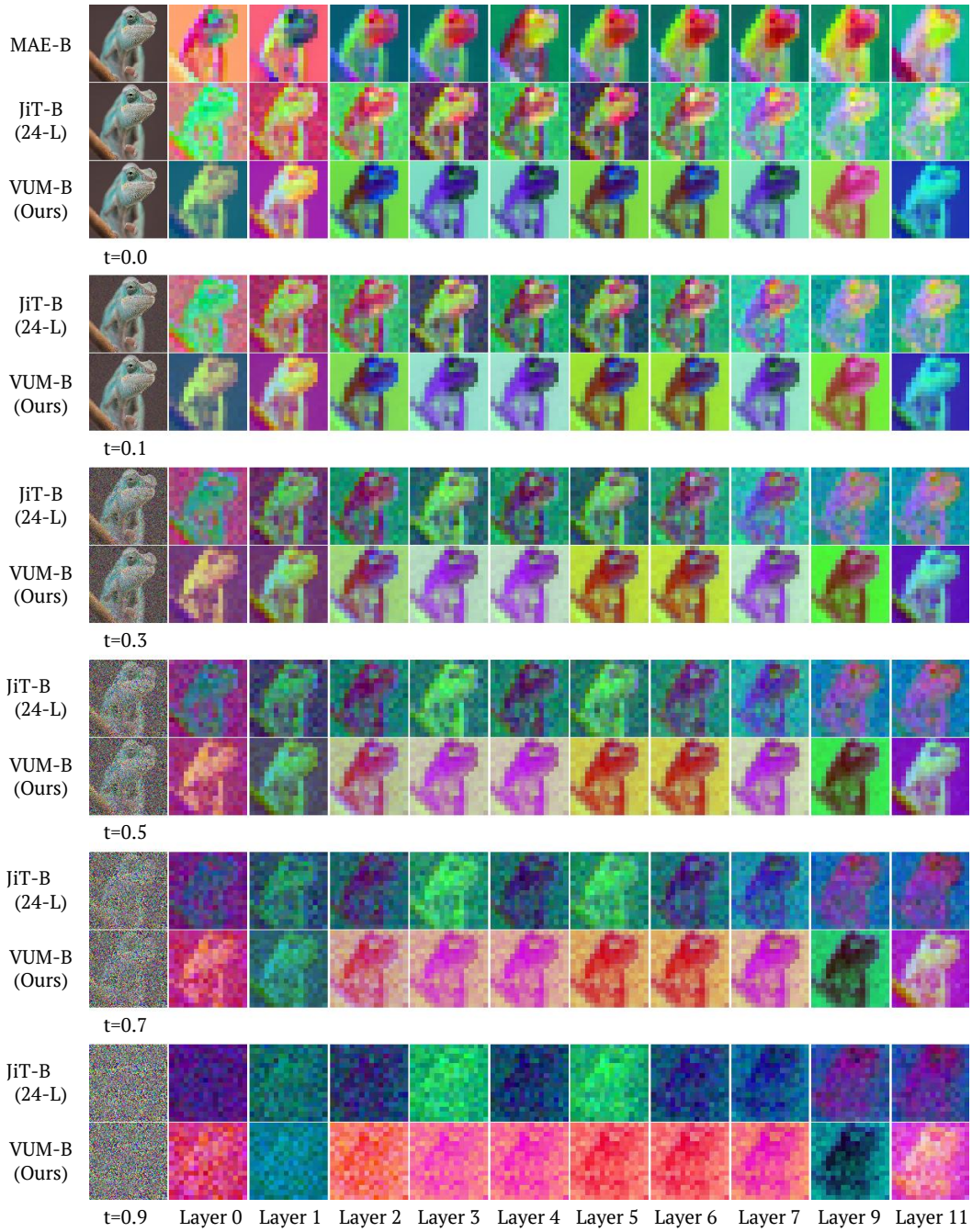


Figure 9: PCA feature map comparison among MAE-B, JiT-B (24 layers), and VUM-B. MAE-B is evaluated exclusively on clean images ($t = 0$), whereas JiT-B and VUM-B are evaluated across varying noise levels $t \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$. For all models, features are extracted from layers $\{0, 1, 2, 3, 4, 5, 6, 7, 9, 11\}$.

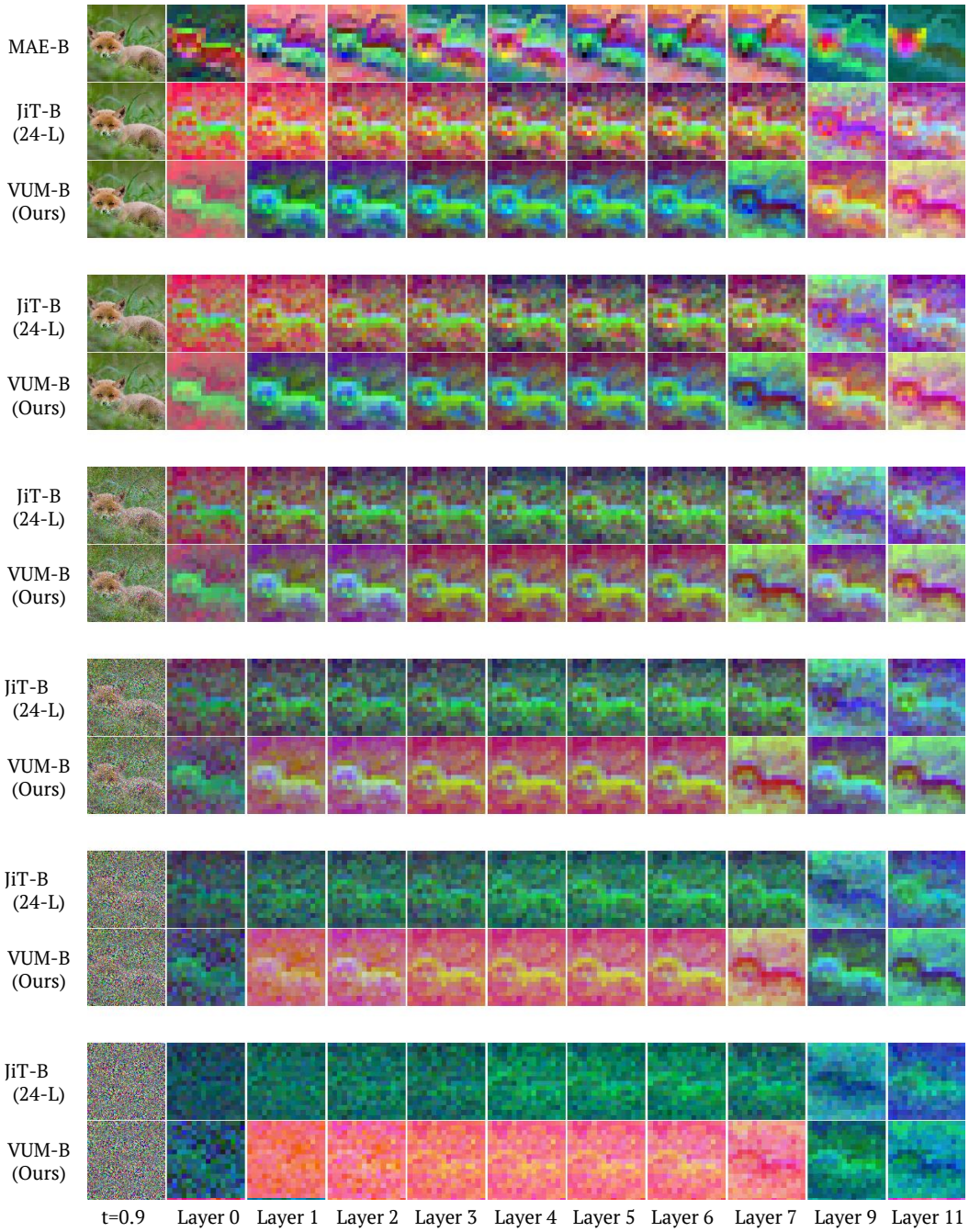


Figure 10: PCA feature map comparison among MAE-B, JiT-B (24 layers), and VUM-B. MAE-B is evaluated exclusively on clean images ($t = 0$), whereas JiT-B and VUM-B are evaluated across varying noise levels $t \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$. For all models, features are extracted from layers $\{0, 1, 2, 3, 4, 5, 6, 7, 9, 11\}$.



Class 12: house finch, linnet, Carpodacus mexicanus



Class 13: junco, snowbird



Class 19: chickadee



Class 35: mud turtle



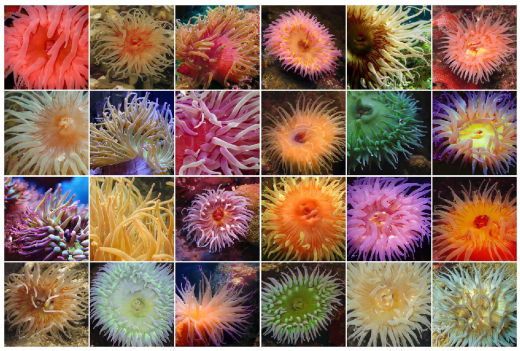
Class 81: boa constrictor, Constrictor constrictor



Class 88: macaw



Class 99: goose



Class 108: sea anemone, anemone

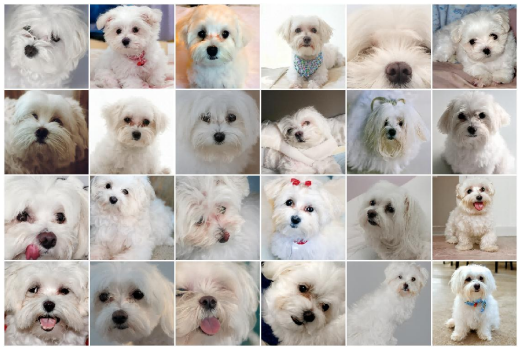
Figure 11: Uncurated, class-conditional samples on ImageNet 256×256 using VUM-L.



Class 140: red-backed sandpiper, dunlin, Erolia alpina



Class 14: pelican



Class 153: Maltese dog, Maltese terrier, Maltese



Class 207: golden retriever



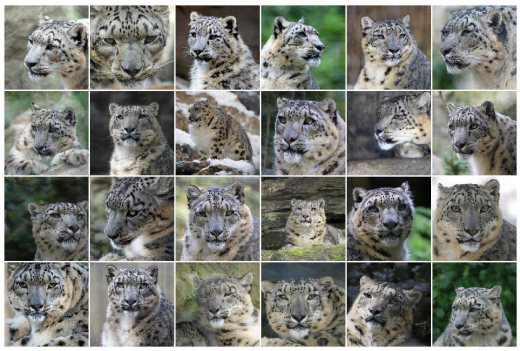
Class 250: Siberian husky



Class 270: white wolf, Arctic wolf, Canis lupus tundrarum



Class 279: Arctic fox, white fox, Alopex lagopus



Class 289: snow leopard, ounce, Panthera uncia

Figure 12: Uncurated, class-conditional samples on ImageNet 256×256 using VUM-L.



Class 291: lion, king of beasts, Panthera leo



Class 309: bee



Class 355: llama



Class 387: lesser panda, red panda



Class 388: giant panda, panda, panda bear, coon bear



Class 437: beacon, lighthouse, beacon light, pharos



Class 483: castle



Class 555: fire engine, fire truck

Figure 13: Uncurated, class-conditional samples on ImageNet 256×256 using VUM-L.



Class 562: fountain



Class 649: megalith, megalithic structure



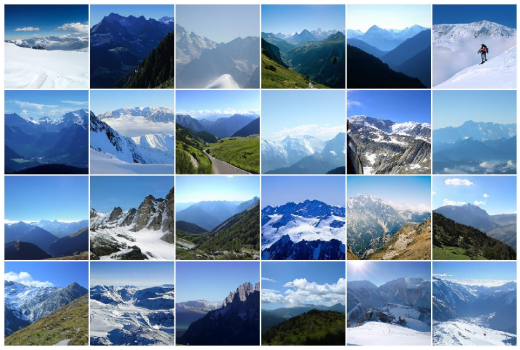
Class 698: palace



Class 927: trifle



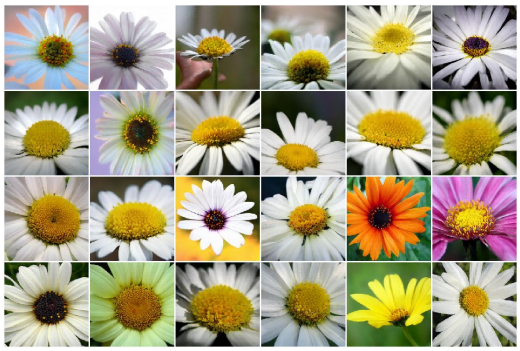
Class 388: giant panda, panda, panda bear, coon bear



Class 970: alp



Class 973: coral reef



Class 985: daisy

Figure 14: Uncurated, class-conditional samples on ImageNet 256×256 using VUM-L.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We provide detailed experimental proofs to support our claims.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations in the final sections.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: This paper does not include new theorems or theoretical assumptions.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide enough information and implementation details to reproduce the experimental results.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We use the public-available data only.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide enough training and test details.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report detailed experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the information of our compute resources.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification:We respect and follow the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This paper is focused on foundation model design, which does not involve such problems.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper is focused on foundation model design, which does not involve such problems.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We respect the credit and license, terms of existing code, data, models used in our research.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: Our research is focused on the network architecture and training paradigm design.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Not involved.Not involved.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Not involved.

    Guidelines:

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

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLM is only a tool in our research.

    Guidelines:

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