LatentUM: Unleashing the Potential of Interleaved Cross-Modal Reasoning via a Latent-Space Unified Model

Jiachun Jin[1], **Zetong Zhou**[1], **Xiao Yang**[2], **Hao Zhang**[3],
Pengfei Liu[1], **Jun Zhu**[2], **Zhijie Deng**[1]
[1]Shanghai Jiao Tong University [2]Tsinghua University [3]UCSD

Abstract

Unified models (UMs) hold promise for their ability to understand and generate content across heterogeneous modalities. Compared to merely generating visual content, the use of UMs for interleaved cross-modal reasoning is more promising and valuable, e.g., for solving understanding problems that require dense visual thinking, improving visual generation through self-reflection, or modeling visual dynamics of the physical world guided by stepwise action interventions. However, existing UMs typically rely on pixel decoding as a bridge when their visual representations for understanding and generation are disjoint, which is both ineffective and computationally cumbersome. In this paper, we introduce **LatentUM**, a novel unified model that represents all modalities within a shared semantic latent space, eliminating pixel-space mediation for model-internal reasoning over generated visual states. By making generated visual tokens directly interpretable to the model itself, LatentUM enables flexible interleaved cross-modal reasoning and generation. Empirically, LatentUM achieves state-of-the-art performance on the Visual Spatial Planning benchmark, pushes the limits of visual generation through self-reflection and demonstrates world modeling capability by predicting future visual states in the shared semantic latent space. Code and models are available at https://github.com/SJTU-DENG-Lab/LatentUM.

1 Introduction

Multimodal intelligence entails the seamless generation of diverse modalities, including text, images, and videos, with unified models (UMs) as a promising avenue for realizing this [9, 12, 13, 63]. However, existing UM approaches remain focused on visual generation tasks, such as image generation [8, 20, 51, 79], image editing [13, 39, 43, 77], and video generation [2, 73, 81], leaving the broader potential of UMs for interleaved cross-modal reasoning largely underexplored. Besides, the performance of UMs on these tasks also falls short of that of task-specialized models [3, 69, 75].

This paper argues that the distinctive potential of UMs lies beyond standalone visual generation: they can serve as systems for interleaved cross-modal reasoning, e.g., solving visually grounded planning problems via step-by-step cross-modal reasoning, improving visual generation through self-reflection, or performing world modeling based on stepwise actions. These tasks are especially valuable for practical use, but place greater emphasis on semantic correctness than on pixel-level fidelity. However, existing UMs are poorly suited to this setting when they maintain distinct visual representations for understanding and generation, requiring pixel decoding to serve as a bridge step [22, 54]. Such pixel-space mediation introduces unnecessary codec bias and cross-modal misalignment [18, 87], leading to potentially degraded instruction-following ability on reasoning-centric tasks.

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

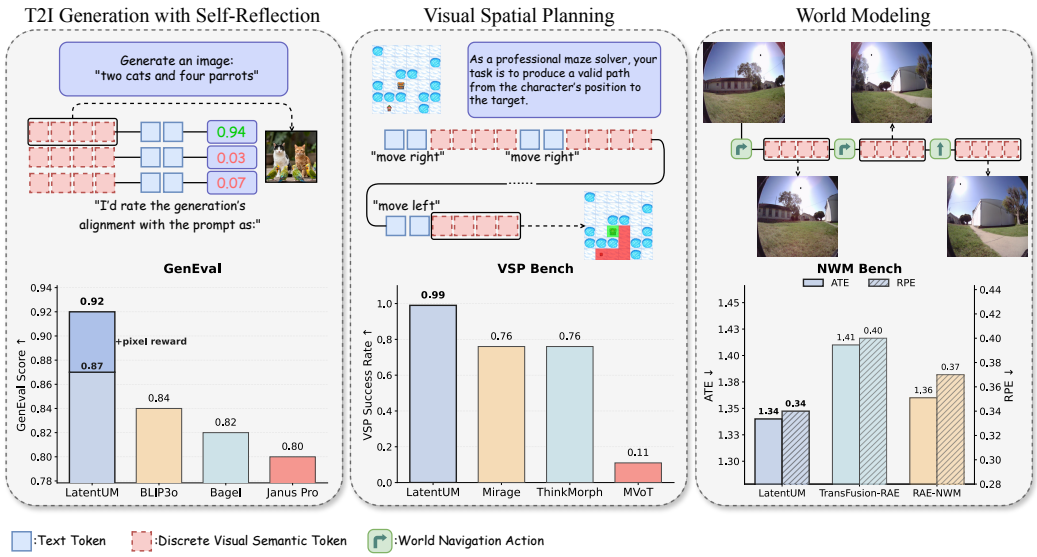


Figure 1: **LatentUM enables model-internal cross-modal reasoning in a shared semantic latent space. Left:** LatentUM improves text-to-image generation via self-reflection over its own generated semantic visual states. **Middle:** LatentUM interleaves textual reasoning with latent visual state predictions for visual spatial planning. **Right:** LatentUM predicts action-conditioned future visual latent states for world modeling.

We propose LatentUM, which embeds various modalities within a semantic latent space, for cross-modal reasoning. Motivated by the observation that reasoning-centric tasks emphasize semantic correctness over pixel-level fidelity, we advocate representing visual information as semantic tokens in the same space as language. CLIP-style visual features [56, 66] provide a natural semantic representation, but their continuous nature (in contrast to discrete language tokens) can raise modeling complexity (e.g., requiring the involvement of diffusion modeling [33]). We thus propose *model behavior aligned quantization* (MBAQ) to discretize CLIP features into discrete visual semantic tokens, whose goal is to preserve the resultant vision-language prediction ability rather than pixel-level reconstruction fidelity. As a result, generated visual tokens become directly interpretable by LatentUM itself, removing the need for pixel-space mediation during model-internal reasoning.

Given the unified representation, LatentUM can be instantiated as an autoregressive (AR) transformer, employing a simple next-token prediction objective. To preserve modality-specific capacity while enabling cross-modal interaction, we implement a *Mixture-of-Modal Experts* (MoME) architecture, where modality-specific transformer block parameters process each modality and shared self-attention enables information exchange [13, 17, 68]. Benefiting from this design, LatentUM can directly inherit the visual understanding capability from off-the-shelf vision-language models (VLMs) [3, 70] by utilizing them as parameter initialization. A diffusion decoder [17, 52] is additionally trained for pixel-space visualization when required, such as in text-to-image generation tasks.

We implement LatentUM$_{\text{Base}}$ from InternVL3.5-4B [70] and train its visual generation branch on 32M text-to-image pairs [8]. Empirically, LatentUM$_{\text{Base}}$ preserves strong visual understanding while achieving competitive visual generation performance among existing unified models. With task-specific post-training, LatentUM enables interleaved cross-modal reasoning for both visual generation with self-reflection and visually grounded planning, achieving state-of-the-art performance among unified models on GenEval [21], GenEval2 [30], and Visual Spatial Planning [78]. We additionally extend LatentUM to action-conditioned world modeling [4, 23], where future visual states are predicted as discrete semantic tokens within the shared latent space, yielding semantically coherent rollouts and competitive performance against specialized world-modeling baselines.

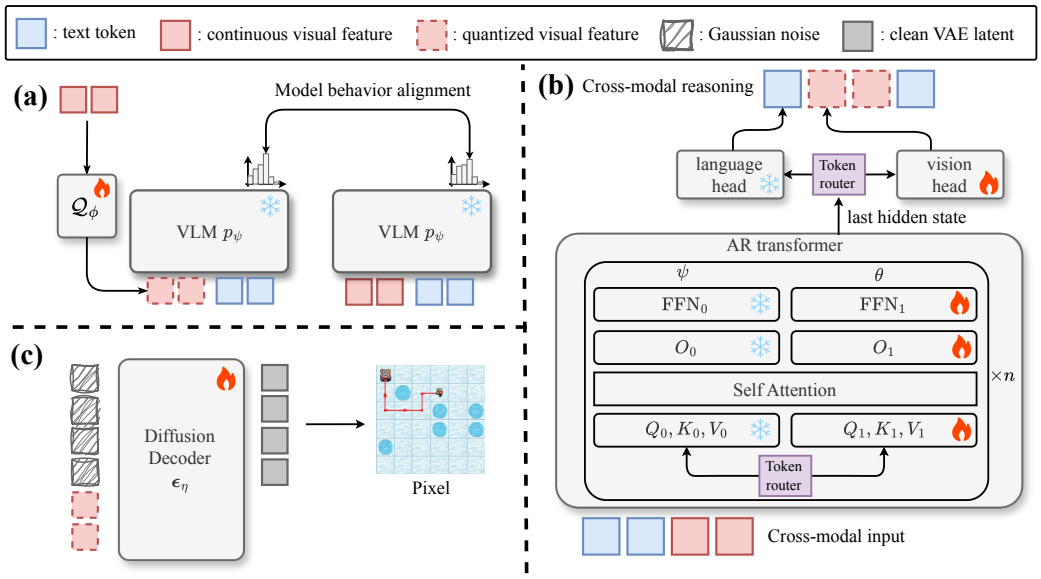


Figure 2: **Overview of LatentUM.** LatentUM unifies visual understanding and generation in a shared semantic latent space. **(a)** MBAQ converts CLIP features into discrete visual tokens by aligning with VLM behavior, preserving understanding-oriented semantics rather than pixel details. **(b)** Mixture-of-Modal Experts (MoME) decouples the understanding and generation branches while sharing self-attention for cross-modal reasoning. **(c)** An optional diffusion decoder renders semantic latents into pixels without training the core model for pixel reconstruction.

2 Method

Prevailing UMs cannot directly perform model-internal reasoning over generated visual states, because their generation and understanding pathways rely on separate visual representations [9, 13, 81]. Generated states therefore have to be decoded into pixels and re-encoded for understanding, introducing codec bias and cross-modal misalignment [87]. LatentUM eliminates this pixel-space mediation by representing both perceived and generated visual content within a shared semantic space. Figure 2 presents an overview of LatentUM, which comprises three components: (i) a visual tokenizer implemented via model behavior aligned quantization (MBAQ); (ii) Mixture-of-Modal Experts (MoME) with modality-specific branches and shared self-attention; and (iii) a decoupled pixel decoder for optional visualization.

2.1 Architecture

Visual Tokenizer. Unlike prior work that relies on reconstruction-oriented visual features [16, 31] learned via pixel space recovery, LatentUM needs to generate understanding-oriented features for effective model-internal cross-modal reasoning. A natural choice is to build upon CLIP features [56], which are inherently aligned with language through contrastive pretraining. This alignment enables visual tokens to participate directly in cross-modal reasoning without pixel-space mediation. Formally, let $\mathcal{I}$ denote an image and $\mathbf{V} \in \mathbb{R}^{d \times L}$ denote the corresponding CLIP features, where L is the sequence length and d is the channel dimension.

However, the generative modeling of continuous CLIP features is non-trivial due to their high dimensionality (e.g., d is typically on the order of thousands [89]). We can resort to extra diffusion modeling for such features on top of the autoregressive modeling of language, yet this can introduce substantial modeling and computational complexity. As a result, we decide to discretize $\mathbf{V}$ into discrete tokens for unified autoregressive modeling with language tokens. Specifically, we leverage multi-codebook quantization (MCQ) [49], which splits each d-dimensional token into C chunks and quantizes them independently with separate codebooks of size K. This yields an effective vocabulary of K^C from only $K \times C$ codebook entries, producing a discrete code matrix $\mathbf{Z} \in [K]^{C \times L}$.

Regarding the quantization objective, we propose *model behavior aligned quantization* (MBAQ). The key insight is that quantized features should preserve the visual understanding capacities of the original ones, rather than merely reconstructing the latter. To this end, we incorporate a VLM p_ψ that takes CLIP features as visual input, and enable it to also accept quantized visual features for visual understanding. Given a VQA dataset $\mathcal{D}_{\text{VQA}} = \{(\mathcal{I}^i, \mathbf{X}^i, \mathbf{Y}^i)\}_{i=1}^N$ with questions $\mathbf{X}^i$ and answers $\mathbf{Y}^i$ of length ℓ_i, we train the quantizer $\mathcal{Q}_\phi$ by minimizing the KL divergence between the VLM's output distributions on $\mathbf{V}$ versus $\mathcal{Q}_\phi(\mathbf{V})$, as illustrated in Figure 2(a). Specifically, the KL divergence at position j of sample i is:

$$\mathcal{L}_{\text{KL}}^{i,j} = \mathbb{D}_{\text{KL}} \left[p_\psi(\mathbf{Y}_j^i \mid \mathbf{Y}_{<j}^i, \mathbf{X}^i, \mathbf{V}^i) \parallel p_\psi(\mathbf{Y}_j^i \mid \mathbf{Y}_{<j}^i, \mathbf{X}^i, \mathcal{Q}_\phi(\mathbf{V}^i)) \right]. \tag{1}$$

We denote the de-quantized features as $\tilde{\mathbf{V}} = \mathcal{Q}_\phi(\mathbf{V}) \in \mathbb{R}^{d \times L}$, which serve as a semantic approximation of the original CLIP features $\mathbf{V}$. The full training objective averages this loss across all positions and samples, combined with the MCQ commitment loss $\mathcal{L}_{\text{MCQ}}$ weighted by λ.

Mixture-of-Modal Experts (MoME). Given such visual tokens, we aim to build a unified autoregressive model to characterize the dependencies within and across modalities. However, naively performing unified training on a Transformer backbone can introduce conflicting optimization signals that degrade performance, as the language and visual generation tasks serve distinct purposes [26, 40, 59].

We address this issue with a Mixture-of-Modal Experts (MoME) architecture, as illustrated in Figure 2(b). The model maintains two parallel branches within each transformer layer: the understanding branch ψ processes interleaved text and visual features for comprehension tasks, while the generation branch θ is dedicated to generating discrete visual tokens. Such a decoupling draws an analogy with the Mixture-of-Transformer (MoT) architecture [13].

Within each layer, the two branches share the self-attention mechanism while maintaining separate feed-forward networks and projection matrices. This shared attention enables cross-modal information flow, allowing the generation branch to leverage understanding context. A simple routing mechanism enables seamless switching between understanding and generation modes at inference time. Routing is determined by a special `<BOI>` token: tokens following `<BOI>` are processed by θ to produce discrete visual codes; otherwise, ψ handles language generation.

For the prediction head, we implement a lightweight causal transformer that predicts the C code indices sequentially within each visual token [49, 72]. This design captures the dependencies among code indices within the same token position, which a simple linear classification head would ignore [49]. We provide further details on the pretraining procedure in Section 2.2.

Decoupled Pixel Decoder. Although the visual features in LatentUM are optimized for semantic reasoning rather than pixel reconstruction, they can be rendered into images by a separate diffusion decoder $\boldsymbol{\epsilon}_\eta$. We adapt a pretrained text-to-image diffusion transformer [17, 52] by replacing its text conditioning with the quantized visual features $\tilde{\mathbf{V}}$, and train it with a standard conditional denoising objective:

$$\mathcal{L}_{\text{diff}}(\eta) = \mathbb{E}_{t, \boldsymbol{\epsilon}, \mathbf{x}_0} \left[\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\eta(\mathbf{x}_t, t, \tilde{\mathbf{V}}) \|^2 \right], \tag{2}$$

where $\mathbf{x}_t$ is the noisy latent at timestep t, and $\boldsymbol{\epsilon}$ is the added noise.

The decoder is trained independently, so the core model never optimizes for pixel reconstruction. As shown in Figure 3 (left), MBAQ features retain enough semantics for faithful rendering while discarding fine-grained pixel details less relevant to understanding, unlike VQVAE-style features.

2.2 Pre-training

For MBAQ training, we use the CLIP encoder of InternVL3.5-4B [66, 70] and train $\mathcal{Q}_\phi$ on LLaVA-v1.5-665K [45]. We instantiate $\text{LatentUM}_{\text{Base}}$ from InternVL3.5-4B, initialize the understanding branch ψ from its pretrained weights, thereby directly inheriting its strong visual understanding ability, and train only the generation branch θ on 32M BLIP3o image-text pairs [8] with a multi-code next-token prediction objective:

$$\mathcal{L}_{\text{MC-NTP}}(\theta) = -\frac{1}{L} \sum_{\ell=1}^{L} \frac{1}{C} \sum_{c=1}^{C} \log p_\theta(\mathbf{Z}_{c,\ell} \mid \mathbf{Z}_{<c,<\ell}, \mathbf{X}), \tag{3}$$

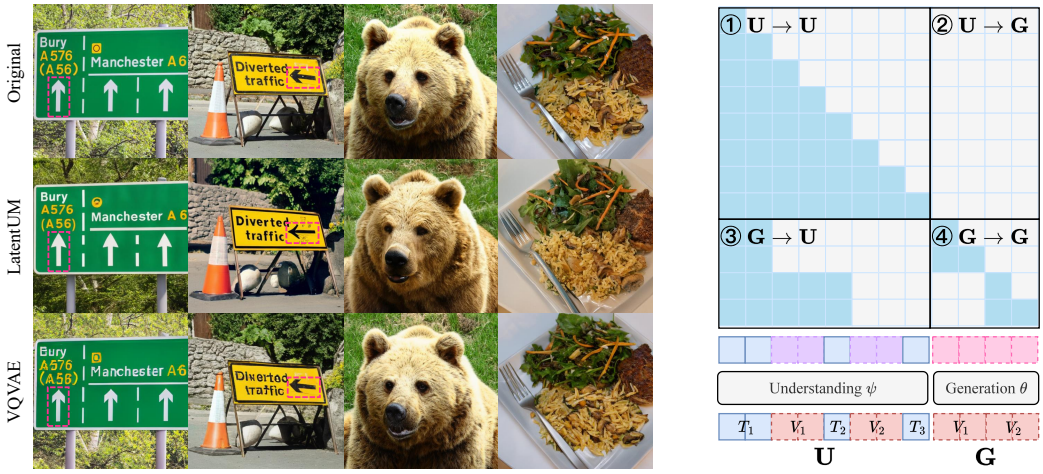


Figure 3: **Left: Pixel reconstruction comparison.** VQVAE preserves low-level details such as arrow styles but loses semantic content such as sign text, whereas our quantizer preserves semantics while discarding non-essential pixel details. **Right: Multi-frame interleaved training.** A dedicated attention mask routes visual tokens through both MoME branches, allowing all visual states to be trained in one forward pass.

where $\mathbf{Z}_{c,\ell}$ denotes the c-th code index of the ℓ-th visual token, and $\mathbf{X}$ denotes the conditioning context. Pre-training uses text-only contexts due to data availability, while interleaved cross-modal conditioning is activated during post-training.

We train the diffusion decoder $\boldsymbol{\epsilon}_\eta$ independently on the same 32M images by fine-tuning the condition branch of a pretrained dual-stream diffusion transformer [17]. Detailed training settings are provided in Appendix B.

2.3 Post-training

Although LatentUM$_{\text{Base}}$ possesses the architectural capacity for model-internal cross-modal reasoning, activating it requires task-specific post-training.

Supervised Fine-tuning. To enable *cross-modal chain-of-thought* reasoning, we jointly fine-tune the understanding branch ψ and generation branch θ on interleaved cross-modal sequences with a next-token prediction objective:

$$\mathcal{L}_{\text{SFT}}(\psi, \theta) = -\sum_{i=1}^{L} \log p_\Theta(\mathbf{o}_i \mid \mathbf{o}_{<i}, \mathbf{X}), \tag{4}$$

where $\Theta \in \{\psi, \theta\}$ selects the branch according to token type, $\mathbf{o}_i$ denotes the i-th token, and $\mathbf{X}$ is the interleaved multimodal context. Generated visual tokens are re-processed by ψ and cached as understanding context for subsequent tokens. During SFT, each visual segment is therefore routed through both θ and ψ with a parallel attention mask, enabling generation loss and understanding-context construction in one forward pass as illustrated in Figure 3 (right).

Reinforcement Learning. Given a prompt, LatentUM samples G cross-modal rollouts $\{\mathbf{O}_g\}_{g=1}^{G}$, where each response may contain both text and visual tokens. We can optimize LatentUM with the following GRPO [58] objective:

$$\mathcal{L}_{\text{GRPO}}(\psi, \theta) = \mathbb{E}_{\{\mathbf{O}_g\}} \left[\frac{1}{G} \sum_{g=1}^{G} \frac{1}{L_g} \sum_{i=1}^{L_g} \min\big(r_g^i \hat{A}_g,\ \text{clip}(r_g^i) \hat{A}_g\big) - \beta\, \mathbb{D}_{\text{KL}}[p_\Theta \| p_{\text{ref}}] \right]. \tag{5}$$

Here $\Theta \in \{\psi, \theta\}$ selects the branch according to token type, $r_g^i = p_\Theta(\mathbf{o}_i)/p_{\text{old}}(\mathbf{o}_i)$ is the clipped probability ratio, and $\hat{A}_g = (\mathcal{R}_g - \bar{\mathcal{R}})/\sigma_{\mathcal{R}}$ is the normalized group advantage. The reward $\mathcal{R}_g$ can be provided by external reward models or by the understanding branch ψ itself via *generate-then-reflect*. In the latter case, ψ acts as a semantic evaluator over generated visual features and returns a

scalar reward. Our experiments instantiate this reward with verification questions, as described in Section 3.2 and Appendix C.

3 Experiments

In this section, we first assess the basic visual understanding and generation capabilities of LatentUM$_{\text{Base}}$ in Section 3.1, then study LatentUM's model-internal cross-modal reasoning abilities in Section 3.2, extend the evaluation to action-conditioned world modeling in Section 3.3, and finally ablate the semantic latent-space design choices in Section 3.4. Detailed experimental setup is provided in Appendix B.1.

3.1 Base Model Capabilities



Figure 4: **Text-to-image gallery of LatentUM.** The last column shows legible in-image text enabled by the unified semantic space.

Table 1: Evaluation results on multimodal understanding benchmarks. † denotes evaluation with quantized visual features.

Model	MME-P ↑	POPE ↑	SEED ↑	MMB ↑	MMMU ↑
Tokenflow XL [55]	1551	87.8	72.6	76.8	43.2
Bagel [13]	1687	-	-	85.0	55.3
JanusFlow [50]	1333	88.0	70.5	-	-
Harmon [79]	1155	87.6	67.1	65.5	-
Chameleon [63]	1057	77.8	-	-	26.7
Orthus [33]	1266	79.6	-	-	28.2
EMU3 [71]	1244	85.2	68.2	58.5	-
VILA-U [80]	1402	85.8	59.0	-	-
Janus Pro [9]	1444	87.4	72.1	79.2	41.0
Show-o2 [81]	1620	-	69.8	79.3	48.9
BLIP3o 4B [8]	1528	-	73.8	78.6	46.6
BLIP3o 8B [8]	1683	-	77.5	83.5	50.6
InternVL3.5 4B [70]	1654	88.9	76.3	80.3	54.6
LatentUM$_{\text{Base}}$	1654	88.9	76.3	80.3	54.6
LatentUM$_{\text{Base}}$ †	1638	85.5	75.0	79.5	52.3

Table 2: Evaluation results on GenEval [21]. + *pixel-reward* denotes using an external pixel-reward model.

Model	Single	Two	Count	Color	Pos.	Color Attr.	Overall ↑
Specialized Text-to-Image Models							
FLUX.1 [Dev] [34]	0.98	0.81	0.74	0.79	0.22	0.45	0.66
SD3.5 Large [17]	0.98	0.89	0.73	0.83	0.34	0.47	0.71
Qwen-Image [75]	0.99	0.92	0.89	0.88	0.76	0.77	0.87
Unified Models							
TokenFlow-XL [55]	0.95	0.60	0.41	0.81	0.16	0.24	0.55
Transfusion [91]	-	-	-	-	-	-	0.63
JanusFlow [50]	0.97	0.59	0.45	0.83	0.53	0.42	0.63
Bagel [13]	0.99	0.94	0.81	0.88	0.64	0.63	0.82
Janus Pro [9]	0.99	0.89	0.59	0.90	0.79	0.66	0.80
Show-o2 [81]	1.00	0.87	0.58	0.92	0.52	0.62	0.76
BLIP3o 4B [8]	-	-	-	-	-	-	0.81
BLIP3o 8B [8]	-	-	-	-	-	-	0.84
LatentUM$_{\text{Base}}$	0.99	0.92	0.72	0.91	0.83	0.73	0.85
LatentUM$_{\text{Vis-Gen}}$	0.99	0.92	0.85	0.87	0.89	0.71	0.87
+ *pixel-reward*	0.99	0.95	0.95	0.92	0.92	0.81	0.92

Visual Understanding. We evaluate LatentUM$_{\text{Base}}$'s visual understanding on MME [19], POPE [42], SEED-Bench [37], MM-Bench [47], and MMMU [88], using both continuous and quantized visual

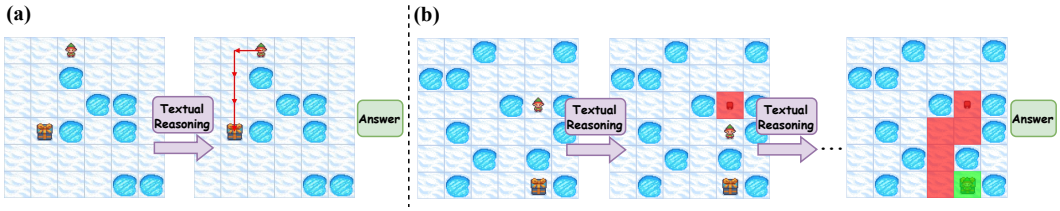


Figure 5: **Two interleaved reasoning paradigms for visual spatial planning. (a)** Coarse-grained: generates a visual plan from the maze layout before solving. **(b)** Fine-grained: alternates textual reasoning with visual state predictions to guide each action.

Table 3: Evaluation results on GenEval2 [30].

Model	TIFA_{GM}	TIFA_{AM}
Bagel + CoT [13]	23.1	70.9
Janus Pro [9]	14.5	58.3
BLIP3o 8B [8]	13.3	59.4
$\text{LatentUM}_{\text{Base}}$	21.4	68.1
$\text{LatentUM}_{\text{Vis-Gen}}$	31.3	72.9

Table 4: Evaluation results on VSP [78].

Model	L3	L4	L5	L6	Avg.
Anole [11]	0.02	0.01	0.00	0.00	0.01
MVoT [38]	0.21	0.11	0.08	0.03	0.11
Mirage [85]	0.93	0.83	0.76	0.51	0.76
Chameleon [63]	-	-	-	-	0.01
InternVL3.5 8B [70]	-	-	-	-	0.08
InternVL3.5 38B [70]	-	-	-	-	0.20
ThinkMorph [22]	-	-	-	-	0.76
$\text{LatentUM}_{\text{Vis-Plan}}$ (coarse)	1.00	0.85	0.83	0.71	0.85
$\text{LatentUM}_{\text{Vis-Plan}}$ (fine)	1.00	1.00	1.00	0.97	0.99

features. As shown in Table 1, $\text{LatentUM}_{\text{Base}}$ retains strong visual understanding by directly inheriting the frozen InternVL3.5 understanding branch. Quantization causes only modest degradation, indicating that MBAQ produces semantically meaningful representations, which is a prerequisite for effective cross-modal reasoning over self-generated visual content.

Visual Generation. Figure 4 shows semantically coherent text-to-image samples from $\text{LatentUM}_{\text{Base}}$, including emergent in-image text rendering. On GenEval [21], $\text{LatentUM}_{\text{Base}}$ achieves an overall score of 0.85, outperforming all unified baselines in Table 2. These results suggest that MBAQ-aligned visual features support strong instruction following in visual generation.

3.2 Cross-Modal Reasoning

Visual Generation with Self-Reflection. LatentUM can improve its visual generation quality through *generate-then-reflect*, a model-internal reasoning process where the model reflects on its own generated semantic visual states without decoding them back to pixels. Following the RL framework in Section 2.3, we post-train $\text{LatentUM}_{\text{Base}}$ with self-reward derived from its own understanding capability, yielding $\text{LatentUM}_{\text{Vis-Gen}}$.

For GenEval [21] and GenEval2 [30], we decompose each training prompt (disjoint from the evaluation prompts) into atomic visual checks, such as object existence, counts, colors, and spatial relations, and use the resulting verification questions to compute self-reward with the understanding branch directly over generated semantic visual features. This avoids decoding the generated visual state into pixels or invoking an external pixel-space reward model. We keep ψ fixed and fine-tune only the visual-generation pathway, decoupling reward evaluation from the learned policy. Detailed question templates, option scoring, reward aggregation, and GRPO settings are provided in Appendix C.

As shown in Table 2, $\text{LatentUM}_{\text{Vis-Gen}}$ achieves 0.87 on GenEval, outperforming all unified model baselines; an external GenEval pixel-reward model [46, 90] further raises the score to 0.92. This gap more likely reflects limitations of GenEval's detection-based metric than a weakness of model-internal self-reward, as valid generations can be penalized by detector false negatives; see Appendix C.4. In contrast, GenEval2 [30] employs MLLM-as-judge evaluation [6], which is more semantically robust; there, as shown in Table 3, $\text{LatentUM}_{\text{Vis-Gen}}$ outperforms all baselines by a large margin, confirming the effectiveness of model-internal self-reward.

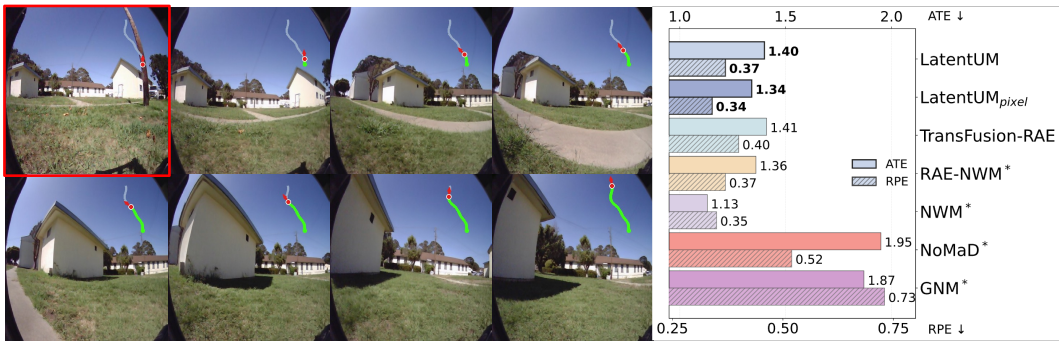


Figure 6: **Left:** Open-loop future-frame rollouts following a predefined action trajectory. The red box denotes the initial context frame, and subsequent frames are predicted future observations. **Right:** Quantitative comparison on the NWM benchmark [4]. * denotes specialist for navigation.

Interleaved Visual Reasoning for Spatial Planning. A key advantage of LatentUM's unified latent space is enabling model-internal reasoning over generated visual states to solve visual reasoning tasks. We evaluate this capability on the Visual Spatial Planning (VSP) benchmark [78], which requires spatial reasoning in a maze-navigation environment. We fine-tune LatentUM$_{\text{Base}}$ on task-specific data for each paradigm, yielding LatentUM$_{\text{Vis-Plan}}$. We investigate two reasoning paradigms that exploit this capability in progressively deeper ways, as illustrated in Figure 5.

Coarse-grained planning. This follows a text-visual-text reasoning flow: it first analyzes the maze layout in text, then generates a complete visual plan, and finally derives the textual solution from this plan. The generated plan serves as an intermediate visual reasoning state in the shared latent space. We use the 6,000 training samples from ThinkMorph [22] in the format of Figure 5(a).

Fine-grained step-by-step planning. This repeats a text-visual-text loop: each textual reasoning produces an updated visual state. This stepwise interleaving yields denser cross-modal context throughout the reasoning trajectory. We construct 18,000 filtered training samples in the format of Figure 5(b), with details provided in Appendix D.

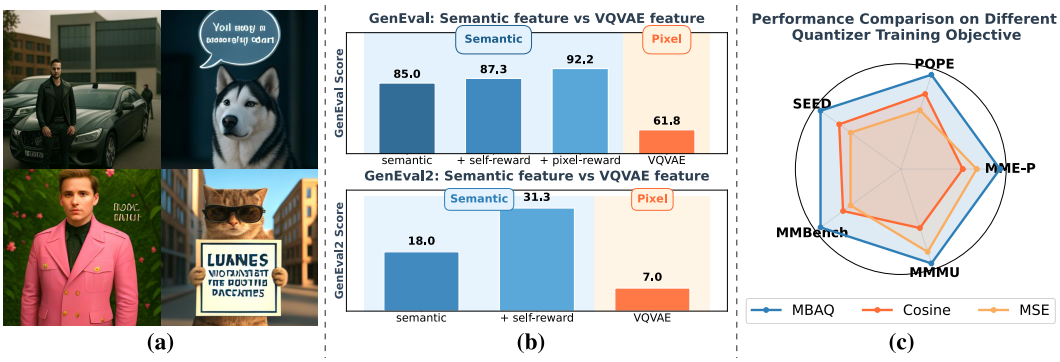
As reported in Table 4, LatentUM$_{\text{Vis-Plan}}$ outperforms prior visual reasoning models under both planning paradigms, including the directly comparable ThinkMorph baseline [22], which is fine-tuned from Bagel [13] with the same data and relies on pixel-space mediation. These gains validate our core hypothesis: unified semantic representations enable more effective model-internal cross-modal reasoning than pixel-space bridging, while the coarse-to-fine improvement shows that the shared latent space supports deeper interleaved reasoning for complex spatial planning.

3.3 World Modeling

Beyond reasoning over static generated visual states, we ask whether the same latent space can support temporally grounded reasoning: predicting future visual states conditioned on actions [23].

We adopt the Navigation World Model (NWM) [4] evaluation setting and fine-tune LatentUM$_{\text{Base}}$ on egocentric robot navigation dataset RECON [57], to obtain LatentUM$_{\text{WM}}$. Following Transfusion-RAE [65], we represent navigation actions as standard text tokens, seamlessly integrating into LatentUM's unified autoregressive formulation. Each training sample is constructed by conditioning on four context frames and a text-based navigation action to predict the discrete semantic tokens of the subsequent frame. During inference, LatentUM supports fully latent rollout by iteratively feeding predicted semantic codes as subsequent context. Since the NWM evaluation is sensitive to fine-grained pixel-space structure, we also evaluate a pixel-space recurrent interface that renders predicted semantic codes into images and re-encodes them at each step, while keeping latent semantic-token prediction as the learned dynamics model.

To validate the learned dynamics, we perform open-loop future-frame prediction: Figure 6 (left) shows temporally consistent rollouts that follow a predefined action trajectory while preserving the initial scene layout. Notably, due to the inherited language-vision alignment, LatentUM$_{\text{WM}}$ also exhibits an emergent capability of zero-shot language-conditioned navigation, enabling it to



(a) (b) (c)

Figure 7: **Ablation study.** **(a)** Text-to-image generation with the VQVAE baseline on selected prompts from Figure 4. **(b)** GenEval and GenEval2 comparison, showing stronger instruction following from semantically aligned features. **(c)** Visual understanding comparison across quantizer training objectives, where MBAQ consistently outperforms feature-reconstruction baselines.

natively follow free-form textual instructions, as detailed in Appendix F. Furthermore, we evaluate under the NWM zero-shot planning protocol. As shown in Figure 6 (right), fully latent rollout achieves 1.40 ATE / 0.37 RPE, while pixel-space recurrence improves this to 1.34 / 0.34, surpassing Transfusion-RAE [65] and remaining competitive with the specialized NWM baseline. These results suggest that the shared semantic latent space extends beyond static reasoning to action-conditioned future-state prediction, offering a preliminary step toward temporally grounded world modeling.

3.4 Ablation Studies

We conduct ablation studies to validate two core design choices in LatentUM: (i) using semantically aligned features instead of pixel-reconstruction features for visual generation, and (ii) training the quantizer with MBAQ rather than feature reconstruction objectives for visual understanding.

Semantic Features for Visual Generation. We first validate that semantically aligned features outperform pixel-reconstruction features for instruction-following in visual generation. We train an alternative model that differs from LatentUM *only* in the visual features: it adopts VQVAE features from LlamaGen [60], while architecture and training data remain identical. Figure 7(a) shows qualitative results. The VQVAE baseline generates plausible images but exhibits a clear quality gap compared to LatentUM, particularly in text rendering. Figure 7(b) quantifies this gap: LatentUM outperforms the baseline on both GenEval and GenEval2, confirming that semantically aligned visual features are crucial for instruction-following in visual generation.

MBAQ for Visual Understanding. We next investigate how different quantizer training objectives affect visual understanding. We compare MBAQ against two alternative quantizers that directly reconstruct continuous semantic features using either MSE loss or cosine similarity loss [24], sharing the same architecture. As shown in Figure 7(c), MBAQ consistently outperforms both reconstruction-based alternatives across five visual understanding benchmarks. This indicates that incorporating next-token prediction behavior during quantizer training is more effective than approaches that focus solely on feature reconstruction [24, 53].

4 Related Work

Unified models (UMs) process and create content across multiple modalities in one architecture. Early approaches enable multimodal generation by extending pretrained LLMs with discrete visual tokens [11, 33, 61, 63, 64, 71, 81, 91]. Representative works include Chameleon [63] and Emu3 [71] that employ unified transformers for next-token prediction across modalities, Show-o [81] and Transfusion [91] that combine autoregressive modeling with diffusion. More recent efforts explore unified architectures with improved training strategies [9, 12, 13, 50] and interleaved image-text generation capabilities [11, 26, 86]. Despite these advances, most UMs primarily focus on visual generation tasks such as text-to-image synthesis and image editing [34, 69, 75]. We discuss more related work in Appendix A.

5 Conclusion

We introduced LatentUM, a latent-space unified model in which generated visual tokens remain directly interpretable by the model for model-internal cross-modal reasoning. By combining MBAQ, MoME, and an optional pixel decoder, LatentUM avoids pixel-space mediation during reasoning while still supporting visualization. Empirically, LatentUM preserves strong visual understanding, outperforms unified baselines on GenEval and GenEval2, reaches state-of-the-art performance on Visual Spatial Planning, and shows preliminary action-conditioned world-modeling capability.

Several limitations remain. At the model level, LatentUM is trained at a modest scale and currently supports fixed-resolution generation. Moreover, although the MBAQ objective transfers across VLM families, the learned quantizer is backbone-specific and must be retrained for each target VLM. At the application level, self-reward for visual generation has so far been validated only with compositional verification questions, and fully latent long-horizon rollout in world modeling can be further strengthened for tasks requiring precise spatial fidelity. Future work should therefore scale the model and data, support variable-resolution and longer-context generation, design holistic rewards for open-ended generation, and further develop fully latent temporal prediction for planning.

Acknowledgments and Disclosure of Funding

This work was supported by the Shanghai Key Technology R&D Program "New Generation of Information Technology" (No. 25511103700), and the National Natural Science Foundation of China (Nos. 62306176 and 92470118).

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[2] Inclusion AI, Biao Gong, Cheng Zou, Chuanyang Zheng, Chunluan Zhou, Canxiang Yan, Chunxiang Jin, Chunjie Shen, Dandan Zheng, Fudong Wang, et al. Ming-omni: A unified multimodal model for perception and generation. *arXiv preprint arXiv:2506.09344*, 2025.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15791–15801, 2025.

[5] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[6] Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*, 2024.

[7] Harold Haodong Chen, Xinxiang Yin, Wen-Jie Shu, Hongfei Zhang, Zixin Zhang, Chenfei Liao, Litao Guo, Qifeng Chen, and Ying-Cong Chen. Show, don't tell: Morphing latent reasoning into image generation. *arXiv preprint arXiv:2602.02227*, 2026.

[8] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

[9] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[10] Yinbo Chen, Rohit Girdhar, Xiaolong Wang, Sai Saketh Rambhatla, and Ishan Misra. Diffusion autoencoders are scalable image tokenizers. *arXiv preprint arXiv:2501.18593*, 2025.

[11] Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024.

[12] Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, et al. Emu3. 5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

[13] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[14] Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*, 2024.

[15] Sinan Du, Jiahao Guo, Bo Li, Shuhao Cui, Zhengzhuo Xu, Yifu Luo, Yongxian Wei, Kun Gai, Xinggang Wang, Kai Wu, et al. Vqrae: Representation quantization autoencoders for multimodal understanding, generation and reconstruction. *arXiv preprint arXiv:2511.23386*, 2025.

[16] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

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

[18] Weichen Fan, Haiwen Diao, Quan Wang, Dahua Lin, and Ziwei Liu. The prism hypothesis: Harmonizing semantic and pixel representations via unified autoencoding. *arXiv preprint arXiv:2512.19693*, 2025.

[19] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2025. URL https://arxiv.org/abs/2306.13394.

[20] Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, et al. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv preprint arXiv:2507.22058*, 2025.

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

[22] Jiawei Gu, Yunzhuo Hao, Huichen Will Wang, Linjie Li, Michael Qizhe Shieh, Yejin Choi, Ranjay Krishna, and Yu Cheng. Thinkmorph: Emergent properties in multimodal interleaved chain-of-thought reasoning. *arXiv preprint arXiv:2510.27492*, 2025.

[23] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

[24] Jiaming Han, Hao Chen, Yang Zhao, Hanyu Wang, Qi Zhao, Ziyan Yang, Hao He, Xiangyu Yue, and Lu Jiang. Vision as a dialect: Unifying visual understanding and generation via text-aligned representations. *Advances in Neural Information Processing Systems*, 38:158430–158459, 2026.

[25] Ruiyan Han, Zhen Fang, XinYu Sun, Yuchen Ma, Ziheng Wang, Yu Zeng, Zehui Chen, Lin Chen, Wenxuan Huang, Wei-Jie Xu, et al. Unicorn: Towards self-improving unified multimodal models through self-generated supervision. *arXiv preprint arXiv:2601.03193*, 2026.

[26] Jitai Hao, Hao Liu, Xinyan Xiao, Qiang Huang, and Jun Yu. Uni-x: Mitigating modality conflict with a two-end-separated architecture for unified multimodal models. *arXiv preprint arXiv:2509.24365*, 2025.

[27] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

[28] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37, 2024.

[29] Runhui Huang, Chunwei Wang, Junwei Yang, Guansong Lu, Yunlong Yuan, Jianhua Han, Lu Hou, Wei Zhang, Lanqing Hong, Hengshuang Zhao, et al. Illume+: Illuminating unified mllm with dual visual tokenization and diffusion refinement. *arXiv preprint arXiv:2504.01934*, 2025.

[30] Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. Geneval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025.

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

[32] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[33] Siqi Kou, Jiachun Jin, Zhihong Liu, Chang Liu, Ye Ma, Jian Jia, Quan Chen, Peng Jiang, and Zhijie Deng. Orthus: Autoregressive interleaved image-text generation with modality-specific heads. *arXiv preprint arXiv:2412.00127*, 2024.

[34] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

[35] Yann LeCun. A path towards autonomous machine intelligence. *Open Review*, 62(1):1–62, 2022.

[36] Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*, 2025.

[37] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[38] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*, 2025.

[39] Han Li, Xinyu Peng, Yaoming Wang, Zelin Peng, Xin Chen, Rongxiang Weng, Jingang Wang, Xunliang Cai, Wenrui Dai, and Hongkai Xiong. Onecat: Decoder-only auto-regressive model for unified understanding and generation. *arXiv preprint arXiv:2509.03498*, 2025.

[40] Teng Li, Quanfeng Lu, Lirui Zhao, Hao Li, Xizhou Zhu, Yu Qiao, Jun Zhang, and Wenqi Shao. Unifork: Exploring modality alignment for unified multimodal understanding and generation. *arXiv preprint arXiv:2506.17202*, 2025.

[41] Yanghao Li, Rui Qian, Bowen Pan, Haotian Zhang, Haoshuo Huang, Bowen Zhang, Jialing Tong, Haoxuan You, Xianzhi Du, Zhe Gan, et al. Manzano: A simple and scalable unified multimodal model with a hybrid vision tokenizer. *arXiv preprint arXiv:2509.16197*, 2025.

[42] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

[43] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

[44] Haokun Lin, Teng Wang, Yixiao Ge, Yuying Ge, Zhichao Lu, Ying Wei, Qingfu Zhang, Zhenan Sun, and Ying Shan. Toklip: Marry visual tokens to clip for multimodal comprehension and generation. *arXiv preprint arXiv:2505.05422*, 2025.

[45] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

[46] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl, 2025. *URL https://arxiv. org/abs/2505.05470*, 2:5, 2025.

[47] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024.

[48] Zhiheng Liu, Weiming Ren, Haozhe Liu, Zijian Zhou, Shoufa Chen, Haonan Qiu, Xiaoke Huang, Zhaochong An, Fanny Yang, Aditya Patel, et al. Tuna: Taming unified visual representations for native unified multimodal models. *arXiv preprint arXiv:2512.02014*, 2025.

[49] 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.

[50] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7739–7751, 2025.

[51] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

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

[53] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

[54] Luozheng Qin, Jia Gong, Yuqing Sun, Tianjiao Li, Mengping Yang, Xiaomeng Yang, Chao Qu, Zhiyu Tan, and Hao Li. Uni-cot: Towards unified chain-of-thought reasoning across text and vision. *arXiv preprint arXiv:2508.05606*, 2025.

[55] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2545–2555, 2025.

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

[57] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. *arXiv preprint arXiv:2104.05859*, 2021.

[58] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[59] Wei Song, Yuran Wang, Zijia Song, Yadong Li, Haoze Sun, Weipeng Chen, Zenan Zhou, Jianhua Xu, Jiaqi Wang, and Kaicheng Yu. Dualtoken: Towards unifying visual understanding and generation with dual visual vocabularies. *arXiv preprint arXiv:2503.14324*, 2025.

[60] 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.

[61] 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.

[62] Hao Tang, Chenwei Xie, Xiaoyi Bao, Tingyu Weng, Pandeng Li, Yun Zheng, and Liwei Wang. Unilip: Adapting clip for unified multimodal understanding, generation and editing. *arXiv preprint arXiv:2507.23278*, 2025.

[63] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[64] Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning, 2024. URL https://arxiv.org/abs/2412.14164.

[65] Shengbang Tong, David Fan, John Nguyen, Ellis Brown, Gaoyue Zhou, Shengyi Qian, Boyang Zheng, Théophane Vallaeys, Junlin Han, Rob Fergus, et al. Beyond language modeling: An exploration of multimodal pretraining. *arXiv preprint arXiv:2603.03276*, 2026.

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

[67] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[69] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[70] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[71] 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.

[72] Yuqing Wang, Zhijie Lin, Yao Teng, Yuanzhi Zhu, Shuhuai Ren, Jiashi Feng, and Xihui Liu. Bridging continuous and discrete tokens for autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18596–18605, 2025.

[73] Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. Univideo: Unified understanding, generation, and editing for videos. *arXiv preprint arXiv:2510.08377*, 2025.

[74] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837, 2022.

[75] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[76] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977, 2025.

[77] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

[78] Qiucheng Wu, Handong Zhao, Michael Saxon, Trung Bui, William Yang Wang, Yang Zhang, and Shiyu Chang. Vsp: Assessing the dual challenges of perception and reasoning in spatial planning tasks for vlms. *arXiv preprint arXiv:2407.01863*, 2024.

[79] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Zhonghua Wu, Qingyi Tao, Wentao Liu, Wei Li, and Chen Change Loy. Harmonizing visual representations for unified multimodal understanding and generation. *arXiv preprint arXiv:2503.21979*, 2025.

[80] 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.

[81] Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*, 2025.

[82] Eric Xing, Mingkai Deng, Jinyu Hou, and Zhiting Hu. Critiques of world models. *arXiv preprint arXiv:2507.05169*, 2025.

[83] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2025. *URL https://arxiv. org/abs/2411.10440*, 2024.

[84] Zhiyuan Yan, Kaiqing Lin, Zongjian Li, Junyan Ye, Hui Han, Zhendong Wang, Hao Liu, Bin Lin, Hao Li, Xue Xu, et al. Can understanding and generation truly benefit together–or just coexist? *arXiv e-prints*, pages arXiv–2509, 2025.

[85] Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025.

[86] Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, et al. Echo-4o: Harnessing the power of gpt-4o synthetic images for improved image generation. *arXiv preprint arXiv:2508.09987*, 2025.

[87] Chao Yi, Yu-Hang He, De-Chuan Zhan, and Han-Jia Ye. Bridge the modality and capability gaps in vision-language model selection. *Advances in Neural Information Processing Systems*, 37:34429–34452, 2024.

[88] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

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

[90] Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

[91] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

A More Related Work

Building effective UMs requires visual representations for both understanding and generation. Prior work argues that these two tasks require fundamentally different features—semantic features for understanding [56, 66] versus pixel details for generation [16, 31, 67]—leading to approaches that adopt separate visual encoders [9, 26, 29, 40, 76, 84] or dual vocabularies [41, 55, 59]. Other works pursue unified tokenizers with shared representations [2, 18, 39, 44, 48, 49, 53, 60, 62, 79, 80], or explore semantic features directly for generation [10, 15, 20, 24, 89].

Most UMs focus on single-turn generation or understanding, but a growing body of work investigates interleaved reasoning. Visual Chain-of-Thought extends text-based CoT [32, 74] to multimodal settings. Some approaches generate explicit intermediate visual steps [7, 28, 83], while others reason implicitly in latent spaces [14, 27, 36, 38, 85]. Most relevant to our work are ThinkMorph [22], Uni-CoT [54], and UniCorn [25], which study interleaved text-image reasoning traces. These approaches, however, rely on pixel-space mediation between visual understanding and generation [22, 54], where visual features must be decoded to pixels and re-encoded for understanding, introducing codec bias and cross-modal misalignment [18, 87].

World models predict future environmental states conditioned on past observations and actions [1, 23, 35, 82], a capability that naturally aligns with the unified understanding-and-generation objective of UMs. Recent unified models have begun to embrace this connection: Emu3.5 [12] demonstrates that next-token prediction on interleaved multimodal sequences yields generalizable world-modeling abilities, and Transfusion-RAE [65] finds that unified multimodal pretraining on action-conditioned video naturally gives rise to world modeling. LatentUM adopts the same perspective, predicting future visual states as semantic tokens in the shared latent space and treating world modeling as an instance of cross-modal reasoning.

B Training Details

B.1 Experimental Setup

During the pre-training of $\text{LatentUM}_{\text{Base}}$, the visual generation resolution is fixed to 448×448, represented by 256 tokens. The quantizer contains 101M trainable parameters, with each latent vector chunked into 8 parts and a codebook of 2048 codes per part [49]. The visual generation branch comprises 3633M parameters, and the autoregressive head is a 3-layer transformer with 283M parameters. For the decoupled pixel decoder, we fine-tune the conditioning branch (990M parameters) of MMDiT from Stable Diffusion 3.5 Medium [17].

B.2 MBAQ Tokenizer Training

The MBAQ quantizer follows a compress-quantize-expand pipeline. A two-layer MLP with GELU activation projects the input CLIP features from d=4096 to a bottleneck dimension of d_e=256. The projected features are quantized by a multi-codebook quantizer: the 256-dimensional vector is evenly partitioned into C=8 chunks of 32 dimensions each, and each chunk is independently quantized using a separate codebook of size K=2048, yielding an effective vocabulary of $K^C = 2048^8$ from only $K \times C = 16{,}384$ total codebook entries. A second two-layer MLP with GELU activation then projects the de-quantized features back to the LLM hidden size of 2560 for downstream language model consumption. Codebook embeddings are initialized uniformly in $[-1/K,\ 1/K]$.

The full training objective is:

$$\mathcal{L}(\phi) = \mathcal{L}_{\text{KL}} + \lambda\, \mathcal{L}_{\text{MCQ}}, \quad \lambda = 1.0, \tag{6}$$

where the KL divergence loss $\mathcal{L}_{\text{KL}}$ is computed only over answer token positions (Section 2.1). The MCQ commitment loss $\mathcal{L}_{\text{MCQ}}$ is averaged across all C codebooks:

$$\mathcal{L}_{\text{MCQ}} = \frac{1}{C} \sum_{c=1}^{C} \Big[\underbrace{\|\mathbf{z}_q^c - \text{sg}[\mathbf{z}^c]\|_2^2 + \beta\, \|\text{sg}[\mathbf{z}_q^c] - \mathbf{z}^c\|_2^2}_{\text{codebook + commitment loss}} - \underbrace{\alpha\, H(\bar{p}^c)}_{\text{entropy regularizer}} \Big], \tag{7}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator, β=0.25 is the commitment cost, and α=0.1 weights the entropy regularizer. $H(\bar{p}^c) = -\sum_k \bar{p}_k^c \log \bar{p}_k^c$ is the entropy of the soft code assignment distribution $\bar{p}^c = \text{mean}_{b,\ell}\, \text{softmax}(-\mathbf{d}_{b,\ell}^c)$, where $\mathbf{d}_{b,\ell}^c$ contains the squared distances to all codebook

entries for position (b, ℓ) in codebook c. Maximizing this entropy regularizes each codebook toward uniform code usage.

Gradient propagation through the quantization step uses the straight-through estimator (STE) [5]. Codebook utilization is tracked via EMA with decay γ=0.99. Codes whose EMA usage falls below $0.03/K$ are identified as dead codes and restarted: at most 64 dead codes are reset per step by replacing their embeddings with randomly sampled inputs from the current batch.

B.3 MoME Pretraining

The MoME generation branch is constructed by adding dedicated *vision* counterparts to every transformer layer of InternVL3.5-4B, following the Mixture-of-Transformers (MoT) design [13]. Specifically, for each Qwen3 decoder layer, we duplicate the Q/K/V/O projection matrices and Q/K normalization layers into vision-specific copies (`q/k/v/o_proj_vision`, `q/k_norm_vision`), and add a separate vision MLP (`mlp_vision`). All vision-specific parameters are initialized from the corresponding pretrained understanding parameters. At each forward pass, a `vision_token_mask` routes each token to the appropriate branch: visual generation tokens are processed by the `_vision` parameters, while text and understanding tokens use the original frozen parameters. Both branches share the same key–value attention computation, enabling cross-modal interaction through shared self-attention. A two-layer MLP projector maps the de-quantized MBAQ features into the LLM hidden space before being concatenated with text embeddings. The visual understanding branch (vision encoder and original LLM parameters) is fully frozen throughout pretraining.

The AR vision head is a 3-layer causal Transformer decoder with hidden size 2560, 32 attention heads, MLP ratio 4.0, RMSNorm, and 1D rotary position embeddings. It takes the MoME hidden state at each visual token position as a prefix and autoregressively predicts C=8 codebook indices in sequence, implementing the MC-NTP objective (Eq. 3).

The pretraining objective is cross-entropy over the predicted codebook indices as defined in Eq. 3. We apply classifier-free guidance (CFG) training by randomly dropping the text conditioning with probability 0.1. The model is trained on 32M text-image pairs from the BLIP3o dataset [8] (Short-Caption, Long-Caption, and JourneyDB subsets) at a fixed resolution of 448×448 pixels (256 tokens per image) with a maximum sequence length of 512 (including the text prompt token length). MoME pretraining takes approximately 4 days on 64×NVIDIA H100 GPUs.

At inference time, visual tokens are sampled autoregressively using nucleus sampling with top-k=32, top-p=0.95, and temperature τ=0.5. For standard benchmark evaluations, classifier-free guidance (CFG) is applied with scale s=3.0, where the unconditional prediction is obtained by dropping the text condition. Benefiting from the semantic alignment induced by MBAQ, LatentUM also supports CFG-free generation (s=1.0): since the generative visual tokens share the same semantic space as language, the model produces semantically coherent, instruction-following images even without CFG, a capability not available to models based on pixel-reconstruction features. Figure 8 presents side-by-side comparisons of images generated with different CFG scales under identical prompts.

B.4 Pixel Decoder Training

The pixel decoder is built upon the MMDiTX architecture of Stable Diffusion 3.5 Medium [17], which consists of 24 dual-stream transformer layers with a patch size of 2 operating on 16-channel VAE latents. We replace the original text conditioning module with a learned linear projector that maps the 256-dimensional MBAQ de-quantized embeddings to the MMDiT context dimension. The trainable parameters consist of the `context_embedder` and all `context_block` parameters (the conditioning stream of each transformer layer), totaling approximately 990M parameters. All remaining parameters, including the image stream (`x_embedder`, `x_block`), the VAE, and the MBAQ quantizer are fully frozen throughout training.

The decoder is trained with a flow matching objective [17]. Timesteps are sampled via logit-normal distribution with mean μ=0.0, standard deviation σ=1.0, and mode scale 1.29, following the SD3 training recipe. The training loss is a weighted mean squared error between the predicted and target latents:

$$\mathcal{L}_{\text{diff}}(\eta) = \mathbb{E}_{u,\boldsymbol{\epsilon},\mathbf{x}_0}\left[w(\sigma_u)\,\|\hat{\mathbf{x}}_0 - \mathbf{x}_0\|^2\right], \tag{8}$$

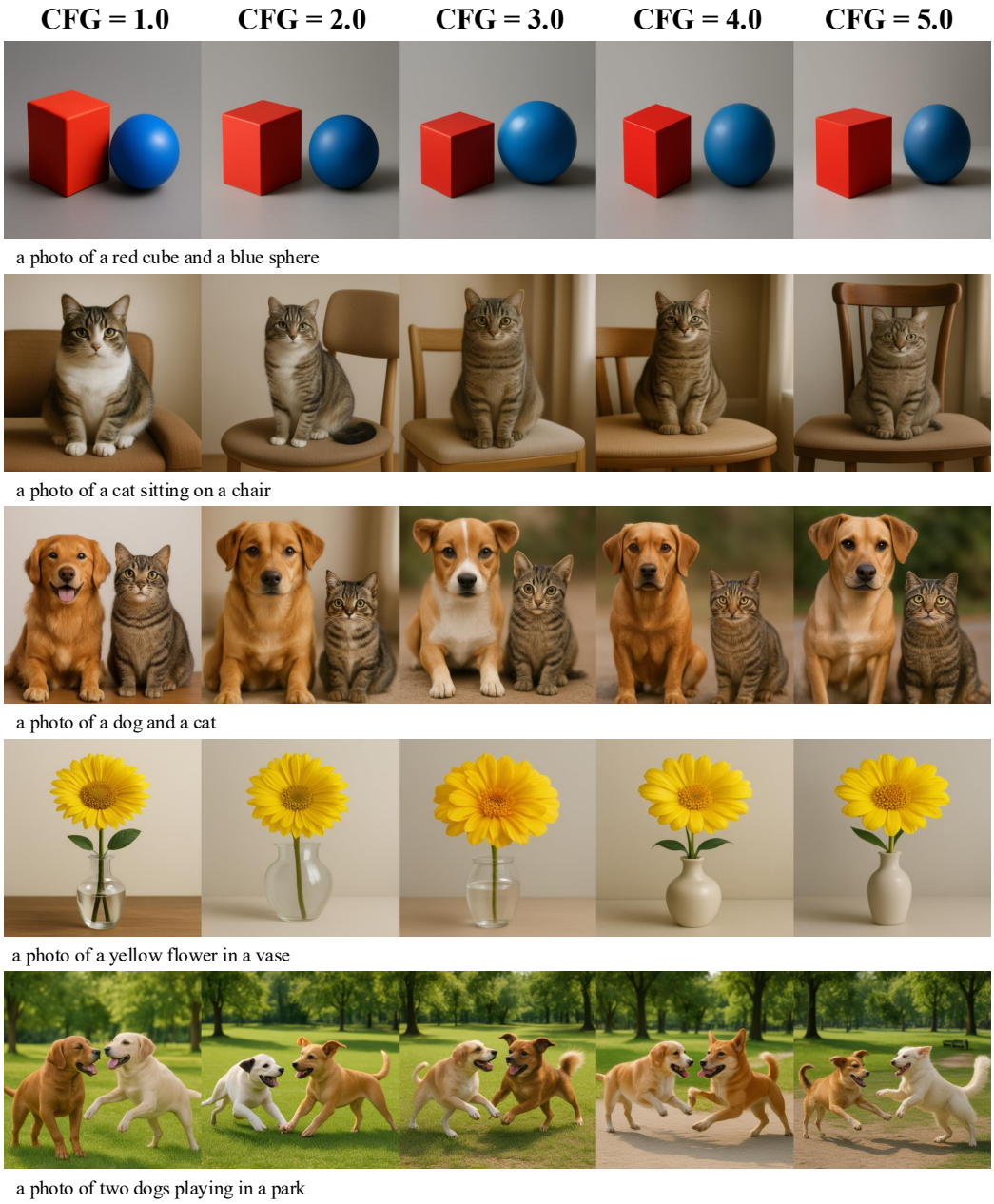


Figure 8: Generation results under different CFG scales (1.0 to 5.0) for identical prompts. LatentUM produces coherent images even at CFG $= 1.0$.

where $\hat{\mathbf{x}}_0$ is the model's predicted clean latent, σ_u is the noise level at timestep u, and $w(\sigma_u)$ is the logit-normal loss weight.

The decoder is pretrained on the same 32M text-image pairs from BLIP3o [8] used for MoME pretraining, with per-GPU batch size 45. This stage takes approximately 2 days on 32×NVIDIA H100 GPUs.

At inference time, the pixel decoder uses a Flow Matching Euler discrete scheduler [17] with N=25 denoising steps at a fixed resolution of 448×448. The MBAQ de-quantized features $\tilde{\mathbf{V}}$ serve as the sole conditioning signal, with classifier-free guidance scale set to s=1.0 (i.e., no unconditional guidance branch). This is possible because $\tilde{\mathbf{V}}$ already carries rich semantic content from the MBAQ-

aligned latent space, providing sufficiently strong conditioning. This phenomenon is also reported by X-Omni which also conducts autoregressive generation over semantic tokens [20].

C Additional Analysis of Visual Generation with Self-Reflection

C.1 GRPO setup

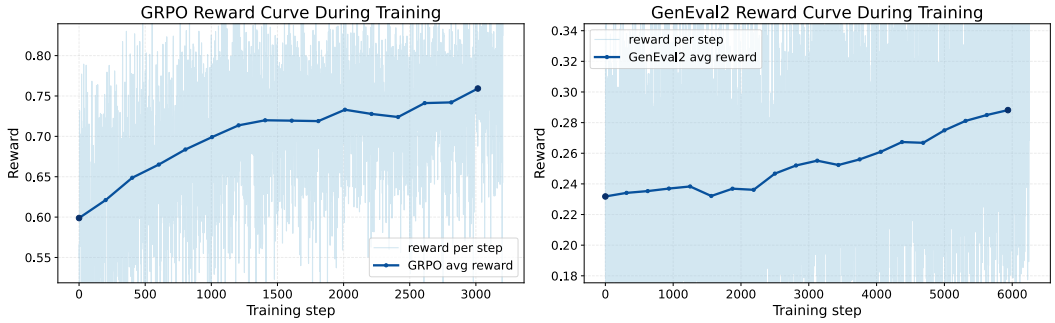


Figure 9: Reward curve during GRPO training for self-reflection visual generation.

For visual generation with self-reflection, we optimize $\text{LatentUM}_{\text{Base}}$ with GRPO using prompt-wise grouped rollouts. Each prompt produces $G{=}16$ samples, and advantages are normalized within each prompt group. We use PPO-style clipping with clip parameter 0.1 and KL regularization weight $\beta{=}0.005$. Rollouts are generated without classifier-free guidance, using guidance scale 1.0, sampling temperature 0.9, top-$p{=}0.95$.

The trainable parameters are restricted to the visual-generation pathway introduced by MoME, including the vision-specific transformer parameters, the visual projector, and the autoregressive vision head. This restriction keeps post-training focused on improving generation behavior while preserving the pretrained understanding backbone that supplies the self-reward signal. Concretely, the understanding branch ψ is frozen throughout self-reward RL and is used only as the evaluator, while the visual-generation pathway θ is updated as the policy. The pixel decoder is not used for reward computation; rewards are computed directly from the generated semantic visual features, avoiding an additional decode-and-detect loop during training.

C.2 Prompt decomposition into verification questions

The self-reward is computed by decomposing each generation prompt into a set of atomic visual checks and then asking the understanding branch to answer them as multiple-choice questions. The decomposition covers the semantic factors evaluated by GenEval-style benchmarks, including object existence, object count, color, material, pattern, and pairwise spatial relations. For each prompt, we derive the ground-truth answer to each check directly from the prompt specification. Object existence questions are created for mentioned objects, count questions are created when cardinality is specified, attribute questions bind colors/materials/patterns to their target objects, and relation questions check pairwise spatial constraints. Each check is converted into a multiple-choice question with alphabet-prefixed options, such as `A. 0`, `B. 1`. The following is an example:

Prompt: "five black teddy bears"

Q1 (count). Answer the following question with the given option only. How many valid teddy bears are there in the image? Valid means the object is clearly visible, complete and not partially occluded by other objects and contains no artifacts.
Options: A: 0, B: 1, C: 2, D: 3, E: 4, F: 5, G: 6, H: 7, I: 8, J: 9, K: 10.
Ground-truth option: F (5).

Q2 (color). Answer the following question with the given option only. What color is the teddy bear in the image?
Options: A: red, B: orange, C: yellow, D: green, E: blue, F: purple, G: pink, H: brown, I: black, J: white.
Ground-truth option: I (black).

Q3 (existence). Answer the following question with the given option only. Does a/an teddy bear exist in the image? Answer yes only when the object is clearly visible, complete and not partially occluded by other objects and contains no artifacts.
Options: A: Yes, B: No.
Ground-truth option: A (Yes).

During rollout, the generated latent visual sequence is first mapped back to semantic visual features. The understanding branch then evaluates these features with each verification question without explicitly decoding a textual answer. Instead, at the answer position, we directly extract the logits of all candidate option tokens (e.g., A/B/C/...) and apply a softmax over this restricted option set. Specifically, for a verification question ν_m with candidate option-token logits $\{\rho_{m,k}\}_k$ and ground-truth option index $k_m^\star$, we compute the per-question score as $p_m = \frac{\exp(\rho_{m,k_m^\star}/\tau)}{\sum_k \exp(\rho_{m,k}/\tau)}$. This uses the first option-token logits only, so reward evaluation does not require free-form answer decoding.

If a prompt is decomposed into M verification questions and the resulting ground-truth option probabilities are $\{p_m\}_{m=1}^M$, we define the self-reward as

$$\mathcal{R} = \prod_{m=1}^{M} p_m. \tag{9}$$

This multiplicative design is intentionally strict: a sample receives a high reward only when it satisfies all required constraints simultaneously, rather than only a subset of them. As a result, the reward favors globally correct generations with consistent object counts, attributes, and relations.

C.3 Qualitative Comparison Before and After RL

Figure 9 shows that the reward increases steadily during GRPO training. Figure 10 further shows the qualitative effect of self-reward GRPO. Compared with $\text{LatentUM}_{\text{Base}}$, the post-trained model follows compositional constraints more accurately, especially for counting, attribute binding, and spatial relations. This result is consistent with the design of the reward, which directly encourages semantic correctness rather than low-level visual fidelity.

C.4 Failure Cases of Detection-Based Metrics

Figure 11 shows representative examples in which detection-based metrics incorrectly judge the generated samples as failures. In these cases, the generated images are qualitatively correct, but the GenEval detection-based metric still marks them as failures. This indicates that detection-based metrics are not always reliable for evaluating semantic instruction following. This problem is especially visible for dense layouts, small objects, and visually unusual but valid renderings. As a result, a detection-based metric may reflect detector brittleness rather than the true semantic correctness of the generated image.

This is also why we introduce GenEval2 with a VLM-as-a-judge evaluator in the main text. Compared with detection-based scoring, a VLM judge is better aligned with the semantic content of the prompt and is less brittle to visual variation.



Figure 10: Qualitative comparison before and after self-reward GRPO. Top: LatentUM$_{\text{Base}}$ before GRPO. Bottom: the model after self-reward GRPO. The post-trained model follows counting prompts more accurately.



Figure 11: Examples where the generated images are qualitatively correct, but the GenEval detection-based metrics mark them as failures.

D Fine-Grained VSP Data Construction

We construct the fine-grained VSP training set by procedurally generating deterministic FrozenLake mazes and converting each solvable maze into an interleaved action-image trajectory. The benchmark contains four difficulty levels corresponding to 3×3 to 6×6 grids, and we generate mazes at the corresponding scales. For a maze of size $n \times n$, we first sample two distinct cells as the start S and goal G. Every remaining cell is independently assigned to either frozen land F (traversable) or a hole H (blocked), yielding a random layout $m \in \{S, F, H, G\}^{n \times n}$.

Solvability and path filtering. We treat each non-hole cell as a node in a 4-neighbor grid graph and compute a shortest path from S to G with breadth-first search. Layouts with no valid path are

discarded. We further retain only mazes whose shortest-path length lies in a predefined interval, which removes trivial instances and controls the planning horizon. To reduce repeated supervision on identical maps, we de-duplicate mazes by their flattened grid string and keep only one copy of each unique layout. Each retained example therefore corresponds to one unique solvable maze and one expert shortest-path trajectory $s_0, s_1, \ldots, s_T$.

Step-wise rendering. Given the trajectory, we derive an action sequence $a_t \in \{U, D, L, R\}$ from consecutive state pairs (s_{t-1}, s_t). We then render every intermediate state directly from the deterministic FrozenLake simulator. For step t, the rendered image shows the agent at state s_t and overlays all previously visited cells in red; when the agent reaches the goal, the goal cell is additionally highlighted in green. The rendered images are resized to a fixed resolution determined by the maze size, which yields visually consistent supervision across difficulty levels.

Interleaved supervision format. We finally convert each trajectory into the fine-grained format shown in Figure 5(b). The user prompt contains only the initial image I_0, while the target response is an alternating sequence of actions and next-state images:

$$a_1 \texttt{<img>}\, a_2 \texttt{<img>}\, \cdots\, a_T \texttt{<img>}\, \texttt{Done!},$$

where the t-th <img> slot is supervised by the rendered image I_t after executing action a_t. This format forces the model to alternate between textual action prediction and visual state updating at every step, rather than solving the maze with a single coarse plan.

E Improving Decoding Consistency in Long-Horizon Prediction

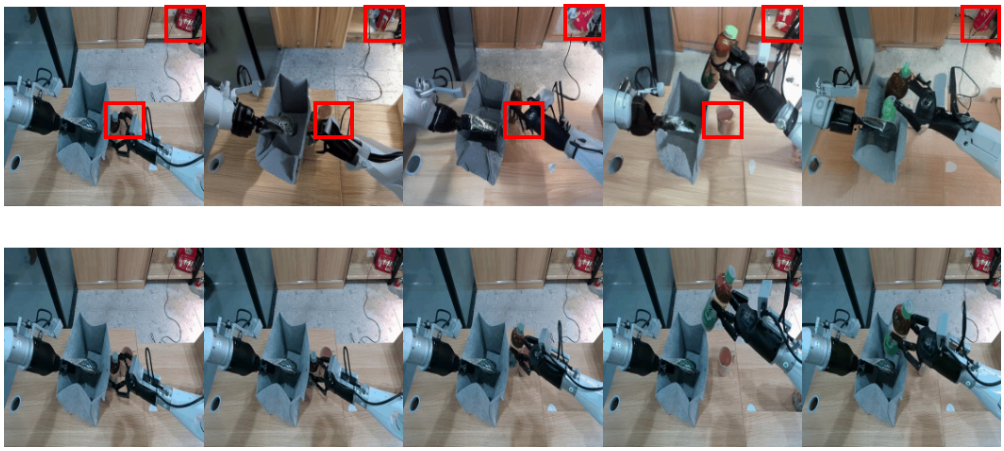


Figure 12: Top: independent keyframe decoding with the original decoupled decoder. Bottom: reference-conditioned decoding with the first-frame VAE latent. The bottom row shows much better cross-frame consistency.

For long-horizon prediction tasks, such as robot-motion forecasting or video keyframe prediction, semantic correctness alone is not sufficient. The decoded frames should also remain visually consistent across time, so that object identity, scene layout, and appearance do not drift from one step to the next. In practice, this is a challenge for a decoupled decoder, because decoding from semantic tokens alone may recover the correct high-level content while still introducing frame-to-frame inconsistency in low-level appearance. Figure 12 shows that adding a first-frame reference during decoding leads to much more consistent keyframe predictions.

To address this issue, we augment the decoder with an additional reference-image condition. In addition to the quantized semantic tokens of the target frame, we feed the decoder with the VAE latent of a reference frame, chosen as the first frame of the predicted sequence. Intuitively, the semantic tokens specify *what* should be rendered at the current step, while the reference latent provides a stable visual anchor for *how* the scene should look across the sequence. This design preserves the flexibility of semantic prediction while improving temporal consistency during decoding.

We find that this reference-conditioned decoder substantially improves the consistency of decoded keyframes in long-horizon settings. In particular, it is better at maintaining stable appearance over extended prediction horizons, which is important for applications such as world models and long-term visual planning that requires pixel space information. These results suggest that when decoded frames are used as intermediate states rather than standalone images, incorporating an explicit reference signal is an effective way to reduce visual drift in rendered pixels.

F Zero-shot language-conditioned navigation



Figure 13: Without language supervision during fine-tuning, LatentUM$_{\text{WM}}$ translates free-form textual instructions (e.g., "get closer to the left house") into accurate future visual states, enabled by the inherited language-vision alignment.

As an emergent capability, we observe that LatentUM$_{\text{WM}}$ can perform zero-shot language-conditioned navigation. Although the model is fine-tuned exclusively on standard action tokens without any free-form language supervision for navigation tasks, it can generalize to natural language instructions.

As illustrated in Figure 13, given an unconstrained textual prompt such as "get closer to the left house," the model leverages the rich language-vision alignment inherited from LatentUM$_{\text{Base}}$ to predict temporally and semantically consistent future visual states. This highlights the robustness and flexibility of the unified semantic latent space in interpreting and grounding open-ended textual conditions into physical-world dynamics.

G Additional Experiments

G.1 Data-Matched Comparison with ThinkMorph on VSP

To isolate the effect of the latent representation from task-specific supervision, we retrain ThinkMorph [22] on exactly the same fine-grained VSP dataset and alternating action-image reasoning format as LatentUM$_{\text{Vis-Plan}}$ (Appendix D). ThinkMorph is initialized from BAGEL-14B [13], a substantially larger backbone than that of LatentUM. Because it relies on pixel-space mediation, each intermediate visual state requires processing ViT features, VAE features, and noisy VAE features, resulting in roughly three times the visual representation overhead. Consequently, on the same cross-modal reasoning trajectories, ThinkMorph produces sequences approximately 4–5$\times$ longer than LatentUM, making each training step several times more computationally expensive.

Table 5: Data-matched comparison on VSP [78]. Both models are trained on the same fine-grained VSP data with the same alternating action-image reasoning format.

Model	L3	L4	L5	L6	Avg.
ThinkMorph [22]	0.92	0.83	0.67	0.43	0.71
LatentUM$_{\text{Vis-Plan}}$	1.00	1.00	1.00	0.97	0.99

As shown in Table 5, ThinkMorph still underperforms LatentUM despite its larger backbone and a training compute budget substantially exceeding that used for LatentUM. Inspection of the evaluation rollouts shows that its intermediate images become progressively noisier over later reasoning steps, indicating stronger error accumulation from repeated pixel-space decoding and re-encoding. This issue is particularly pronounced on higher-level VSP cases, which require longer reasoning trajectories and therefore amplify accumulated visual errors.

G.2 Out-of-Distribution Generalization on VSP

To evaluate out-of-distribution generalization, we test the same $\text{LatentUM}_{\text{Vis-Plan}}$ checkpoint without any additional fine-tuning under five controlled shifts: (1) *unseen object appearances*, where the hole, goal, and agent are drawn with entirely different artwork; (2) *color shift and image degradation*, including hue rotation, contrast scaling, Gaussian noise, and JPEG compression; (3) *grayscale rendering*, which removes the color cue separating holes from the goal so that they must be distinguished by shape alone; (4) *larger grid*, extending the maze from 6×6 to 7×7 at a fixed hole density; and (5) *confusable distractors*, where objects visually similar to holes and to the goal are placed on safe cells. We generate 100 new mazes for each setting. In addition to the task success rate, we report the step-wise optimal-action rate, defined as the fraction of predicted actions that strictly reduce the shortest-path distance to the goal.

Table 6: Out-of-distribution generalization of $\text{LatentUM}_{\text{Vis-Plan}}$ on VSP without additional fine-tuning.

Setting	Success rate	Step-wise optimal-action rate
In-distribution (6×6)	0.97	0.99
Unseen object appearances (6×6)	0.61	0.89
Color shift and degradation (6×6)	0.39	0.97
Grayscale rendering (6×6)	0.12	0.91
Larger grid (7×7)	0.41	0.87
Confusable distractors (6×6)	0.59	0.85

As shown in Table 6, the success rate decreases under all out-of-distribution settings, whereas the step-wise optimal-action rate remains relatively high. Our error analysis suggests that much of the drop in end-to-end success is caused by failures in termination detection rather than in spatial planning: the model often continues to select actions that correctly move it toward the goal, but fails to recognize that the goal has been reached because the out-of-distribution transformations alter the visual pattern used to identify the goal state. These results indicate that the learned policy retains substantial local planning capability under visual shifts, while its goal-state recognition remains sensitive to appearance changes.

G.3 Cross-VLM Generalization of MBAQ

MBAQ is not intrinsically tied to InternVL3.5: its objective only requires a VLM that consumes continuous visual features and produces language-output distributions, so the same formulation can be applied to other VLMs. To verify this, we repeat MBAQ training and evaluation with Qwen3-VL-2B-Instruct [3].

Table 7: MBAQ applied to Qwen3-VL-2B-Instruct. MBAQ discretizes the continuous features of the Qwen3-VL vision encoder while preserving most of the original visual understanding capability.

Model	MME-P	POPE	SEED	MMBench	MMMU_{val}
Qwen3-VL-2B (continuous)	1493	88.6	75.1	78.0	44.9
Qwen3-VL-2B (discretized via MBAQ)	1453	88.7	74.9	77.2	43.0

As shown in Table 7, behavior-aligned quantization generalizes across VLM families rather than depending specifically on InternVL3.5. Nevertheless, the learned quantizer remains backbone-specific, because different VLMs have distinct visual feature spaces and downstream behaviors; a separate MBAQ quantizer should therefore be trained for each target VLM.

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 did it in the Abstract and Introduction.

   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 did it in the Conclusion.

   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 any theoretical result.

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 disclose all detailed needed to reproduce the main experiments.

   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 will provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results.

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: See Experiments and Appendix

   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: [No]

   Justification: Error bars are not reported because it would be too computationally expensive.

   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 it in Experiments and Appendix.

   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: Yes, we follow the 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: [No]

    Justification: This paper focuses on academic research and there's no explicit negative social impacts.

    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 poses no such risks.

    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 strictly followed the licenses.

    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: This paper does not release new assets.

    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: This paper does not involve crowdsourcing nor research with human subjects.

    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: Our research does not require the IRB.

    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: We did not use LLMs in the core method development.

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.