Representation Forcing for Bottleneck-Free Unified Multimodal Models

Yuqing Wang[1,2 †] **Zhijie Lin**[2 ‡] **Ceyuan Yang**[2] **Yang Zhao**[2] **Fei Xiao**[2]
Hao He[3,2 †] **Qi Zhao**[2] **Zihan Ding**[2] **Fuyun Wang**[3,2 †] **Shuai Wang**[4,2 †]
Youliang Zhang[5,2 †] **Haoqi Fan**[2] **Xihui Liu**[1 ✉]

[1]University of Hong Kong [2]ByteDance Seed
[3]The Chinese University of Hong Kong [4]Nanjing University [5]Tsinghua University

Abstract

Unified multimodal models (UMMs) aim to handle perception and generation in a single model. Yet existing UMMs still rely on a frozen, separately pretrained VAE for image generation, imposing a structural bottleneck. Naively removing it introduces a quality gap, as the model must learn both high-level structure and low-level details from raw pixels. In this paper, we propose Representation Forcing (RF), a technique that closes this gap by making representation prediction a native capability of the model. Concretely, RF forces the model to autoregressively predict visual representations as intermediate tokens before pixels; these tokens then stay in context to guide pixel diffusion within the same backbone. By turning representations from perception outputs into generation targets, RF eliminates the need for any external generative latent space. We find that RF benefits both understanding and generation. On image generation, our pixel-space model with RF matches state-of-the-art VAE-based unified models. On image understanding, pixel-space RF generally outperforms its VAE-based variant. Together, these results offer an effective step toward end-to-end, bottleneck-free UMMs. Project page: `https://yuqingwang1029.github.io/RepresentationForcing`

1 Introduction

The ability to perform understanding (text output) and generation (pixel output) within a unified framework represents a fundamental step toward general-purpose multimodal intelligence [5, 11, 50, 54, 57]. Prevailing unified multimodal models (UMMs) pursue this by bringing language and image generation into a shared transformer backbone [11, 34, 66], with next-token prediction for language and diffusion for image generation. However, despite this unification, the image generation pathway still depends on a separately pretrained, frozen VAE [15, 27, 42]: images are compressed into latents before diffusion is applied, and pixels are recovered through a fixed decoder. This creates a structural bottleneck. The latent space is optimized for reconstruction rather than the objectives of the unified model, and its lossy compression imposes a hard upper bound on generation quality that further training of the UMM cannot overcome. Removing this bottleneck is an important step toward end-to-end UMMs.

A natural alternative is to generate directly in pixel space. Recent works have shown pixel-space diffusion to be feasible for standalone generation models [8, 28, 49]. However, we find that directly applying these methods in UMMs fails to match the quality of VAE-based counterparts. We attribute this to the broader image distribution and richer text conditioning in UMMs: the model must learn

[†]Work done during an internship at ByteDance Seed.
[‡]Project lead.
[✉]Corresponding author.

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

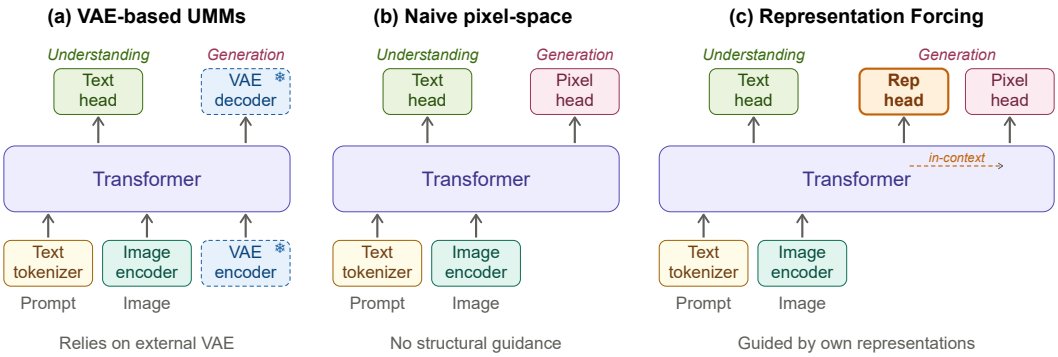

Figure 1: **Architectural comparison.** (a) Prevailing UMMs rely on a frozen VAE encoder and decoder for image generation, creating a structural bottleneck. (b) Naively removing the VAE and generating directly in pixel space eliminates this bottleneck but loses structural guidance, leading to a quality gap. (c) Representation Forcing closes this gap by training the transformer decoder to autoregressively predict visual representations (Rep head) before pixel generation. These representations are trained to match features from the model's own understanding encoder and remain in context within the shared transformer, providing structural guidance for pixel-space diffusion without any external latent space.

both the high-level semantic structure and fine-grained details of an image from the same raw signal. This motivates an intermediate representation that separates these two factors, so that the diffusion process can focus on low-level rendering without falling back to an external latent space.

The key question is where such a representation should come from. We observe that UMMs already provide one internally: their understanding pathway learns visual representations that capture high-level structure, such as object identity, spatial layout, and scene composition. In understanding, the encoder [37, 44, 46, 64] extracts these representations from observed images. In generation, however, no image is available, and the model must produce them from the input context alone. This means the model must learn to predict these representations on its own.

In this paper, we propose **Representation Forcing** (RF), an approach that closes this gap by making representation prediction a native capability of the model. Our key idea is to ground the high-level visual representation in the decoder itself. Concretely, we use visual representations extracted by the understanding encoder as targets, and train the model decoder to predict them autoregressively under the same next-token prediction objective used for language. These predicted representations provide an explicit structural scaffold between text and pixels; they remain in the sequence as in-context conditioning, guiding pixel-space generation within the shared transformer. By turning representations from perception outputs into generation targets, RF grounds understanding and generation in a single representation space, without relying on a separately pretrained latent space (Figure 1).

To validate Representation Forcing, we apply it to both pixel-space and VAE-based UMMs under controlled settings with the same architecture, data, and training budget. On image generation, our pixel-space model with RF matches the VAE-based baseline across standard benchmarks while preserving rich textural details (Figure 2). On understanding, pixel-space RF outperforms its VAE-based variant, showing that pixel-space generation is more compatible with unified multimodal modeling than VAE-based generation. Ablations further reveal that RF is critical for pixel-space generation, while also bringing improvements to VAE-based settings. Together, these findings demonstrate that Representation Forcing benefits both directions of unified multimodal modeling, offering an effective step toward pixel-space, bottleneck-free UMMs. Here, a bottleneck refers to a frozen, separately pretrained module that cannot be optimized on the training data, and end-to-end means that all parameters are trainable and jointly optimized.

The main contributions of this work are summarized as follows:

- We propose Representation Forcing (RF), a simple approach that closes the quality gap of pixel-space image generation in unified multimodal models, eliminating the need for any pretrained VAE. RF trains the model to autoregressively predict visual representations



Figure 2: Text-to-image generation results at 1024×1024 resolution from our pixel-space unified model with Representation Forcing.

as intermediate tokens, which then serve as in-context structural guidance for pixel-space diffusion within the same backbone.

- RF benefits both image generation and understanding across pixel-space and VAE-based UMMs. In particular, our pixel-space model with RF matches the VAE-based counterpart on generation and outperforms it on understanding, suggesting that pixel-space generation is more compatible with unified multimodal modeling than VAE-based generation.
- Our work advocates for unified multimodal models where perception and generation share a single, end-to-end-learned representation space, rather than coordinate across separately pretrained components such as external VAEs. Representation Forcing is a step toward fully unified multimodal models, where all capabilities are learned end-to-end within the model itself rather than inherited from independently trained components.

2 Related Work

Unified Multimodal Models. Existing UMMs broadly fall into two families. The first generates within a *single backbone*, either by modeling images as discrete tokens (Chameleon [5], Emu3 [50]) or by attaching diffusion in a VAE latent space (Transfusion [66], Show-o [57], JanusFlow [34]). These approaches further address the interference between understanding and generation through decoupled visual encoders in Janus series [9, 34, 54] or modality-specific experts in BAGEL [11], but all depend on a separately pretrained visual tokenizer—a VQVAE or a continuous VAE—to define the latent space for generation. The second family *stitches an LLM with an external diffusion model*: the LLM predicts visual representations such as CLIP features, which then condition a separately trained diffusion decoder to render images, as in Emu2 [45], SEED-X [20], BLIP3-o [6], and MetaQueries [38]. More recently, Omni [60] expands the set of natively trained modalities to text, images, video, and 3D geometry, and rolls out intermediate multimodal context within the backbone before decoding. A growing line of work further seeks to remove the pretrained VAE that bottlenecks these models. Concurrent with our work, SenseNova-U1 [14] and Tuna-2 [32] build pixel-space UMMs that discard the visual encoder and generate directly from raw patches. Representation Forcing shares this goal of a bottleneck-free, end-to-end UMM, but takes a different path. It builds on the same MoT backbone design [11] and retains a jointly trained understanding encoder, turning its representations into intermediate generation targets that serve as in-context conditioning for pixel-space diffusion within the same transformer, yielding a fully end-to-end UMM.

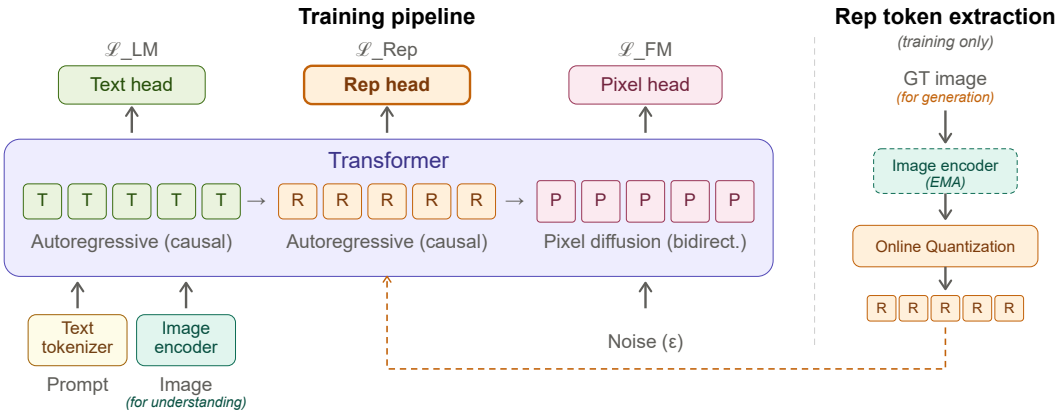

Figure 3: **Training pipeline of Representation Forcing.** *Left:* The decoder processes a unified sequence of text tokens (T), representation tokens (R), and pixel patches (P) within a shared transformer. Text and representation tokens are predicted autoregressively under next-token prediction ($\mathcal{L}_{\text{LM}}$ and $\mathcal{L}_{\text{Rep}}$), while pixel patches are generated via bidirectional diffusion from noise ($\mathcal{L}_{\text{FM}}$). The image encoder provides continuous visual features to the transformer for understanding tasks. *Right:* For generation training, an EMA copy of the image encoder extracts features from the ground-truth image, which are discretized via online quantization into representation tokens. These tokens provide both the training targets for $\mathcal{L}_{\text{Rep}}$ and the teacher-forcing inputs at R positions. At inference, the right panel is bypassed entirely: the decoder predicts representation tokens from the text prompt alone, and these tokens remain in context to guide pixel-space diffusion.

Pixel-Space Generation. State-of-the-art diffusion models [3, 42] typically operate in the latent space of a pretrained VAE [27], which reduces compute and enables high-resolution synthesis but prevents end-to-end training. A recent line of work has explored generating directly in pixel space [13, 23], enabling end-to-end learning from raw data: JiT [28] demonstrates that plain Vision Transformers with x-prediction can generate on raw pixels, and other approaches explore alternative pixel-space architectures [8, 24, 49]. These methods mostly focus on standalone generation on the ImageNet [12] dataset. In UMMs, the broader image distribution and richer text conditioning leave naive pixel-space diffusion with a clear quality gap; Representation Forcing closes this gap by providing a structural scaffold from the model's own understanding encoder.

Representation Learning for Generation. Several works explore richer visual representations to improve generation. REPA [61] aligns intermediate diffusion features with frozen pretrained representations to accelerate training convergence; RAE and related works [43, 51, 65] goes further by replacing the VAE entirely with frozen pretrained encoders such as DINOv2 [37] and SigLIP [64], providing a semantically richer latent space. Closely related, Latent Forcing [1] produces high-level structure before pixels by jointly diffusing pixels and frozen DINOv2 latents under separate noise schedules. Despite their differences, all these methods still rely on a frozen, separately pretrained representation space. Representation Forcing instead unfreezes this space: the understanding encoder is trained jointly and end-to-end with the rest of the model, and its features are discretized online into representation tokens that the decoder learns to predict autoregressively. High-level visual structure thus becomes a native output learned within the model rather than inherited from a fixed external space, consistent with our goal of a bottleneck-free, end-to-end UMM.

3 Representation Forcing

The design principle behind Representation Forcing is simple: in understanding, the encoder maps images to representations that capture high-level structure; in generation, the decoder mirrors this by predicting representations from text alone, before rendering them into pixels. In Sec. 3.1, we describe where these representations come from and how they are formulated. In Sec. 3.2, we show how the decoder predicts them to guide pixel-space generation. In Sec. 3.3, we present the overall architecture and training objective. The full training pipeline is illustrated in Figure 3.

3.1 Representations from Understanding

Rather than relying on an external latent space, we seek an intermediate representation that captures high-level structure *from within the model itself*, so that pixel-space diffusion can focus on low-level rendering. The understanding encoder provides a natural source: its features, trained for visual comprehension, already encode the structural content we need. The encoder is jointly trained with the rest of the model. To make its features predictable by the decoder under the same next-token objective as text, and easy to sample at inference, we discretize them into a sequence of *visual representation tokens* via vector quantization. Beyond enabling unified training, discretization encourages the representations to retain high-level structure while discarding fine-grained details, which achieves the factorization we seek between representation prediction and pixel rendering. We validate this choice against continuous regression in Sec. 4.4.

We perform this discretization through online vector quantization, which requires no separate pre-trained tokenizer. Since the encoder is jointly trained, its features evolve throughout training; we therefore extract features from an exponential moving average (EMA) copy of the encoder, providing slow-moving targets that keep the discrete assignments stable. Specifically, we use patch-level features from the last layer of the EMA encoder, before the final norm.

We maintain a codebook of K prototype embeddings, and tokenization proceeds in three steps. First, we compute the cosine similarity between each patch-level feature and all K prototypes, yielding a score matrix over the batch. Second, rather than greedily assigning each feature to its most similar prototype—which would let a few popular prototypes absorb most features and collapse the codebook—we balance the score matrix with a Sinkhorn–Knopp normalization following SwAV [4], which softly constrains every prototype to receive an equal share of the batch. Third, each feature is assigned to its highest-scoring prototype under the balanced scores, producing a discrete token index: intuitively, a feature keeps its most similar prototype unless that prototype is over-subscribed, in which case it falls back to its next-best match. Finally, the codebook is updated online with EMA updates following VQ-VAE [47], where each prototype moves toward the mean of the features assigned to it. This yields one representation token per spatial patch, forming a sequence that mirrors the spatial layout of the image and can be predicted in raster-scan order.

3.2 Generating Pixels via Predicted Representations

With the representation tokens defined, we now describe how the decoder uses them to generate images. During training, the EMA encoder provides the representation tokens as ground-truth targets, and the decoder learns to predict them autoregressively under cross-entropy loss, within the same next-token prediction stream as text. At inference, the encoder is no longer involved: the decoder produces the representation tokens from the text prompt alone. We call this process **Representation Forcing**: on one hand, the understanding encoder's representations force the decoder to learn the same high-level visual structure; on the other hand, the decoder's predicted representations force the pixel generation process to follow the intended semantic layout.

Once predicted, the representation tokens remain in the sequence and serve as in-context conditioning for pixel-space generation; the pixel patches form the final image, while the representation tokens themselves are not part of the visible output. The same backbone performs this generation through flow matching. Following JiT [28], we adopt x-prediction with velocity loss. Given clean patches $\mathbf{x}$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, noisy patches at time $t \in [0, 1]$ are

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

The decoder predicts $\mathbf{x}_\theta$, and the flow-matching loss is

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\big\|\mathbf{v}_\theta - \mathbf{v}\big\|^2, \tag{2}$$

where $\mathbf{v} = \mathbf{x} - \boldsymbol{\epsilon}$ and $\mathbf{v}_\theta = (\mathbf{x}_\theta - \mathbf{z}_t)/(1 - t)$.

At the sequence level, as shown in Figure 3, the model processes a unified token sequence structured as [text tokens, representation tokens, pixel patches]. Within the shared self-attention of the backbone, text and representation tokens follow causal attention as in standard autoregressive modeling, while the noisy pixel patches attend bidirectionally to each other and causally to all preceding text and representation tokens. This attention pattern is what makes the representation tokens act as in-context conditioning: the high-level structure they encode flows into pixel generation through standard self-attention, without any additional cross-attention or injection module.

3.3 Training and Inference

Our model adopts the Mixture-of-Transformers (MoT) architecture following BAGEL [11]. All tokens interact through shared multi-head self-attention over the full sequence, but each token is processed by modality-specific attention projections and feed-forward experts based on its type. We maintain three groups of experts: one for multimodal understanding, one for representation token prediction, and one for pixel generation. Each representation token is embedded as the sum of two learnable embeddings: a 2D spatial position embedding, and a token identity embedding indexed by the discrete token ID. The latter is stored in a separate K-entry table.

The model is trained end-to-end with a combined objective:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \mathcal{L}_{\text{FM}} + \mathcal{L}_{\text{Rep}}, \tag{3}$$

where $\mathcal{L}_{\text{LM}}$ is the cross-entropy loss for text next-token prediction, $\mathcal{L}_{\text{Rep}}$ is the cross-entropy loss for representation token prediction, and $\mathcal{L}_{\text{FM}}$ is the flow-matching loss for pixel generation following the x-prediction formulation in Sec. 3.2. To support classifier-free guidance at inference, we independently drop the text conditioning and the representation token sequence with probability 0.1 during training.

At inference, generation proceeds in two stages. The decoder first produces the full representation token sequence autoregressively from the text prompt. Conditioned on both the text and the predicted representation tokens, the decoder then performs iterative denoising from Gaussian noise to synthesize the final image directly in pixel space. Classifier-free guidance is applied to both conditions.

4 Experiments

4.1 Experimental Setup

Architecture. Our model is initialized from Qwen3-30B-A3B [59], a pretrained Mixture-of-Experts (MoE) language model, and follows the Mixture-of-Transformers (MoT) architecture [11] with three modality-specific pathways: understanding, representation prediction, and pixel generation. The image encoder is DINOv3 ViT-H+/16 [44] with NaViT-style variable-resolution support [10], jointly trained with the rest of the model. The understanding and pixel-generation pathways both adopt the Qwen3-30B-A3B MoE architecture (30.5B parameters each, 3.3B activated per token), and the representation pathway is a dense branch of 2.8B parameters. In total, the model contains 64.7B parameters including the image encoder, while only $\sim$3.3B are activated per token. For the representation tokens, we use a codebook of $K{=}16{,}384$ prototypes for the online vector quantization and extract the targets at a low resolution: the input to the representation branch has its longer side capped at 288 pixels, and its patch features are $2{\times}2$ pooled into at most $9{\times}9{=}81$ tokens. Since the representations only need to carry coarse structure, their sequence length does not grow with the output resolution. For pixel-space generation, we use a $16{\times}16$ patch size and adopt x-prediction with velocity loss [28].

Data. We follow the data construction and filtering pipeline of BAGEL [11], training on a mixture of (1) text-only data for language modeling and (2) large-scale text–image pairs covering both image-to-text understanding (general VQA, document comprehension, spatial reasoning) and text-to-image generation.

Training. We adopt a three-stage training strategy following [11]: (i) *alignment*: with the backbone and encoder frozen, we train only the MLP connector for 10K iterations; (ii) *joint pre-training*: we unfreeze all components and jointly optimize on text and text–image pairs at resolutions up to 256 for 50K iterations; (iii) *continued training*: we extend resolutions up to 1024 for 20K iterations. Throughout training, image resolutions are sampled dynamically within the per-stage maximum and packed via NaViT-style variable-resolution batching. More details are in the appendix.

Baselines. For controlled comparison, we train VAE-based variants using the WanX-2.1 VAE [52], replacing the pixel input/output with VAE latents while keeping the rest of the architecture, training data, and optimization identical. The four variants in our controlled study (Pixel, Pixel+RF, VAE, VAE+RF) differ *only* in (a) the generation pathway and (b) whether RF is enabled. Ablations are conducted at 256 resolution; main results are reported at 1024 resolution.

Table 1: **Evaluation of text-to-image generation.** † refers to methods using LLM rewriter. Our pixel-space model with Representation Forcing (RF-Pixel) matches state-of-the-art VAE-based unified models on both GenEval and DPG-Bench, without relying on any pretrained VAE.

	GenEval							DPG
Model	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑	Overall↑
Generation Only								
PixArt-α [7]	0.98	0.50	0.44	0.80	0.08	0.07	0.48	71.11
SDv2.1 [42]	0.98	0.51	0.44	0.85	0.07	0.17	0.50	68.09
DALL-E 2 [41]	0.94	0.66	0.49	0.77	0.10	0.19	0.52	–
SDXL [39]	0.98	0.74	0.39	0.85	0.15	0.23	0.55	74.65
DALL-E 3 [2]	0.96	0.87	0.47	0.83	0.43	0.45	0.67	83.50
SD3-Medium [16]	0.99	0.94	0.72	0.89	0.33	0.60	0.74	84.08
FLUX.1-dev† [3]	0.98	0.93	0.75	0.93	0.68	0.65	0.82	84.00
Seedream 3.0 [19]	0.99	0.96	0.91	0.93	0.47	0.80	0.84	88.27
Z-Image-Turbo [63]	1.00	0.95	0.77	0.89	0.65	0.68	0.82	84.86
Qwen-Image [53]	0.99	0.92	0.89	0.88	0.76	0.77	0.87	88.32
Unified Models								
Chameleon [5]	–	–	–	–	–	–	0.39	–
LWM [31]	0.93	0.41	0.46	0.79	0.09	0.15	0.47	–
SEED-X [20]	0.97	0.58	0.26	0.80	0.19	0.14	0.49	–
TokenFlow-XL [40]	0.95	0.60	0.41	0.81	0.16	0.24	0.55	73.38
ILLUME [48]	0.99	0.86	0.45	0.71	0.39	0.28	0.61	–
Janus [54]	0.97	0.68	0.30	0.84	0.46	0.42	0.61	–
Transfusion [66]	–	–	–	–	–	–	0.63	–
Emu3† [50]	0.99	0.81	0.42	0.80	0.49	0.45	0.66	81.60
Show-o [57]	0.98	0.80	0.66	0.84	0.31	0.50	0.68	–
Show-o2 [58]	1.00	0.87	0.58	0.92	0.52	0.62	0.76	86.14
Janus-Pro-7B [9]	0.99	0.89	0.59	0.90	0.79	0.66	0.80	84.19
MetaQuery-XL† [38]	–	–	–	–	–	–	0.80	82.05
BLIP3-o [6]	–	–	–	–	–	–	0.84	81.60
UniWorld-V1† [30]	0.98	0.93	0.81	0.89	0.74	0.71	0.84	81.38
OmniGen2† [55]	0.99	0.96	0.74	0.98	0.71	0.75	0.86	83.57
BAGEL [11]	0.99	0.94	0.81	0.88	0.64	0.63	0.82	85.07
BAGEL† [11]	0.98	0.95	0.84	0.95	0.78	0.77	0.88	–
RF-Pixel (ours)	0.99	0.93	0.84	0.89	0.74	0.66	0.84	84.15
RF-Pixel (ours)†	0.98	0.95	0.88	0.87	0.92	0.70	0.88	-

4.2 Image Generation

We evaluate our pixel-space model with RF (denoted RF-Pixel in Table 1) on two standard text-to-image benchmarks: GenEval [21] for compositional generation and DPG-Bench [25] for dense prompt following. We compare against both generation-only models and unified multimodal models. Existing unified models all rely on separately pretrained generative components—either a frozen VAE/VQVAE within a single backbone (e.g., BAGEL, Show-o, Janus-Pro), or an external diffusion module conditioned on LLM-predicted features (e.g., BLIP3-o, MetaQuery, SEED-X). In contrast, our model operates entirely in pixel space without any separately pretrained generative module.

As shown in Table 1, without LLM rewriter, RF-Pixel achieves a GenEval overall score of 0.84, slightly outperforming the BAGEL baseline (0.82) and matching unified models such as BLIP3-o (0.84). With LLM rewriter, RF-Pixel reaches 0.88, matching the state-of-the-art among unified models. On DPG-Bench, RF-Pixel scores 84.15 without rewriter, comparable to state-of-the-art VAE-based unified models. These results demonstrate that Representation Forcing effectively closes the quality gap between pixel-space and VAE-based generation, enabling an end-to-end pixel-space unified model to match VAE-based counterparts.

Table 2: **Impact of RF on understanding.** We compare understanding performance with and without RF under both VAE-based and pixel-space generation. MME* reports the average accuracy across all perception and cognition questions. +x/−x denotes the change from adding RF. RF improves both settings, with large gains on general visual understanding and only slight reductions on document-oriented tasks. Pixel+RF benefits more from RF than VAE+RF (6/8 vs. 5/8 benchmarks improved), and outperforms VAE+RF on 6 out of 8 benchmarks.

	General Visual Understanding					Document & Diagram		
	MMMU	HalluBench	MME*	BLINK	RealWorldQA	AI2D	DocVQA	ChartQA
VLM-only	56.2	65.0	79.7	56.2	65.8	90.3	89.3	86.0
VAE	51.0	55.7	71.3	52.2	65.2	90.7	90.0	78.8
VAE+RF	49.6 −1.4	61.3 +5.6	79.3 +8.0	52.9 +0.7	66.6 +1.4	87.8 −2.9	88.3 −1.7	80.5 +1.7
Pixel	49.9	63.7	76.6	49.4	63.1	85.8	90.0	81.7
Pixel+RF	54.2 +4.3	64.8 +1.1	80.2 +3.6	53.0 +3.6	65.8 +2.7	90.3 +4.5	88.0 −2.0	81.3 −0.4

4.3 Image Understanding

Beyond generation quality, we compare how different generation pathways affect understanding performance. As shown in Table 2, we train four generation variants—VAE-based and pixel-space, each with and without RF—on top of the same first-stage VLM baseline, using identical architecture and training data. Since we focus on pretraining-stage comparison, no post-training is applied. We include the VLM-only baseline as a reference.

We evaluate on 8 benchmarks spanning two categories: (1) general visual understanding, including MMMU [62], HallusionBench [22], MME [17], BLINK [18], and RealWorldQA [56], which test general visual understanding, hallucination robustness, and real-world perception; and (2) document and diagram understanding, including AI2D [26], DocVQA [36], and ChartQA [35], which test structured visual comprehension.

RF consistently improves understanding under both generation pathways. For pixel-space generation, RF improves 6 of 8 benchmarks, with substantial gains on MMMU ($+4.3$), MME ($+3.6$), BLINK ($+3.6$), AI2D ($+4.5$), and RealWorldQA ($+2.7$), with small reductions on DocVQA (-2.0) and ChartQA (-0.4). For VAE-based generation, RF improves 5 of 8 benchmarks, notably HalluBench ($+5.6$) and MME ($+8.0$). The improvements are concentrated on benchmarks that test high-level visual understanding—object recognition, spatial comprehension, and scene-level perception—which aligns with the nature of the representation tokens: they are derived from the understanding encoder and encode semantic structure rather than fine-grained details. The reductions on DocVQA and ChartQA, which rely heavily on precise text recognition and layout parsing, suggest that these capabilities are less directly supported by representation-level guidance.

Pixel+RF outperforms VAE+RF on 6 out of 8 benchmarks. We attribute this to the removal of the external VAE latent space, which allows understanding and generation to share a single representation space more tightly. This result aligns with our broader motivation of moving toward bottleneck-free unified multimodal models.

4.4 Ablation Studies

We conduct ablation studies to analyze the key design choices of Representation Forcing in Table 3.

Representations are critical for pixel-space generation. We evaluate RF under both pixel-space and VAE-based generation, with results shown in Table 3a. All four variants share the same architecture and training setup, differing only in the generation pathway and the presence of RF. Without RF, pixel-space generation scores only 0.25 on GenEval, while VAE-based generation reaches 0.52. As illustrated in Figure 4, the pixel-space model without RF tends to produce images with poor structure, such as distorted objects and incoherent compositions, suggesting that the model struggles to jointly learn high-level layout and low-level detail from raw pixels alone. With RF, pixel-space generation jumps to 0.76, matching the VAE-based counterpart at 0.77, and the generated images become structurally coherent. RF also improves VAE-based generation from 0.52 to 0.77, confirming that explicit representation guidance benefits both settings, though the effect is most pronounced in pixel space.

Table 3: **Ablation studies.** All experiments are conducted under pixel-space generation at 256 resolution unless otherwise noted. (a) Effect of RF under both pixel-space and VAE-based generation (Δ: gain from RF). (b) Comparison between RF (decoder prediction) and REPA (auxiliary alignment) as representation guidance strategies. (c) Where representation tokens are used: during training (Train) and/or as in-context conditioning for pixel diffusion at inference (Infer.). (d) Discrete vs. continuous representation token formulation. (e) Effect of codebook size K. (f) Choice of understanding encoder evaluated on VLM-only benchmarks.

(a) **RF for pixel and VAE gen.**

	Pixel	VAE
w/o	0.25	0.52
w/	**0.76**	**0.77**
Δ	+0.51	+0.25

(b) **Prediction vs. alignment.**

	GenEval↑
w/o	0.25
REPA	0.43
RF (ours)	**0.76**

(c) **Where rep. tokens are used.**

Train	Infer.	GenEval↑
×	×	0.25
✓	×	0.59
✓	✓	**0.76**

(d) **Discrete vs. continuous (causal).**

	GenEval↑
w/o	0.25
Continuous	0.26
Discrete	**0.76**

(e) **Rep. token codebook size.**

	GenEval↑
K=16384	0.76
K=32768	**0.77**

(f) **Understanding encoder.**

	MMMU	HalluBench	MME*	BLINK	RealWorldQA
SigLIP2	**56.3**	64.4	76.8	53.4	62.5
DINOv3	56.2	**65.0**	**79.7**	**56.2**	**65.8**

Decoder prediction outperforms auxiliary alignment. RF is not the only way to incorporate visual representations into generation. REPA [61] takes an alternative approach by adding an auxiliary loss on the diffusion side that encourages the model's intermediate features to align with encoder representations. We apply both methods to the same pixel-space UMM under identical training conditions, using the same visual encoder (DINOv3 [44]) as the representation source, with results shown in Table 3b. Without any guidance, pixel-space generation scores only 0.25. REPA improves this to 0.43, confirming that representation guidance helps, but the gain remains limited. RF achieves 0.76, substantially outperforming REPA. We attribute this difference to where the representation enters the generation process. REPA encourages feature similarity as a side objective, but the aligned features do not explicitly condition the generation output during inference. RF places predicted representations directly in the decoder's sequence, where pixel patches attend to them through shared self-attention. In high-dimensional pixel space, this direct conditioning proves more effective than implicit feature alignment.

Representation tokens help in both training and inference. We further verify this attribution by disentangling the two roles of representation tokens in RF: prediction targets during training and in-context conditioning at inference (Table 3c). We keep RF training unchanged but withhold the predicted tokens from the pixel-diffusion context at inference, a setting the model has seen during training since the representation token sequence is randomly dropped for classifier-free guidance (Sec. 3.3). This lowers GenEval from 0.76 to 0.59, confirming that in-context conditioning at inference accounts for a substantial part of the gain. The remaining improvement over the baseline (0.25 to 0.59) reflects the benefit of the training objective itself, which alone already surpasses REPA's alignment objective (0.43).

Discrete prediction outperforms continuous regression. The representation tokens can be formulated in different ways. Beyond our discrete approach with autoregressive cross-entropy prediction, an alternative is to predict continuous features directly: for this baseline, we attach a MAR-style diffusion head [29] that denoises each representation position independently, while the context across positions remains causal. We compare both formulations under the same setting (Table 3d). Continuous regression scores only 0.26, providing no improvement over the baseline without RF, while discrete tokens achieve 0.76. We attribute this to two factors. First, causally predicting high-dimensional continuous vectors is prone to error accumulation, where small prediction errors at early positions compound and degrade later predictions. Discrete tokens avoid this by reducing each position to a categorical choice, which is more robust under sequential prediction. We also tried denoising the continuous representations with bidirectional attention, similar to Latent Forcing [1], which avoids such accumulation; this variant also works, but still underperforms the discrete formulation with a gap of 0.1. Second, discretization naturally encourages the representations to retain high-level



Figure 4: **Qualitative comparison of pixel-space generation with and without RF.** Without RF, the model tends to produce images with poor structure, such as distorted object shapes and incoherent compositions. With RF, the model generates more coherent structures by first predicting high-level visual representations before pixel rendering, which provides explicit structural guidance for the diffusion process.

structure while discarding fine-grained detail—precisely the factorization that RF is designed to provide. Continuous targets preserve more low-level information, undermining this factorization.

Codebook size. The codebook size K controls the expressiveness of the discrete representation space. We compare K=16,384 and K=32,768 in Table 3e. The two settings perform comparably (0.76 vs. 0.77), suggesting that the model is not sensitive to codebook size within this range. We use K=16,384 in all other experiments, which is consistent with common practice in vector quantization.

Understanding encoder. The choice of understanding encoder determines the quality of the representations that RF learns to predict. We compare SigLIP2 [46], a contrastive vision-language model, and DINOv3 [44], a self-supervised vision model, as the image encoder backbone (Table 3f). We find DINOv3 outperforms SigLIP2 on 4 out of 5 understanding benchmarks in our setting. We attribute DINOv3's advantage to its self-supervised training objective, which produces features with richer spatial and structural information compared to SigLIP2's language-aligned features. This aligns with RF's design: the representation tokens are meant to capture visual structure, which benefits from an encoder that prioritizes spatial fidelity over text alignment. We use DINOv3 as the default encoder.

5 Discussion

Limitations. Due to computational constraints, our model is initialized from a pretrained large language model rather than trained from scratch on multimodal data. While this provides a strong language-grounded starting point, fully from-scratch multimodal pretraining may yield richer joint representations and is an important direction for future work. We also focus on still-image generation and do not extend RF to video or other temporal modalities.

Conclusion. In this paper, we present Representation Forcing, a simple method for pixel-space image generation in unified multimodal models. The key idea is to let the decoder predict its own understanding representations as autoregressive targets before rendering pixels, providing structural guidance to pixel-space diffusion from within the same sequence. Our experiments show that this single mechanism is enough to close the quality gap with VAE-based generation while also improving multimodal understanding. The same representation serves both directions—interpreting visual inputs and guiding their generation—pointing to a closer integration of perception and generation. We see RF as a step toward fully end-to-end native multimodal learning, where all multimodal capabilities are acquired directly from raw inputs within a single model. We hope this work inspires further research in this direction.

Acknowledgments and Disclosure of Funding

This work is supported by the Research Grants Council of Hong Kong through NSFC-RGC Joint Research Scheme under grant N_HKU769/25.

References

[1] Alan Baade, Eric Ryan Chan, Kyle Sargent, Changan Chen, Justin Johnson, Ehsan Adeli, and Li Fei-Fei. Latent forcing: Reordering the diffusion trajectory for pixel-space image generation. *arXiv preprint arXiv:2602.11401*, 2026.

[2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. *OpenAI Technical Report, https://cdn.openai.com/papers/dall-e-3.pdf*, 2023.

[3] Black Forest Labs. FLUX. `https://github.com/black-forest-labs/flux`, 2024.

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

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

[6] 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.

[7] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024.

[8] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. PixelFlow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 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] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n' Pack: NaViT, a vision transformer for any aspect ratio and resolution. In *NeurIPS*, 2023.

[11] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

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

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, volume 34, pages 8780–8794, 2021.

[14] Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026.

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

[16] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

[17] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[18] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *ECCV*, 2024.

[19] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xuanda Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Xin Xia, Xuefeng Xiao, Zhonghua Zhai, Xinyu Zhang, Qi Zhang, Yuwei Zhang, Shijia Zhao, Jianchao Yang, and Weilin Huang. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025.

[20] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. SEED-X: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024.

[21] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, 2023.

[22] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusion-Bench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *CVPR*, 2024.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, 2020.

[24] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (SiD2): 1.5 FID on ImageNet512 with pixel-space diffusion. In *CVPR*, 2025.

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

[26] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016.

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

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

[29] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

[30] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. UniWorld-V1: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

[31] Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with blockwise RingAttention. *arXiv preprint arXiv:2402.08268*, 2024.

[32] Zhiheng Liu, Weiming Ren, Xiaoke Huang, Shoufa Chen, Tianhong Li, Mengzhao Chen, Yatai Ji, Sen He, Jonas Schult, Belinda Zeng, Tao Xiang, Wenhu Chen, Ping Luo, Luke Zettlemoyer, and Yuren Cong. Tuna-2: Pixel embeddings beat vision encoders for multimodal understanding and generation. *arXiv preprint arXiv:2604.24763*, 2026.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[34] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, Liang Zhao, Yisong Wang, Jiaying Liu, and Chong Ruan. JanusFlow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *CVPR*, 2025.

[35] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of ACL*, 2022.

[36] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. *arXiv preprint arXiv:2007.00398*, 2020.

[37] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

[38] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between modalities with MetaQueries. *arXiv preprint arXiv:2504.06256*, 2025.

[39] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

[40] 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 *CVPR*, 2025.

[41] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

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

[43] Minglei Shi, Haolin Wang, Wenzhao Zheng, Ziyang Yuan, Xiaoshi Wu, Xintao Wang, Pengfei Wan, Jie Zhou, and Jiwen Lu. Latent diffusion model without variational autoencoder. *arxiv: 2510.15301*, 2025.

[44] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

[45] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. 2023.

[46] 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.

[47] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017.

[48] Chunwei Wang, Guansong Lu, Junwei Yang, Runhui Huang, Jianhua Han, Lu Hou, Wei Zhang, and Hang Xu. ILLUME: Illuminating your LLMs to see, draw, and self-enhance. In *ICCV*, 2025.

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

[50] 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.

[51] Yuqing Wang, Chuofan Ma, Zhijie Lin, Yao Teng, Lijun Yu, Shuai Wang, Jiaming Han, Jiashi Feng, Yi Jiang, and Xihui Liu. Cubic discrete diffusion: Discrete visual generation on high-dimensional representation tokens. In *CVPR*, 2026.

[52] WanTeam, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[53] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[54] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *CVPR*, 2025.

[55] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. OmniGen2: Towards instruction-aligned multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

[56] xAI. RealWorldQA. `https://huggingface.co/datasets/xai-org/RealworldQA`, 2024.

[57] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *ICLR*, 2025.

[58] Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. In *NeurIPS*, 2025.

[59] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[60] Ceyuan Yang, Zhijie Lin, Yang Zhao, Fei Xiao, Hao He, Qi Zhao, Chaorui Deng, Kunchang Li, Zihan Ding, Yuwei Guo, Fuyun Wang, Fangqi Zhu, Xiaonan Nie, Shenhan Zhu, Shanchuan Lin, Hongsheng Li, Weilin Huang, Guang Shi, and Haoqi Fan. Context unrolling in omni models. *arXiv preprint arXiv:2604.21921*, 2026.

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

[62] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *CVPR*, 2024.

[63] Z-Image Team. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

[64] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

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

[66] 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. In *ICLR*, 2025.

A Implementation Details

Training. We train using AdamW [33] (β_1=0.9, β_2=0.95, ϵ=10^{-8}, weight decay 0.1, gradient clipping 1.0). The learning rate follows linear warmup followed by a constant schedule, with a base rate of 5×10^{-5} in Stages 1–2 and 2.5×10^{-5} in Stage 3 for high-resolution stability. Newly initialized generation-related parameters use a $4\times$ multiplier on the base rate, while the LLM backbone keeps the base rate. Each GPU processes sequences of 32,768 tokens, packed via NaViT-style variable-resolution batching. The online vector quantization codebook is updated via momentum (decay 0.9999, 1 Sinkhorn-Knopp iteration, temperature 0.5); pseudocode is provided in Algorithm 1. For classifier-free guidance, we independently drop the text condition and the entire representation token sequence, each with probability 0.1.

Inference. We maintain an exponential moving average of model parameters with decay 0.9999 and perform inference using the EMA model. Generation proceeds in two stages: the decoder first produces the full representation token sequence autoregressively from the text prompt using top-k sampling, then denoises Gaussian noise into pixel patches over 25 flow-matching steps with dynamic timestep shifting [16], conditioned on the text and predicted representation tokens. We apply two-condition CFG with w_{rep}=2.0 for representation token sampling and w_{pix}=3.0 for pixel patch denoising.

B Online Vector Quantization Algorithm

Algorithm 1 Pseudocode of Online Vector Quantization (PyTorch-like).

```
# f_m: EMA understanding encoder
# X: batch of samples (BxHxWx3)
# C: visual prototypes (KxD)
# m: momentum (default 0.9999)
# t: temperature (default 0.5)

Z = f_m(X).view(B * L, D)           # extract continuous features via EMA encoder
Z = normalize(Z, dim=1)
score = matmul(Z, C.T) / t          # pairwise cosine sims: (B*L, K)
score = softmax(score, dim=1)       # row-normalize
score = softmax(score, dim=0)       # column-normalize (Sinkhorn-Knopp, 1 iteration)
A = argmax(score, dim=1)            # discrete assignment: (B*L,)

N_k, C_new = zeros(K), zeros(K, D)
A_c = A.view(B * L, 1).expand(B * L, D)
C_new = scatter_add(C_new, dim=0, index=A_c, src=Z)
N_k = scatter_add(N_k, dim=0, index=A, src=ones(B * L))
C_new = normalize(C_new / N_k.view(K, 1), dim=1) # new prototypes from assignments
C = m * C + (1 - m) * C_new          # momentum update
C = normalize(C, dim=1)
```

C Broader Impact

Like other text-to-image generation systems, Representation Forcing could potentially be misused to generate misleading or harmful visual content, including disinformation, non-consensual imagery, or deepfakes. Standard safeguards used for unified multimodal models—including safety filters, output watermarking, and controlled access—apply to RF-based systems.

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: The abstract and introduction accurately reflect our contributions: (1) Representation Forcing as a method to close the quality gap between pixel-space and VAE-based generation in unified multimodal models, and (2) demonstrating that RF benefits both generation and understanding. Limitations are discussed in the Limitations paragraph.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations explicitly: due to computational constraints, our model is initialized from a pretrained LLM rather than trained from scratch on multimodal data, and we focus on still-image generation without extending to video or other temporal modalities.

   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: The paper does not include theoretical results.

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 fully disclose the algorithm in Section 3 (method), implementation details (optimizer, learning rates, EMA, online vector quantization, classifier-free guidance, and inference procedure) in Appendix A, and pseudocode for the online vector quantization mechanism in Algorithm 1.

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: [No]

   Justification: Code and data are not open-sourced at this time. We may consider releasing them, subject to internal approval. The method is described in detail in the main paper and Appendix A, with pseudocode for the online vector quantization mechanism (Algorithm 1), providing a clear path to reproducibility.

   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: Section 4.1 describes the main experimental setup including model architecture, data, three-stage training schedule, and baselines. Appendix A provides full implementation details including optimizer hyperparameters, learning rate schedules, sequence length, EMA, online vector quantization parameters, classifier-free guidance, and inference procedure.

   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: We do not report error bars due to the substantial computational cost of training unified multimodal models at this scale. This is consistent with prior work in this area, such as BAGEL.

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: Training was performed on 64 H20 GPUs across the three training stages.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research conforms to it.

   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: [Yes]

Justification: We discuss broader impacts in Appendix.

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: We do not release pretrained model checkpoints or training data in this submission, so no specific safeguards are required.

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: All existing models, datasets, and benchmarks used are cited in the paper, including Qwen3-A3B, DINOv3, BAGEL, WanX-2.1 VAE, JiT, and evaluation benchmarks (GenEval, DPG-Bench, MMMU, HallusionBench, MME, BLINK, RealWorldQA, AI2D, DocVQA, ChartQA). We use these assets according to their respective 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: The 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 research does not involve crowdsourcing or 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: This research does not involve human subjects, so IRB approval is not applicable.

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: This research uses a pretrained LLM (Qwen3-A3B) as a standard backbone for the unified multimodal model, clearly described in the methodology. We do not use LLMs in any non-standard or originally-developed manner.

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.