STARFlow2: Bridging Language Models and Normalizing Flows for Unified Multimodal Generation

Ying Shen[†*] **Tianrong Chen**[◇] **Yuan Gao**[◇] **Yizhe Zhang**[◇] **Yuyang Wang**[◇]
Miguel Angel Bautista[◇] **Shuangfei Zhai**[◇] **Josh Susskind**[◇] **Jiatao Gu**[◇⋆]
[◇]Apple [†]University of Illinois Urbana-Champaign

Abstract

Unified multimodal models that understand, reason over, and generate interleaved text–image sequences remain structurally fragmented: existing approaches either sacrifice visual fidelity through discrete tokenization, impose structural asymmetry by combining causal text generation with iterative diffusion-based denoising, or degrade pretrained understanding when adapting vision-language models for generation. We observe that autoregressive normalizing flows are autoregressive Transformers—sharing the same causal mask, KV-cache mechanism, and left-to-right structure as LLMs—making them the most natural paradigm for truly unified multimodal generation that is continuous, single-pass, and purely causal. We present STARFlow2, built on the Pretzel architecture that vertically interleaves a frozen pretrained VLM stream with a TARFlow stream via residual skip connections, both operating under the same causal mask. This design simultaneously preserves pretrained multimodal understanding, enables high-fidelity continuous image generation, and achieves structural unification under a single causal mechanism. Combined with a deep-shallow flow design and a unified FAE latent space, STARFlow2 supports cache-friendly interleaved generation where both text and visual outputs directly enter the KV-cache without re-encoding. Experiments demonstrate strong performance across image generation and multimodal understanding benchmarks, validating autoregressive flows as a viable foundation for unified multimodal modeling.

1 Introduction

Unified multimodal models that perceive, reason over, and generate interleaved text–image sequences have emerged as a key goal toward general-purpose AI [67; 56; 11; 62]. By treating images and text as interleaved steps in a shared generation sequence, such models can support interactive multi-turn editing [17; 68] and problem solving with visual thoughts [24; 10].

Despite growing interest, existing "unified" multimodal models are not truly unified in their generation mechanisms. One line of work discretizes images into tokens and trains a single language model over the joint text-image sequence [56; 31; 9; 6]. While architecturally elegant, this approach sacrifices the continuous nature of visual data—quantization introduces information loss and limits generation fidelity [40; 57]. A more popular paradigm combines autoregressive language modeling for text with diffusion-based denoising for images within a single backbone [67; 61; 62; 50; 38; 11]. However, these two generation mechanisms are structurally different: text tokens are generated causally under a left-to-right mask, while images require iterative denoising often with different attention patterns. Generated images cannot directly enter the causal KV-cache as reusable context—a separate re-encoding step is needed for interleaved generation. Mixture-of-Transformers (MoT) [32], adopted in

[*]Work done while JG holding a joint affiliation with University of Pennsylvania, and YS working as a research intern at Apple MLR.

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

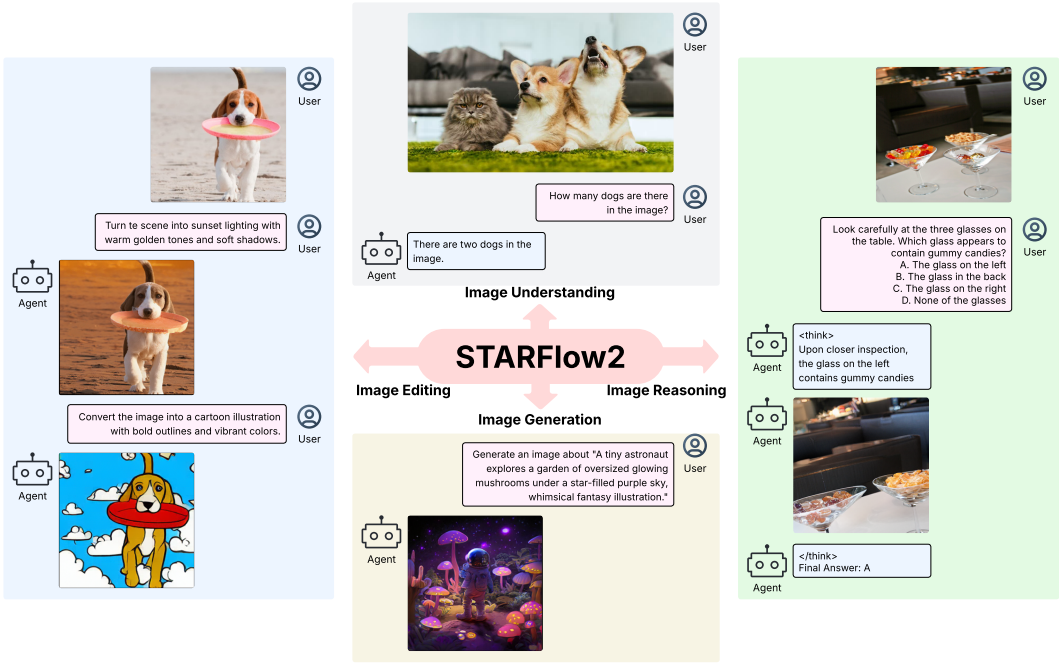


Figure 1: **STARFlow2 as a unified multimodal architecture.** A single model supports image generation, editing, understanding, and reasoning across diverse image-centric tasks.

BAGEL [11], routes different modalities to modality-specific feed-forward parameters while sharing attention. Though this appears unified, it remains two specialized sub-networks sharing only attention within a single Transformer backbone. Moreover, as we show empirically (§ 5.3), MoT faces an inherent dilemma when combined with TARFlow: freezing the VLM leads to poor generation quality, while finetuning the VLM degrades multimodal understanding.

We argue that a truly unified architecture must simultaneously satisfy three desiderata:

- **(D1) Preserve pretrained VLM understanding**—retain the strong multimodal perception and reasoning capabilities of a pretrained vision-language model without degradation from generation training.
- **(D2) High-fidelity continuous image generation**—generate images in continuous latent space without quantization loss, maintaining visual quality comparable to dedicated generative models.
- **(D3) Structurally unified causal generation**—generate both text and images under the same causal mechanism (same mask, same KV-cache, single-pass decoding), without diffusion's iterative denoising or re-encoding overhead.

Discrete tokenization violates (D2); diffusion hybrids violate (D3); and MoT, depending on training strategy, violates either (D1) or (D2).

Recently, STARFlows [65; 19; 20] have shown that normalizing flows, when parameterized by causal Transformers, can generate continuous visual data with quality matching or exceeding diffusion models. Crucially, these models generate token-by-token from left to right—using the *same* causal mask, the *same* KV-cache mechanism, and the *same* autoregressive structure as LLMs. The only difference is the output head: instead of predicting discrete token logits, the flow predicts affine transformation parameters for continuous latents. In other words, there is *no structural gap* between *autoregressive flows* and *language models*—making flows a natural paradigm to satisfy (D2) and (D3) simultaneously: continuous, single-pass, and purely causal.

Building on this insight, we introduce STARFlow2, a unified multimodal model built on the **Pretzel architecture**—named for the characteristic shape formed by its two streams crossing through vertical skip connections (fig. 2). Pretzel vertically interleaves a pretrained VLM stream (for language

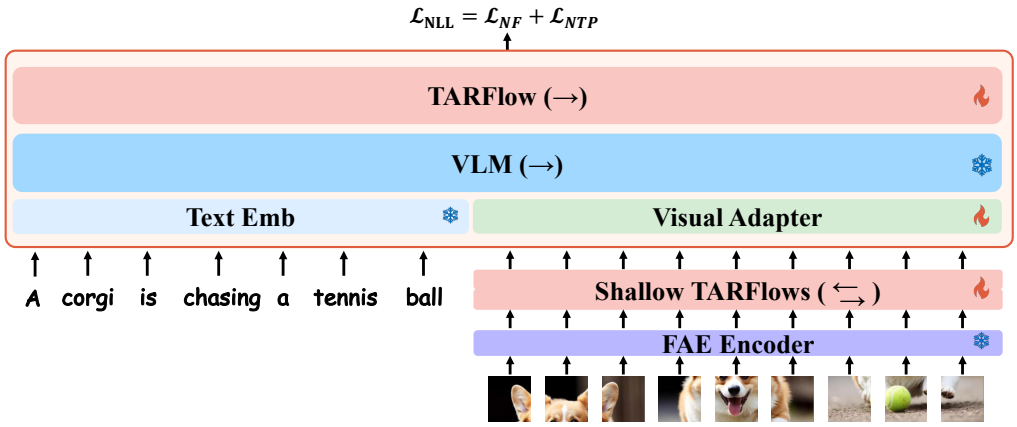


Figure 2: **Overview of the Pretzel architecture in STARFlow2.** A VLM stream and a TARFlow stream are vertically interleaved via crossing skip connections, operating on the same multimodal sequence under a shared causal mask. Shallow TARFlows refine visual latents locally. The model is trained with a unified NLL objective.

modeling and multimodal understanding) with a TARFlow stream (for continuous visual generation) via residual skip connections, satisfying (D1) by keeping the VLM frozen while enabling rich cross-modal interaction. Both streams process the same interleaved multimodal sequence under the same causal mask, achieving true architectural unification (D3). Unlike MoT's horizontal separation—where different tokens route to different parameters—Pretzel interleaves the two streams vertically, allowing both to attend over all tokens and exchange information through skip connections at every position. Combined with a deep-shallow flow design [19] and a unified FAE latent space [16], STARFlow2 supports cache-friendly interleaved text-image generation without visual re-encoding, while maintaining the fidelity of continuous-space generation (D2) and exact likelihood training.

Our contributions are as follows:

- We present STARFlow2, the first unified multimodal framework where both text and image generation employ the same autoregressive Transformer mechanism under the same causal mask, enabling cache-friendly interleaved generation without quantization, iteration, or visual re-encoding (D2, D3).
- We propose the Pretzel architecture that vertically interleaves a frozen pretrained VLM with a TARFlow backbone via residual skip connections—in contrast to MoT's horizontal modality separation—preserving pretrained understanding while enabling rich cross-modal interaction within a single causal sequence model (D1).
- Experiments on multimodal understanding and image generation benchmarks demonstrate that STARFlow2 simultaneously achieves strong performance across all three desiderata, validating autoregressive flows as a foundation for unified multimodal generation.

2 Preliminaries

Unified Multimodal Generation A unified multimodal model processes interleaved text–image sequences $\mathcal{C} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_T)$, where each element $\boldsymbol{c}_t$ is either a discrete text token or a continuous visual latent. The goal is to support both multimodal understanding (image-conditioned text generation) and visual generation (text-conditioned image synthesis) within a single model. Most current approaches build on pretrained vision-language models (VLMs) that already achieve strong multimodal understanding [36; 3], and augment them with image generation capabilities. The central challenge is how to integrate visual generation without degrading the VLM's pretrained understanding or introducing structural asymmetry between modalities.

Feature Auto-Encoder (FAE) STARFlow2 operates in the latent space of a Feature Auto-Encoder (FAE) [16], which provides a compact continuous representation serving both understanding and generation. We train FAE on DINOv2-g/14 [41] features, which we find better suited for generation than SIGLIP-based [66] representations while retaining strong understanding performance. Given an

image, the FAE encoder produces visual latents $\boldsymbol{x} \in \mathbb{R}^{N \times D}$, where N is the number of visual tokens and D is the latent dimensionality. This shared latent space enables a single representation to serve as both input for multimodal understanding and the generation target for normalizing flows.

Autoregressive Normalizing Flows Normalizing flows (NFs) [12; 48; 13; 27; 21] are likelihood-based generative models that learn an invertible mapping between a simple distribution (e.g., a standard Gaussian) and a complex data distribution. In particular, given a continuous input $\boldsymbol{x} \sim p_{\text{data}}, \boldsymbol{x} \in \mathbb{R}^D$, an NF learns a bijection $f_\theta : \mathbb{R}^D \to \mathbb{R}^D$ that maps data $\boldsymbol{x}$ to latents $\boldsymbol{z} = f_\theta(\boldsymbol{x})$. Derived from the change-of-variables formula, NFs can be trained end-to-end via a tractable maximum-likelihood objective:

$$\mathcal{L}_{\text{NF}}(\theta) = -\mathbb{E}_{\boldsymbol{x}} \left[\log p_0(f_\theta(\boldsymbol{x})) + \log \left| \det(J_{f_\theta}(\boldsymbol{x})) \right| \right], \tag{1}$$

where the first term encourages mapping data to high-density regions of a simple prior p_0, and the Jacobian term J_f accounts for the local volume change induced by f_θ, preventing the model from collapsing. Once trained, one automatically obtains a generative model by inverting f_θ, with a sampling process: $\boldsymbol{z} \sim p_0(\boldsymbol{z}), \boldsymbol{x} = f_\theta^{-1}(\boldsymbol{z})$.

Recently, TARFlow-style models [65; 19; 20] have revived normalizing flows for generative modeling by parameterizing them with causal Transformers. Specifically, they instantiate Autoregressive Flows (AFs) [28; 43] by stacking multiple invertible autoregressive flow (AF) blocks with alternating orderings. Given an input presented in the form of a sequence $\boldsymbol{x} \in \mathbb{R}^{N \times D}$, where N is the sequence length and D is the dimension, each AF block applies an affine transform whose parameters are predicted by a causal Transformer under a self-exclusive causal mask for both forward ($\boldsymbol{x} \to \boldsymbol{z}$) and sampling ($\boldsymbol{z} \to \boldsymbol{x}$) process:

$$\boldsymbol{z}_n = \left(\boldsymbol{x}_n - \mu_\theta(\boldsymbol{x}_{<n})\right) / \sigma_\theta(\boldsymbol{x}_{<n}), \quad \boldsymbol{x}_n = \mu_\theta(\boldsymbol{x}_{<n}) + \sigma_\theta(\boldsymbol{x}_{<n}) \cdot \boldsymbol{z}_n, \tag{2}$$

where $\boldsymbol{x}, \boldsymbol{z}$ are the input and output of each block. This can be viewed as "next-token prediction" with affine transformation. STARFlow [19] introduces a deep-shallow architecture, where a deep AF block captures most of the model's capacity, followed by a few shallow AF blocks that further refine the image generation. Note that if we have the deep AF block to follow the left-to-right causal order, it inherits the same causal structure as language models, making them a natural candidate for unifying continuous visual generation with discrete text modeling in an autoregressive manner.

3 STARFlow2

This section details the three components of STARFlow2: the Pretzel architecture that vertically interleaves a pretrained VLM with a TARFlow stream (§ 3.1); the deep-shallow flow design that factorizes visual generation into global multimodal modeling and local refinement (§ 3.2); and the multi-stage training pipeline that progressively activates components (§ 3.3).

3.1 The Pretzel Architecture

The core of STARFlow2 is the Pretzel architecture, which vertically interleaves two autoregressive streams—a pretrained VLM and a TARFlow stream—connected by residual skip connections. Both streams process the same interleaved multimodal sequence $\mathcal{C} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_T)$ under a single left-to-right causal mask, where each element $\boldsymbol{c}_t$ is either a text token or a visual latent.

VLM Stream. The VLM stream is initialized from a pretrained vision-language model (Qwen2.5-VL-7B) and provides high-level semantic representations for language modeling and multimodal understanding. For text positions $t \in \mathcal{M}$, the token is mapped to an embedding via the pretrained text embedding layer. For visual positions $t \in \mathcal{N}$, the intermediate visual latents $\boldsymbol{u}$ (produced by the shallow flow blocks, described in § 3.2) are projected by a lightweight adapter into the VLM representation space. The VLM processes the full interleaved sequence and produces contextual hidden states $\boldsymbol{y}_{\text{vlm}}$.

TARFlow Stream. The TARFlow stream is an autoregressive flow block that operates on the same multimodal sequence under the same causal mask. For each visual latent $\boldsymbol{u}_t$, where $t \in \mathcal{N}$, it applies the autoregressive affine transformation defined in Eq. (2), predicting location and scale parameters

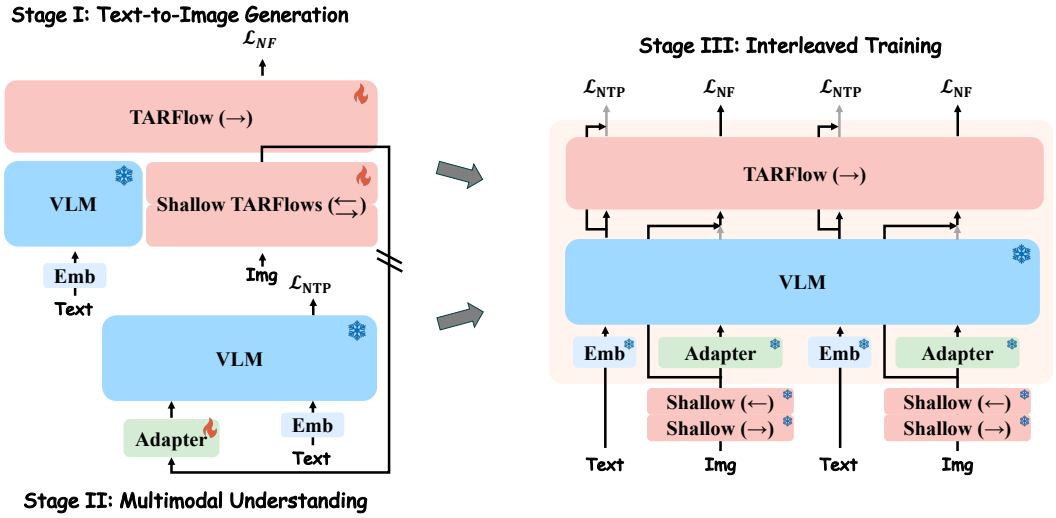


Figure 3: **Multi-Stage Training Pipeline of STARFlow2.** *Stage 1*: Train the TARFlow stream and shallow blocks on text-image pairs for text-to-image generation (VLM frozen). *Stage 2*: Align the visual representation with the VLM by training the adapter on image-to-text tasks (shallow blocks and VLM frozen). *Stage 3*: Activate the vertical skip connections of the Pretzel architecture and jointly optimize on a mixture of understanding, generation, editing, and interleaved tasks.

conditioned on all preceding tokens in the multimodal sequence. For text positions, the TARFlow stream performs standard causal sequence modeling. Because both the VLM and TARFlow streams use the same left-to-right causal structure, they are architecturally compatible—this is what enables true unification.

Vertical Skip Connections. The two streams exchange information through skip connections at every position—the defining feature of the Pretzel architecture (see Stage 3 in fig. 3). Specifically, the TARFlow stream input and output head are defined per-position as:

$$\text{TARFlow input:} \quad \hat{\boldsymbol{c}}_t = \begin{cases} \boldsymbol{u}_t + \boldsymbol{W}_{\text{vlm}} \cdot \boldsymbol{y}_{\text{vlm},t} & \text{if } t \in \mathcal{N} \text{ (visual)} \\ \boldsymbol{y}_{\text{vlm},t} & \text{if } t \in \mathcal{M} \text{ (text)} \end{cases} \tag{3}$$

$$\text{Output sample:} \quad \hat{\boldsymbol{o}}_t = \begin{cases} \mathcal{N}\big(\mu_{\mathcal{D}}(\boldsymbol{y}_{\mathcal{D},t}),\, \sigma^2_{\mathcal{D}}(\boldsymbol{y}_{\mathcal{D},t})\big) & \text{if } t \in \mathcal{N} \text{ (visual)} \\ \text{Cat}\big(\text{softmax}\big(\text{LM}\left(\boldsymbol{y}_{\text{vlm},t} + \boldsymbol{W}_{\mathcal{D}} \cdot \boldsymbol{y}_{\mathcal{D},t}\right)\big)\big) & \text{if } t \in \mathcal{M} \text{ (text)} \end{cases} \tag{4}$$

where $\boldsymbol{W}_{\text{vlm}}$ and $\boldsymbol{W}_{\mathcal{D}}$ are zero-initialized linear projections, $\boldsymbol{y}_{\text{vlm},t}$ and $\boldsymbol{y}_{\mathcal{D},t}$ denote the VLM and TARFlow stream output at position t. The visual skip connection at the TARFlow input preserves the low-level visual information in $\boldsymbol{u}_t$ while injecting high-level semantic information from the VLM into the TARFlow stream. For visual position at the output, the last-layer Deep TARFlow hidden state is projected to predict the affine parameters $(\mu_{\mathcal{D}}, \sigma_{\mathcal{D}})$ to induce the Gaussian distribution of $\mathcal{N}(\mu_{\mathcal{D}},\, \sigma^2_{\mathcal{D}})$ over the intermediate visual latent. For text position, the language modeling head $\text{LM}(\cdot)$ maps the fused text representation to vocabulary logits, which define a categorical distribution $(\text{Cat}(\cdot))$ over the next token. The text skip connection preserves the pretrained language modeling behavior of the VLM while allowing the Deep TARFlow to learn multimodal corrections. Both projections are zero-initialized so that STARFlow2 starts from the pretrained VLM and flow behaviors, gradually learning cross-modal corrections during training.

3.2 Deep-Shallow Flow Design

A single autoregressive pass cannot fully capture the distribution of FAE latents, which exhibit strong local spatial correlations that a purely left-to-right model would need excessive depth to absorb. Following STARFlow [19], STARFlow2 addresses this with a deep-shallow flow design that factorizes the generative process into two stages. A stack of visual-only shallow AF blocks $(f_{\mathcal{S}})$ with alternating scan directions first transforms FAE latents into simpler intermediate representations

$\boldsymbol{u} = f_{\mathcal{S}}(\boldsymbol{x})$ that can be effectively modeled by a single autoregressive pass. The TARFlow stream ($f_{\mathcal{D}}$), within the Pretzel architecture, then models $\boldsymbol{u}$ conditioned on the full multimodal context. This factorization is essential: as shown in Gu et al. [19], the shallow blocks absorb the local complexity of the visual distribution, enabling the deep block to focus on global structure and cross-modal dependencies.

The composed flow yields an exact log-likelihood objective:

$$p(\boldsymbol{x}) = p_0(\boldsymbol{z}) \left|\det J_{f_{\mathcal{D}}}(\boldsymbol{u}; \mathcal{C})\right| \left|\det J_{f_{\mathcal{S}}}(\boldsymbol{x})\right|, \tag{5}$$

where $\boldsymbol{z} = f_{\mathcal{D}}(\boldsymbol{u}; \mathcal{C})$ and p_0 is a standard Gaussian prior. Both the shallow blocks and TARFlow stream contribute to the likelihood computation. Crucially, the shallow blocks operate exclusively on visual latents and do not interfere with the left-to-right causal structure of the Pretzel architecture, preserving cache-friendly interleaved generation.

3.3 Multi-Stage Training Pipeline

We adopt a multi-stage training paradigm that progressively activates components of Pretzel.

Stage 1: Text-to-Image Generation. We first establish a strong visual generation backbone by training on large-scale text-image pairs for text-to-image generation. We optimize the TARFlow stream $f_{\mathcal{D}}$ and the shallow blocks $f_{\mathcal{S}}$, while keeping the pretrained VLM frozen. The VLM encodes text captions into contextual representations that condition the flow, but receives no gradient updates. The training objective minimizes the negative log-likelihood of the composed flow:

$$\begin{aligned}
\mathcal{L}_{\mathrm{NF}} &= \mathbb{E}_{\boldsymbol{x}}\left[\sum_{n=1}^{N}\left(\frac{1}{2}\|\boldsymbol{z}_n\|^2 + \log \sigma_{\mathcal{D}}(\boldsymbol{u}_{<n}; \boldsymbol{c})\right) - \log\left|\det J_{f_{\mathcal{S}}}(\boldsymbol{x})\right|\right] \\
&= \mathbb{E}_{\boldsymbol{x}}\left[\sum_{n=1}^{N} -\log \mathcal{N}(\boldsymbol{u}_n;\, \mu_{\mathcal{D}}(\boldsymbol{u}_{<n}; \boldsymbol{c}),\, \sigma^2_{\mathcal{D}}(\boldsymbol{u}_{<n}; \boldsymbol{c})) - \log\left|\det J_{f_{\mathcal{S}}}(\boldsymbol{x})\right|\right],
\end{aligned} \tag{6}$$

where $\boldsymbol{u} = f_{\mathcal{S}}(\boldsymbol{x})$, $\boldsymbol{z}_n = (\boldsymbol{u}_n - \mu_{\mathcal{D}})/\sigma_{\mathcal{D}}$, and $\boldsymbol{c}$ denotes the preceding multimodal context (e.g., the text caption in Stage 1). The second line reveals that the TARFlow stream performs **Next Gaussian Prediction (NGP)** in $\boldsymbol{u}$-space—the continuous counterpart of next-token prediction: at each visual position, the model predicts the mean and scale of a Gaussian over the next latent $\boldsymbol{u}_n$, conditioned on all preceding tokens, just as an LLM predicts a categorical distribution over the next text token. At inference, sampling from this predicted Gaussian yields:

$$\boldsymbol{u}_n = \mu_{\mathcal{D}}(\boldsymbol{u}_{<n}; \boldsymbol{c}) + \sigma_{\mathcal{D}}(\boldsymbol{u}_{<n}; \boldsymbol{c}) \cdot \boldsymbol{z}_n, \quad \boldsymbol{z}_n \sim \mathcal{N}(0, \boldsymbol{I}). \tag{7}$$

Stage 2: Multimodal Understanding. With the flow components trained, we align the intermediate visual representation $\boldsymbol{u}$ with the pretrained VLM so that it can serve as visual input for multimodal understanding. We train on image-to-text data including captioning and multimodal understanding tasks. We freeze the shallow blocks and VLM, and optimize only the adapter that maps $\boldsymbol{u}$ into the VLM representation space using the next-token prediction loss:

$$\mathcal{L}_{\mathrm{NTP}} = -\frac{1}{|\mathcal{M}|}\sum_{t \in \mathcal{M}} \log p_\theta\left(y_t \mid \mathcal{C}_{<t}\right). \tag{8}$$

Optionally, we can also distill from the frozen VLM (with its original visual encoder) to further improve alignment. This stage ensures the FAE latent space, originally designed for generation, also supports understanding through the VLM.

Stage 3: Interleaved Generation and Understanding. In the final stage, we activate the vertical skip connections of the Pretzel architecture and jointly train on a mixture of data covering multimodal understanding, text-to-image generation, image editing, and interleaved text-image generation. Since both projections $\boldsymbol{W}_{\mathrm{vlm}}$ and $\boldsymbol{W}_{\mathcal{D}}$ are zero-initialized, STARFlow2 starts from the pretrained behaviors of Stages 1–2 and gradually learns cross-modal corrections. The joint objective combines the flow loss and next-token prediction:

$$\mathcal{L} = \mathcal{L}_{\mathrm{NF}} + \lambda\, \mathcal{L}_{\mathrm{NTP}}, \tag{9}$$

where λ balances the two modality losses. This stage unifies all capabilities—understanding, generation, editing, and interleaved synthesis—within the Pretzel framework, with all components jointly optimized end-to-end.

Types	Models	# Params.	MME-p↑	GQA↑	SEED↑	MMB(en)↑	MMMU(val)↑	AI2D↑
Und. Only	LLaVA-v1.5 [36]	7B	1510.7	62.0	58.6	64.3	–	–
	Qwen-VL-Chat [2]	7B	1487.6	57.5	58.2	60.6	–	57.7
	Qwen-2.5-VL-Instruct [3]	7B	1677.9	60.7	75.5	83.8	50.6	82.3
Composite Unified	ILLUME [55]	7B	1445.3	–	72.9	75.1	38.2	71.4
	BLIP3-o [6]	8B	1682.6	–	77.5	75.5	50.6	–
	SEED-X [17]	17B	1457.0	49.1	66.5	70.1	35.6	–
Native Unified	TUNA [38]	1.5B	1461.5	61.4	69.3	–	39.1	71.4
	Janus-Pro [9]	7B	1567.1	62.0	72.1	79.2	41.0	–
	Mogao [33]	7B	1592.0	60.9	74.6	75.0	44.2	–
	Show-o2 [62]	7B	1620.5	63.1	69.8	79.3	48.9	78.6
	TUNA [38]	7B	1641.5	63.9	74.7	–	49.8	79.3
	Emu3 [56]	8B	–	60.3	68.2	58.5	31.6	70.0
	BAGEL [11]	14B	1687.0	–	–	85.0	55.3	–
	STARFlow2 (Ours)	10.6B	1528.8	55.8	71.1	71.5	44.7	67.7

Table 1: **Evaluation on multimodal understanding benchmarks.**

4 Experimental Setup

Datasets We construct a collection of text-image datasets to support the multi-stage training of STARFlow2. In Stage 1, we focus on establishing a strong text-to-image generation backbone using large-scale image-caption data, including an in-house dataset along with CC12M [5], and JourneyDB [51], totaling around 800M text–image pairs. In Stage 2, we train the visual adapter for multimodal understanding using a mixture of CC12M and Cambrian-7M [52], an instruction-style visual question answering data. This stage is trained on approximately 200M examples for image-to-text generation. In Stage 3, we further train STARFlow2 on a broader mixture of datasets covering multimodal understanding, image generation, editing, and interleaved text-image generation datasets, including the in-house dataset in Stage 1, BLIP3-o-60K [6], Cambrian-7M [52], CoMM [8], Pico-Banana [46], OmniEdit [58], and Zebra-CoT [29]. This final stage is trained on approximately 80M examples.

Evaluation We evaluate STARFlow2 on several multimodal understanding benchmarks: MME [15], SEED-Bench [30], MMBench [37], MMMU [64] to assess general multimodal perception and reasoning capability, and GQA [25] for real-world visual reasoning and AI2D [26] for scientific diagram comprehension. For visual generation, we evaluate our model on two widely used benchmarks: GenEval [18] and DPG-Bench [23].

Model and Training Details We employ Qwen2.5-VL-7B-Instruct [3] as the pretrained VLM and FAE [16] trained on DINOv2-g/14 [41] features as the image encoder. The pretrained VLM and the FAE encoder are kept frozen throughout all training stages. We follow the STARFlow [19] design for the causal Deep TARFlow stream and the two visual-only shallow TARFlow blocks. To align flow-based visual latents with the VLM representation space, we introduce a FiLM-style [45] adapter, which first projects visual latents through a lightweight MLP stack and then applies adaptive LayerNorm modulation conditioned on the noise level. In addition, we adopt the multi-noise training strategy from iTARFlow [7] for visual generation. These altogether result in a total of 3.6B trainable parameters. All models are trained at 256×256 resolution with a global batch size of 1024. More details can be found in § C.

5 Results

5.1 Quantitative Results

Multimodal understanding. We evaluate STARFlow2 on multiple multimodal understanding benchmarks, as shown in table 1. STARFlow2 achieves strong performance across standard benchmarks, including MME-P, GQA, SEED, MMBench, MMMU, and AI2D, demonstrating that the Pretzel architecture preserves the pretrained VLM's multimodal perception and reasoning capabilities (D1) while simultaneously supporting flow-based visual generation. Note that STARFlow2 is evaluated at 256×256 resolution due to the current FAE encoder constraint. Despite this limitation, the model maintains effective understanding performance, confirming that integrating a TARFlow stream through vertical skip connections does not compromise the frozen VLM's capabilities.

Type	Method	# Params.	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑
Gen. Only	SD3-Medium [14]	2B	0.99	0.94	0.72	0.89	0.33	0.60	0.74
	FLUX.1 [dev]† [4]	12B	0.98	0.93	0.75	0.93	0.68	0.65	0.82
	Qwen-Image [59]	20B	0.99	0.92	0.89	0.88	0.76	0.77	0.87
Composite Unified	MetaQuery-XL [42]	7B	–	–	–	–	–	–	0.80
	BLIP3-o [6]	8B	–	–	–	–	–	–	0.84
	UniWorld-V1† [34]	12B	0.98	0.93	0.81	0.89	0.74	0.71	0.84
	SEED-X [17]	17B	0.97	0.58	0.26	0.80	0.19	0.14	0.49
Native Unified	Transfusion [67]	7B	–	–	–	–	–	–	0.63
	Janus-Pro [9]	7B	0.99	0.89	0.59	0.90	0.79	0.66	0.80
	Mogao [33]	7B	1.00	0.97	0.83	0.93	0.84	0.80	0.89
	Show-o2 [62]	7B	1.00	0.87	0.58	0.92	0.52	0.62	0.76
	TUNA [38]	7B	1.00	0.97	0.81	0.91	0.88	0.83	0.90
	Emu3 [56]	8B	–	–	–	–	–	–	0.66
	BAGEL [11]	14B	0.99	0.94	0.81	0.88	0.64	0.63	0.82
	BAGEL† [11]	14B	0.98	0.95	0.84	0.95	0.78	0.77	0.88
	STARFlow2 (Ours)	10.6B	0.99	0.89	0.84	0.80	0.86	0.56	0.82

Table 2: **Evaluation of text-to-image generation on GenEval [18].** † refers to the method using LLM rewriters.

Type	Method	# Params.	Global	Entity	Attribute	Relation	Other	Overall↑
Gen. Only	SD3-Medium [14]	2B	87.90	91.01	88.83	80.70	88.68	84.08
	FLUX.1 [dev] [4]	12B	82.10	89.50	88.70	91.10	89.40	84.00
	Qwen-Image [59]	20B	91.32	91.56	92.02	94.31	92.73	88.32
Composite Unified	OmniGen2 [60]	7B	88.81	88.83	90.18	89.37	90.27	83.57
	BLIP3-o [6]	8B	–	–	–	–	–	81.60
	UniWorld-V1 [34]	12B	83.64	88.39	88.44	89.27	87.22	81.38
Native Unified	Janus-Pro [9]	7B	86.90	88.90	89.40	89.32	89.48	84.19
	Mogao [33]	7B	82.37	90.03	88.26	93.18	85.40	84.33
	Show-o2 [62]	7B	89.00	91.78	89.96	91.81	91.64	86.14
	TUNA [38]	7B	90.42	91.68	90.94	91.87	90.73	86.76
	Emu3 [56]	8B	–	–	–	–	–	81.60
	BAGEL [11]	14B	88.94	90.37	91.29	90.82	88.67	85.07
	STARFlow2 (Ours)	10.6B	91.45	91.83	88.91	91.09	88.61	84.94

Table 3: **Evaluation of text-to-image generation on DPG-Bench [23].**

Image Generation. We further evaluate text-to-image generation on GenEval and DPG-Bench, as reported in tables 2 and 3. GenEval measures fine-grained instruction following across object presence, counting, colors, attributes, and spatial relationships, while DPG-Bench focuses on compositional text-to-image alignment at the global, entity, attribute, and relation levels. STARFlow2 achieves 0.82 on GenEval and 84.94 on DPG-Bench, demonstrating that autoregressive normalizing flows generate visually meaningful images (D2) while sharing the same causal decoding structure as text generation (D3).

Effect of Joint Training on Interleaved Data We compare the text-to-image performance of STARFlow2 after Stage 1 and Stage 3 on GenEval and DPG-Bench. Stage 1 trains the TARFlow stream and shallow TARFlows for text-to-image generation, while Stage 3 activates the vertical skip connections and jointly optimizes the model on multimodal understanding, image generation, editing, and interleaved generation data. As shown in table 4, Stage 3 improves image generation performance on both benchmarks, with relative gains of 60.8% on GenEval and 3.6% on DPG-Bench. This indicates that joint multimodal training along with the vertical fusion in the Pretzel does not degrade the pretrained visual generation pathway.

Training Stage	GenEval ↑	DPG-Bench ↑
Stage 1: Text-to-Image Generation	0.51	82.02
Stage 3: Interleaved Training	**0.82**	**84.94**
Δ Improvement	+0.31	+2.92

Table 4: **Effect of interleaved training on text-to-image generation.** We compare STARFlow2 after Stage 1 and Stage 3 using the overall scores on GenEval [18] and DPG-Bench [23].



Figure 4: Text-to-Image generation examples from STARFlow2 at 256 $\times$ 256 resolution.

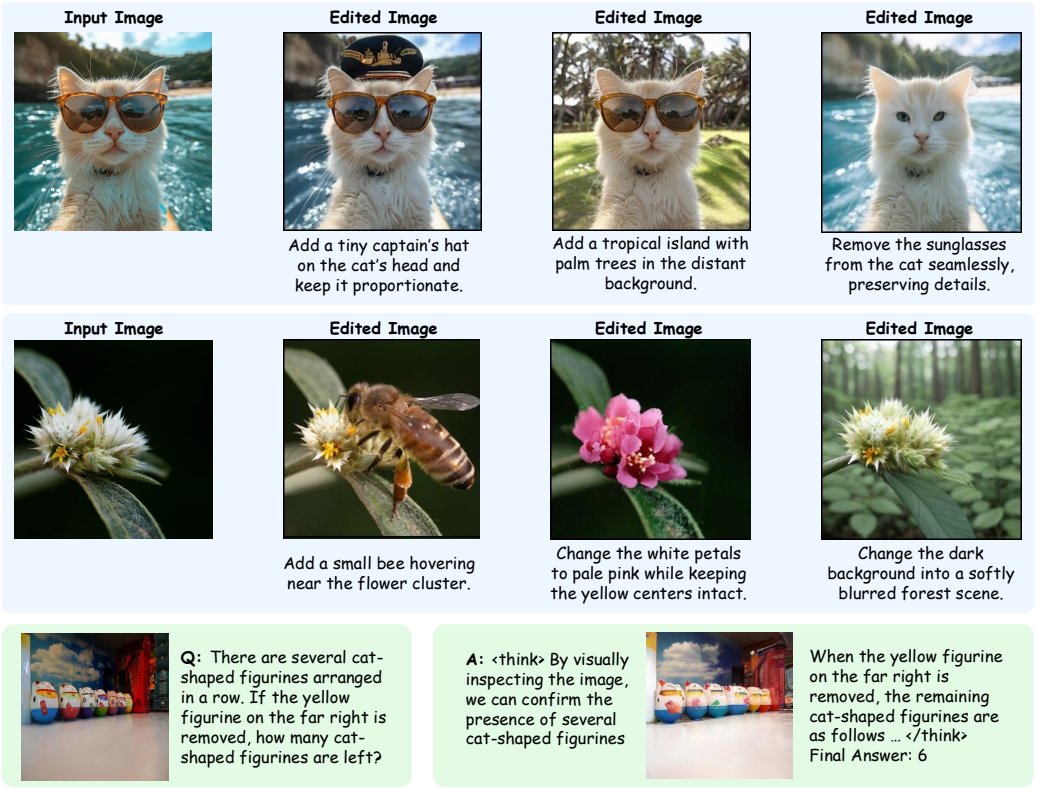


Figure 5: Image editing and interleaved text-image generation examples from STARFlow2.

5.2 Qualitative Results

fig. 4 shows representative text-to-image generation examples from STARFlow2. fig. 5 demonstrates examples in image editing and interleaved text-image generation. The qualitative results show that STARFlow2 can follow editing instructions, modify local attributes, and adjust visual content while preserving the overall scene structure, verifying it as a unified multimodal generator with cache-friendly interleaved generation.

5.3 Pretzel vs. Bagel (MoT)

Mixture-of-Transformers (MoT) [32] has been widely adopted in unified multimodal models [50; 11; 33], including BAGEL [11]. When applying MoT to combine a pretrained VLM with TARFlow, we find two failure modes: (1) Freezing the VLM and training only the TARFlow-specific branch leads to inferior generation (fig. 6), potentially because horizontal MoT-style fusion is ill-suited to single-pass causal autoregressive flows: unlike diffusion models, TARFlow cannot iteratively incorporate VLM conditioning across layers and instead relies mainly on same-layer attention; (2) Jointly finetuning the VLM degrades understanding (MME drops to $\sim$800), suggesting that naively adapting VLM parameters for flow-based generation risks erasing pretrained understanding ability before learning effective unified generation. These observations motivate Pretzel: by vertically interleaving a frozen VLM with a trainable TARFlow stream through zero-initialized skip connections, Pretzel preserves pretrained understanding while enabling the flow to access rich VLM representations at every position. Empirically, Pretzel improves generation while maintaining substantially stronger understanding than MoT.



Figure 6: Images generated by adopting the MoT-style [32] fusion.

6 Related Work

Generative Modeling Paradigms. Text generation is dominated by autoregressive LLMs [1], while visual generation is led by diffusion and flow-matching methods [22; 49; 44; 35; 14] whose iterative sampling is structurally distinct from single-pass autoregressive decoding. Discrete tokenization [54; 63; 40] bridges this gap but introduces quantization loss. Normalizing flows [12; 48; 27; 21] offer exact likelihood and single-pass sampling; recent TARFlow-style models [65; 19; 20] parameterize flows with causal Transformers, matching diffusion quality while sharing the same left-to-right structure as LLMs. STARFlow2 extends autoregressive flows from vision-only generation to unified multimodal modeling for the first time.

Unified Multimodal Models. A prominent approach combines autoregressive language modeling with diffusion for images [67; 61; 62; 50; 38; 39; 11; 33], but inherits a structural asymmetry: text tokens enter the KV-cache causally while images require iterative denoising and re-encoding for interleaved generation. MoT [32], adopted in BAGEL [11], routes modalities to separate feed-forward parameters—a horizontal separation that maintains two sub-networks within one shell. Discrete unified approaches [56; 31; 9; 6] avoid the hybrid design but sacrifice continuous fidelity. STARFlow2 achieves true unification via the Pretzel architecture, which vertically interleaves TARFlow and VLM streams under the same causal mask with skip connections, avoiding both re-encoding overhead and routing complexity.

Visual Representations. Many unified models decouple understanding and generation representations [9; 61; 53], while recent work explores shared representations [38; 47]. STARFlow2 operates in the FAE latent space [16], which provides compact continuous latents serving both understanding and flow-based generation within a single representation.

7 Conclusion

We presented STARFlow2, a unified multimodal model that bridges language models and normalizing flows under the same causal Transformer mechanism via the Pretzel architecture. By vertically interleaving a frozen pretrained VLM with a TARFlow stream via residual skip connections, STARFlow2 satisfies three desiderata: preserving pretrained multimodal understanding (D1), generating high-fidelity continuous images without quantization (D2), and unifying both modalities under a single causal mechanism without diffusion's iterative denoising (D3). Together with a deep–shallow TARFlow design and a unified FAE latent space, STARFlow2 supports multimodal understanding, text-to-image generation, image editing, and interleaved text–image generation with cache-friendly inference. Experiments demonstrate strong image generation while retaining the pretrained VLM's capabilities. These results establish autoregressive normalizing flows as a principled foundation for unified multimodal modeling. Scaling to higher resolutions, end-to-end training across all components, and improving fine-grained visual fidelity remain important directions for future work (see § A).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *CoRR*, abs/2308.12966, 2023.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv e-prints*, pages arXiv–2506, 2025.

[5] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021.

[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] Tianrong Chen, Jiatao Gu, David Berthelot, Joshua Susskind, and Shuangfei Zhai. Normalizing flows with iterative denoising. *arXiv preprint arXiv:2604.20041*, 2026.

[8] Wei Chen, Lin Li, Yongqi Yang, Bin Wen, Fan Yang, Tingting Gao, Yu Wu, and Long Chen. Comm: A coherent interleaved image-text dataset for multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8073–8082, 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] Ethan Chern, Zhulin Hu, Steffi Chern, Siqi Kou, Jiadi Su, Yan Ma, Zhijie Deng, and Pengfei Liu. Thinking with generated images. *arXiv preprint arXiv:2505.22525*, 2025.

[11] 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.

[12] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[13] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[14] 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.

[15] 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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[16] Yuan Gao, Chen Chen, Tianrong Chen, and Jiatao Gu. One layer is enough: Adapting pretrained visual encoders for image generation. *arXiv preprint arXiv:2512.07829*, 2025.

[17] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024.

[18] 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.

[19] Jiatao Gu, Tianrong Chen, David Berthelot, Huangjie Zheng, Yuyang Wang, Ruixiang Zhang, Laurent Dinh, Miguel Angel Bautista, Josh Susskind, and Shuangfei Zhai. Starflow: Scaling latent normalizing flows for high-resolution image synthesis. *arXiv preprint arXiv:2506.06276*, 2025.

[20] Jiatao Gu, Ying Shen, Tianrong Chen, Laurent Dinh, Yuyang Wang, Miguel Angel Bautista, David Berthelot, Josh Susskind, and Shuangfei Zhai. Starflow-v: End-to-end video generative modeling with normalizing flow. *arXiv preprint arXiv:2511.20462*, 2025.

[21] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International conference on machine learning*, pages 2722–2730. PMLR, 2019.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[23] 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.

[24] 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*, 37:139348–139379, 2024.

[25] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[26] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016.

[27] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[28] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

[29] Ang Li, Charles Wang, Deqing Fu, Kaiyu Yue, Zikui Cai, Wang Bill Zhu, Ollie Liu, Peng Guo, Willie Neiswanger, Furong Huang, et al. Zebra-cot: A dataset for interleaved vision language reasoning. *arXiv preprint arXiv:2507.16746*, 2025.

[30] 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.

[31] 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.

[32] Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

[33] Chao Liao, Liyang Liu, Xun Wang, Zhengxiong Luo, Xinyu Zhang, Wenliang Zhao, Jie Wu, Liang Li, Zhi Tian, and Weilin Huang. Mogao: An omni foundation model for interleaved multi-modal generation. *arXiv preprint arXiv:2505.05472*, 2025.

[34] 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.

[35] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[36] 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.

[37] 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.

[38] 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.

[39] Zhiheng Liu, Weiming Ren, Xiaoke Huang, Shoufa Chen, Tianhong Li, Mengzhao Chen, Yatai Ji, Sen He, Jonas Schult, Belinda Zeng, et al. Tuna-2: Pixel embeddings beat vision encoders for multimodal understanding and generation. *arXiv preprint arXiv:2604.24763*, 2026.

[40] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024.

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

[42] 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.

[43] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[44] 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.

[45] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[46] Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing. *arXiv preprint arXiv:2510.19808*, 2025.

[47] 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.

[48] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

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

[50] Weijia Shi, Xiaochuang Han, Chunting Zhou, Weixin Liang, Xi Victoria Lin, Luke Zettlemoyer, and Lili Yu. Lmfusion: Adapting pretrained language models for multimodal generation. *arXiv preprint arXiv:2412.15188*, 2024.

[51] Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, et al. Journeydb: A benchmark for generative image understanding. *Advances in neural information processing systems*, 36:49659–49678, 2023.

[52] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

[53] 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. *arXiv preprint arXiv:2412.14164*, 2024.

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

[55] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21612–21622, 2025.

[56] 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.

[57] Yuqing Wang, Zhijie Lin, Yao Teng, Yuanzhi Zhu, Shuhuai Ren, Jiashi Feng, and Xihui Liu. Bridging continuous and discrete tokens for autoregressive visual generation. *arXiv preprint arXiv:2503.16430*, 2025.

[58] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. In *The Thirteenth International Conference on Learning Representations*, 2024.

[59] 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.

[60] 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, 2026. URL https://arxiv.org/abs/2506.18871.

[61] 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. *arXiv preprint arXiv:2408.12528*, 2024.

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

[63] Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *ICLR*, 2024.

[64] 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.

[65] Shuangfei Zhai, Ruixiang ZHANG, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Ángel Bautista, Navdeep Jaitly, and Joshua M Susskind. Normalizing flows are capable generative models. In *Forty-second International Conference on Machine Learning*, 2025.

[66] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952. IEEE, 2023.

[67] 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.

[68] Zijun Zhou, Yingying Deng, Xiangyu He, Weiming Dong, and Fan Tang. Multi-turn consistent image editing. *arXiv preprint arXiv:2505.04320*, 2025.

A Limitations and Future Work

While STARFlow2 demonstrates the potential of TARFlow-style normalizing flows for unified multimodal modeling, several limitations remain. First, STARFlow2 relies on a multi-stage training pipeline to stably integrate the pretrained VLM, FAE visual representation, adapter, and TARFlow components. Although effective, this staged procedure introduces additional complexity and may lead to under-optimization. A natural direction for future work is to optimize all components end-to-end, allowing the visual representation and cross-modal fusion modules to be jointly shaped by both next-token prediction and TARFlow-based likelihood objectives.

Second, the current model is constrained by the pretrained FAE encoder. In particular, the image resolution and visual quality are limited by the FAE latent space, which can affect fine-grained visual fidelity and text rendering. Replacing the pretrained FAE encoder with a more native visual representation, such as pixel-level or patch-level embeddings, is a promising direction. This would reduce dependence on an external visual tokenizer or autoencoder and move STARFlow2 toward a more fully native unified multimodal model.

Finally, although STARFlow2 supports multimodal understanding, image generation, editing, and interleaved text–image generation in a single causal framework, it is not yet state-of-the-art on all benchmarks. Improving data scale, training stability, visual representation learning, and long-context interleaved generation remains important future work. Nevertheless, our results suggest that autoregressive normalizing flows offer a promising foundation for unified multimodal modeling, providing a new direction that combines continuous visual generation, exact likelihood training, and cache-friendly causal decoding within a single architecture.

B Impact Statement

The proposed method explores autoregressive normalizing flows as a foundation for unified multimodal understanding and generation. By enabling text and visual latents to be generated under the same causal framework, this work may contribute to more efficient and flexible multimodal systems, particularly for interleaved text-image generation, image editing, and interactive visual reasoning. More broadly, unified multimodal models could improve accessibility and communication by helping users express ideas across modalities, generate visual explanations, and interact with information in more natural ways. They may also support applications in domains such as media production, data visualization, simulation, and assistive technologies.

C Implementation Details

C.1 Architecture Design

Component	Specification
Pretrained VLM	Qwen2.5-VL-7B-Instruct [3] (frozen)
FAE	FAE [16] on DINOv2-g/14 [41] (frozen)
Deep TARFlow $f_{\mathcal{D}}$	24 Transformer Layers, width 3072
Shallow TARFlows $f_{\mathcal{S}}$	2 Blocks, 4 Transformer Layers each Block, width 3072
Visual adapter	1 MLP, 1 FiLM Layer
Trainable parameters	3.6B

Table 5: Model specification of STARFlow2. The pretrained VLM and FAE encoder are kept frozen.

C.2 Training Details

STARFlow2 is trained on 64 H100 GPUs. In all the experiments, we share the following training configuration for our proposed STARFlow2.

```
training config:
    batch_size=1024
```

```
optimizer='AdamW'
adam_beta1=0.9
adam_beta2=0.95
adam_eps=1e-8
min_learning_rate=1e-6
learning_rate_schedule=cosine
weight_decay=1e-4
mixed_precision_training=bf16
```

We use a learning rate of $1e-4$ for Stage 1 and Stage 2 training and a learning rate of $5e-5$ for Stage 3

D Analysis of Vertical Skip Connections

The Pretzel architecture employs the vertical skip connections to allow information exchange between the VLM and TARFlow stream. Since these connections are activated only in Stage 3 with zero-initialized projections, as described in § 3.3, we therefore examine whether the later-activated connections become effectively used after training.

We first focus on the visual vertical skip connection at the TARFlow input in Eq. (3), which injects the VLM representation $\boldsymbol{y}_{\mathrm{vlm},t}$ into TARFlow at each visual position $t \in \mathcal{N}$. Specifically, we measure the contribution ratio r_t^{vis} and directional alignment s_t^{vis} between the intermediate TARFlow visual latent and the projected VLM feature as follows:

$$r_t^{\mathrm{vis}} = \frac{\|\boldsymbol{W}_{\mathrm{vlm}}\boldsymbol{y}_{\mathrm{vlm},t}\|_2}{\|\boldsymbol{u}_t\|_2 + \|\boldsymbol{W}_{\mathrm{vlm}}\boldsymbol{y}_{\mathrm{vlm},t}\|_2}, \quad s_t^{\mathrm{vis}} = \cos(\boldsymbol{u}_t, \boldsymbol{W}_{\mathrm{vlm}}\boldsymbol{y}_{\mathrm{vlm},t}), \quad t \in \mathcal{N}. \tag{10}$$

We perform text-to-images generation from 50 randomly sampled text prompts and visualize the distributions of these two quantities in fig. 7 (a). We observe that the contribution ratio has a mean of 0.472, indicating that the projected VLM stream accounts for a substantial fraction of the fused representation magnitude. Meanwhile, the near-zero cosine similarity suggests that the VLM stream contributes complementary information through the vertical skip connection.

Similarly, for the textual vertical skip connection at the output in Eq. (4), we analyze how much the TARFlow stream contributes to the final language-modeling representation. For each text position $t \in \mathcal{M}$, we define the contribution ratio r_t^{txt} and the cosine similarity s_t^{txt}:

$$r_t^{\mathrm{txt}} = \frac{\|\boldsymbol{W}_{\mathcal{D}}\boldsymbol{y}_{\mathcal{D},t}\|_2}{\|\boldsymbol{y}_{\mathrm{vlm},t}\|_2 + \|\boldsymbol{W}_{\mathcal{D}}\boldsymbol{y}_{\mathcal{D},t}\|_2}, \quad s_t^{\mathrm{txt}} = \cos(\boldsymbol{y}_{\mathrm{vlm},t}, \boldsymbol{W}_{\mathcal{D}}\boldsymbol{y}_{\mathcal{D},t}), \quad t \in \mathcal{M}. \tag{11}$$

As shown in fig. 7 (b), the textual skip connection exhibits a much smaller contribution ratio, suggesting that the projected TARFlow output states $\boldsymbol{W}_{\mathcal{D}}\boldsymbol{y}_{\mathcal{D},t}$ only lightly corrects the pretrained VLM representation. This is consistent with the design goal of preserving the pretrained language modeling capability while allowing TARFlow to provide modest multimodal corrections.

E Qualitative Examples

figs. 8 and 9 shows qualitative examples for text-to-image generation and image editing examples.

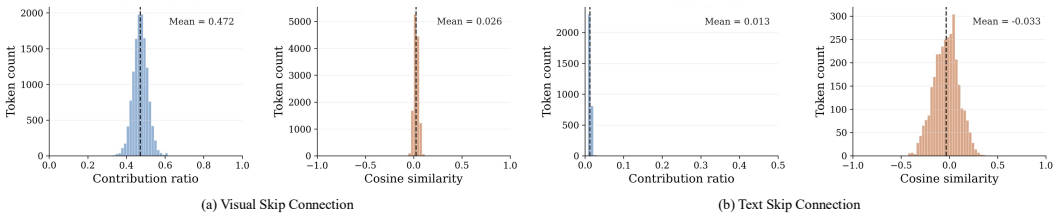


Figure 7: **Analysis of the vertical skip connection.**

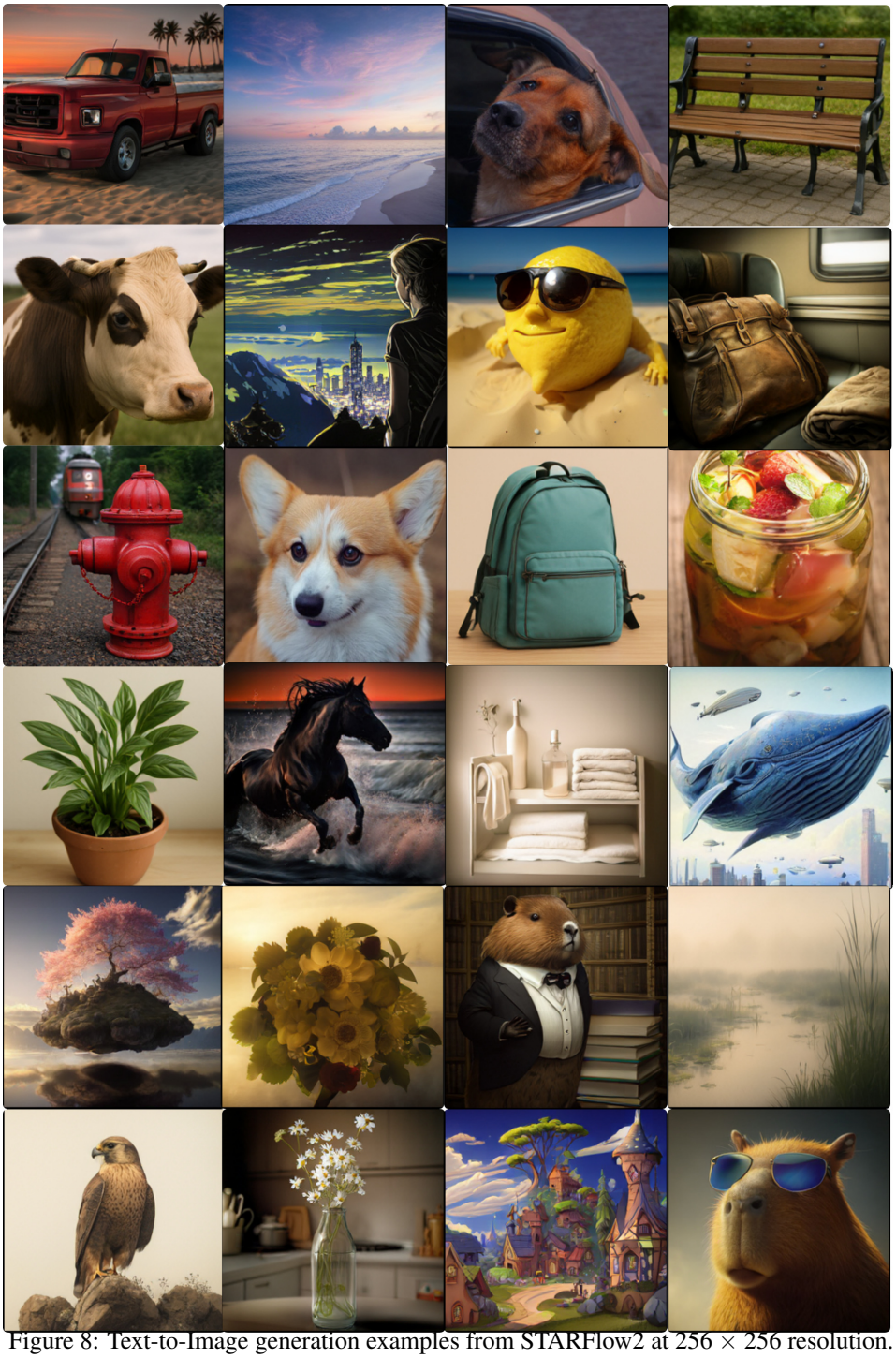


Figure 8: Text-to-Image generation examples from STARFlow2 at 256 × 256 resolution.

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 made the main claim of "proposing the first unified multimodal framework where both text and image generation employ the same autoregressive Transformer

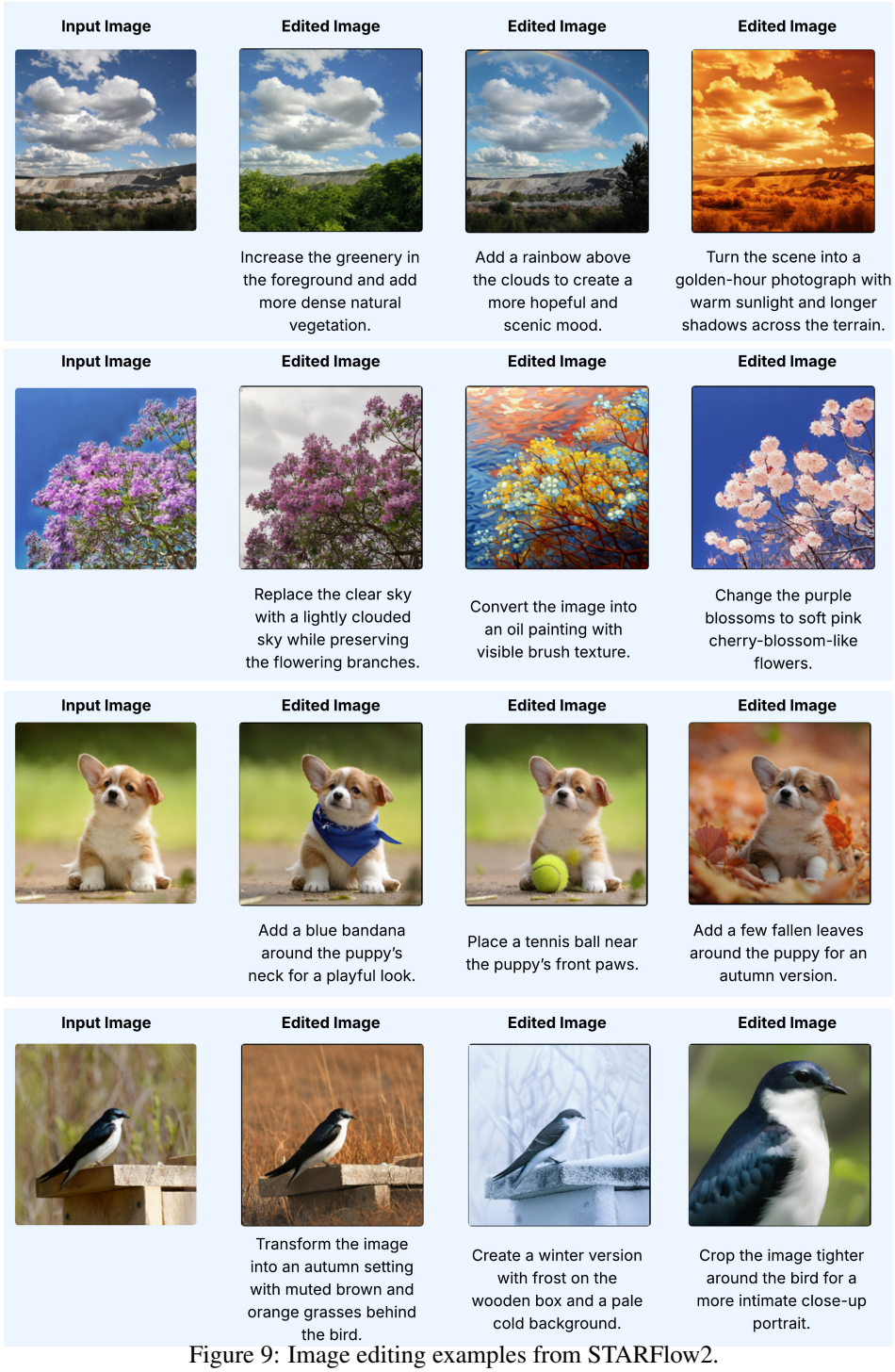


Figure 9: Image editing examples from STARFlow2.

mechanism under the same causal mask, achieving true architectural unification without quantization or iterative denoising" in both 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: See § A

   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: N/A

   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 include the model hyperparameters in main § C for people to reproduce. We list the training data we used in the main paper.

   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 will be released after acceptance and internal review.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We specified the details about training and testing

   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: Due to the costly nature of each run, we did not include error bars

   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 mention the compute requirements in the 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: This paper conducted in 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: [Yes]

Justification: Due to page limits, we include societal impacts in the 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: [No]

Justification: We did not plan to release pretrained models therefore no risk for misuse.

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 cited all the papers for codes and datasets we used.

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: [No]

Justification: There are no newly released 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: There are no crowdsourcing steps.

    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: There is no crowdsourcing nor research with human subjects.

    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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    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.