SCALING AUTOREGRESSIVE TEXT-TO-IMAGE GENERATIVE MODELS WITH CONTINUOUS TOKENS

Anonymous authors
Paper under double-blind review

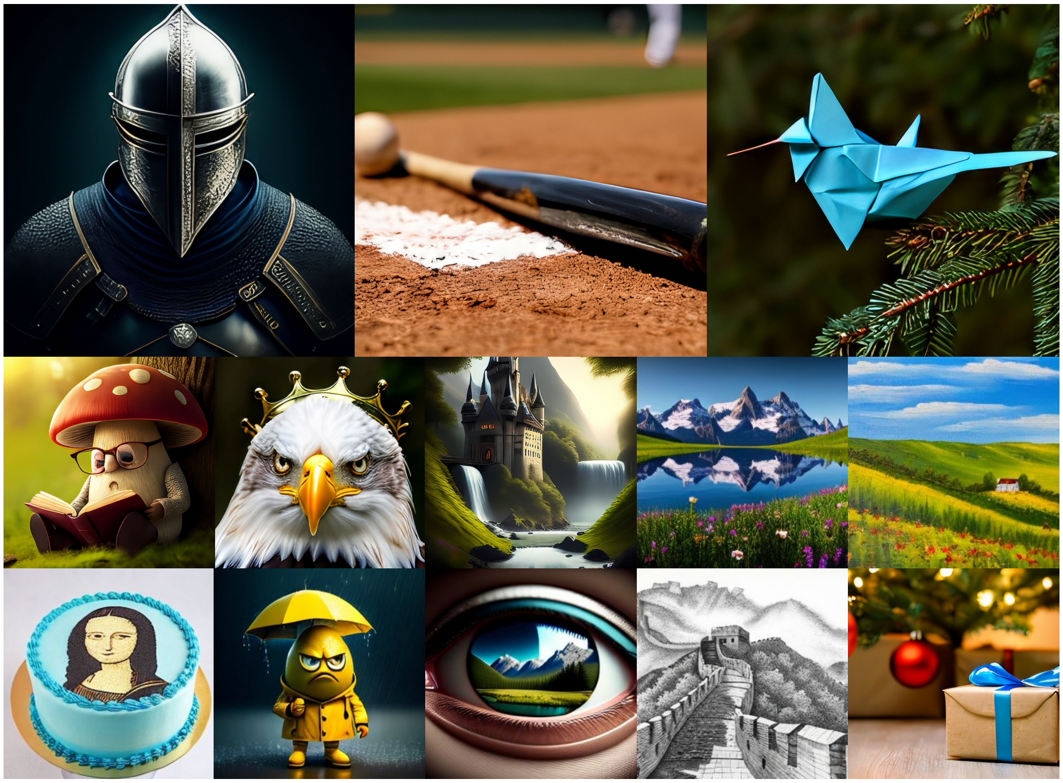


Figure 1: Samples from our Fluid 10.5B autoregressive model with continuous tokens. Please see the appendix for the text prompts to generate these images.

ABSTRACT

Scaling up autoregressive models in vision has not proven as beneficial as in large language models. In this work, we investigate this scaling problem in the context of text-to-image generation, focusing on two critical factors: whether models use discrete or continuous tokens, and whether tokens are generated in a random or fixed raster order using BERT- or GPT-like transformer architectures. Our empirical results show that, while all models scale effectively in terms of validation loss, their evaluation performance—measured by FID, GenEval score, and visual quality—follows different trends. Models based on continuous tokens achieves significantly better visual quality than those using discrete tokens. Furthermore, the generation order and attention mechanisms significantly affect the GenEval score: random-order models achieve notably better GenEval scores compared to raster-order models. Inspired by these findings, we train Fluid, a random-order autoregressive model on continuous tokens. Fluid 10.5B model achieves a new state-of-the-art zero-shot FID of 6.16 on MS-COCO 30K, and 0.69 overall score on the GenEval benchmark. We hope our findings and results will encourage future efforts to further bridge the scaling gap between vision and language models.

1 INTRODUCTION

Scaling laws underpin the unprecedented success of large language models (LLMs). Empirically, increasing the number of parameters in autoregressive models consistently leads to significant performance improvements and the emergence of new capabilities in natural language processing (NLP) tasks (Devlin et al., 2018; Radford et al., 2018; Brown et al., 2020; Kaplan et al., 2020; Wei et al., 2022). This empirical relationship has inspired numerous efforts to scale up language models, resulting in the development of many highly capable models (Bubeck et al., 2023; Team et al., 2023; Achiam et al., 2023).

Encouraged by this success, many attempts have been made to adopt and scale up autoregressive models in computer vision, particularly for generative tasks like text-to-image generation (Yu et al., 2021; 2023; Bai et al., 2024; El-Nouby et al., 2024). However, the performance and visual quality of content generated by these models often fall short compared to other generative models, such as diffusion models (Ho et al., 2020; Saharia et al., 2022; Rombach et al., 2022a; Esser et al., 2024), leaving it unclear whether similar scaling laws apply to the vision domain.

We propose several hypotheses for the performance gap. First, the vector quantization (VQ) (van den Oord et al., 2017) step, which is required for most visual autoregressive models, may introduce significant information loss, ultimately limiting model performance. Second, unlike the inherently sequential nature of language, generating visual content might benefit more from a different autoregressive prediction order. Third, there is often a confusion between two levels of generalizability when evaluating scaling laws in vision models: (a) generalization to new data using the same metric as the training loss (commonly referred to as validation loss), and (b) generalization to a new metric or problem different from the training objective, such as FID (Heusel et al., 2017), the GenEval benchmark (Ghosh et al., 2024), or visual quality. We hypothesize that power-law scaling (Kaplan et al., 2020) applies to (a) for autoregressive models on vision data, but not necessarily to (b).

To investigate these hypotheses, we conduct a comprehensive empirical study on the scaling behavior of autoregressive models in the context of text-to-image generation. Specifically, we explore two key factors: whether the model operates on continuous or discrete tokens, and whether tokens are generated in a random or fixed raster order. To this end, we utilize the Diffusion Loss (Li et al., 2024) to make autoregressive models compatible with continuous tokens. We generalize BERT-like vision model MaskGIT (Chang et al., 2022) as random-order autoregression, as it conceptually predicts output tokens in a randomized order while retaining the autoregressive nature of "predicting next tokens based on known ones". We analyze the behavior of four autoregressive variants, each employing different combinations of these two factors. We scale their parameters from 150M to 3B and evaluate their performance using three metrics: validation loss, FID (Heusel et al., 2017), and GenEval score (Ghosh et al., 2024). We also inspect the visual quality of the generated images.

Our experiments indicate that VQ-based models, regardless of whether they use a random or fixed raster order, exhibit a *slower* improvement in FID scores when scaling model size compared to models operating on continuous tokens. VQ models also produce images of lower visual quality, likely due to information loss introduced by vector quantization.

Furthermore, the token generation order and the associated attention mechanism primarily influence the global structure of the generated image. In the GenEval benchmark, random-order models with bidirectional attention significantly outperform raster-order models with causal attention, particularly when generating multiple objects. Random-order models can readjust the global structure at every prediction step, whereas raster-order models cannot. This suggests that the token generation order plays a crucial role in achieving better text-to-image alignment.

Our experiments also demonstrate that validation loss scales as a power-law with model size, no matter whether the model operates with continuous or discrete tokens. This implies scalable behavior at the level of (a) generalization—generalizing to new data using the same metric as the training loss—which aligns with observations in language models (Kaplan et al., 2020). However, as for generalizing to a different metric, such as FID or GenEval score, although performance consistently improves with better validation loss, the trend may not follow a strict power-law.

Building on these findings, we scale the Fluid model, *i.e.*, random-order model with continuous tokens, up to 10.5B parameters and train it using the WebLI dataset (Chen et al., 2022). The resulting Fluid 10.5B model achieves a zero-shot FID of 6.16 on MS-COCO and a GenEval overall score of

0.69, comparing favorably with leading text-to-image generative models such as DALL-E 3 (Betker et al., 2023) and Stable Diffusion 3 (Esser et al., 2024). We hope that our empirical findings and positive results could shed light on the scaling behavior for text-to-image generation models and further innovation in this frontier.

2 RELATED WORK

Text-to-image diffusion models. The dominant approaches are based on diffusion models. Dall-E 2 (OpenAI, 2023), Imagen (Ho et al., 2022), and Stable Diffusion (Rombach et al., 2022a) revolutionized text-to-image generation. Most recently, MidJourney (Midjourney, 2023), SD v3 (Esser et al., 2024), and Imagen3 (Baldridge et al., 2024) can generate realistic images that are hard to tell from real images. However, generating samples is usually computationally expensive due to the multiple forward passes.

Autoregressive (AR) models. While being the defacto model for language modeling, AR models are lagging behind diffusion models for text-to-image generation. AR models (Yu et al., 2022; Chang et al., 2023; Yu et al., 2023) are often used together with discrete tokenizers (van den Oord et al., 2017; Esser et al., 2021), which often limit the modeling capability. For example, Parti (Yu et al., 2022) scales up the model to 20B, which obtains a slightly lower/better FID score of 7.23 on MS-COCO than 7.27 by the 3.4B diffusion-based Imagen. Here we find that, with a continuous tokenizer, our small 369M model can already achieve the same FID score as Parti.

Recently Li et al. (2024) challenges the conventional wisdom and replaces the discrete tokenizer with a continuous tokenizer via a diffusion loss. They introduced masked autoregressive (MAR) model obtains strong results for class-conditioning generation on ImageNet. However, scaling MAR models for text-to-image generation is unexplored. Here we empirically study the scaling behavior for MAR and report several findings important for the research community.

Scaling language models. Kaplan et al. (2020) empirically observed that, for language model, the validation loss scales as a power-law with model size, dataset size, and the amount of compute used for training. Hoffmann et al. (2022) discovered that contemporary LLMs are under-trained, and that for compute-optimal training, the model size and the number of training tokens should be scaled equally. Under the same compute budget while using 4x more more data, their 70B Chinchilla outperforms much larger models, such as the 530B Megatron-Turing NLG (Smith et al., 2022). Wei et al. (2022) found that larger models have emergent abilities that are not present in smaller models. These observations have inspired significant efforts to scale up language models to trillions of parameters (Achiam et al., 2023; Team et al., 2023; Dubey et al., 2024).

Scaling vision models. Similar scaling law has been obscure for computer vision. For recognition models (Tan, 2019; Zhai et al., 2022; He et al., 2022; Dehghani et al., 2023), scaling often comes with diminishing returns. For example, Dehghani et al. (2023) scale ViT up from 3B to 22B but only observed 0.25% accuracy increase in ImageNet linear probing. The scaling of generative models is more promising - DiT (Peebles & Xie, 2023) shows consistent improvement in generation quality when scaling up compute and model size (despite only up to 600M). The follow-up Sora (OpenAI, 2024) further shows the potential to scale up for video generation. In this paper, we perform a comprehensive empirical study of AR models for text-to-image generation and our findings shed light on closing the gap in scaling behavior between vision and language generative models.

3 PRELIMINARY: AUTOREGRESSIVE IMAGE GENERATION

Given a sequence of tokens $\{x^1, x^2, ..., x^n\}$ where the superscript $1 \leq i \leq n$ specifies an order, autoregressive models (Gregor et al., 2014; van den Oord et al., 2016b;a; Parmar et al., 2018; Chen et al., 2018; 2020) formulate the generation problem as "next token prediction":

$$p(x^1, ..., x^n) = \prod_{i=1}^{n} p(x^i \mid x^1, ..., x^{i-1}). \tag{1}$$

Following the chain rule, the network is trained to model $p(x^i \mid x^1, ..., x^{i-1})$ and generate tokens iteratively. While all autoregressive models share this fundamental approach, differences in their

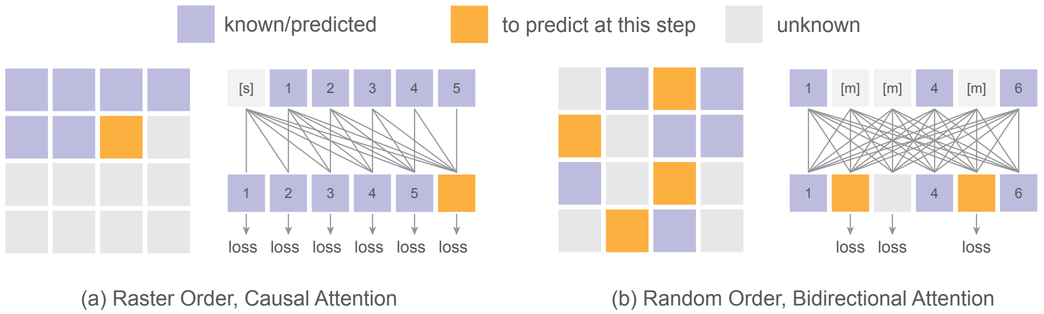

known/predicted
to predict at this step
unknown
[s]
1
2
3
4
5
loss
[m]
6
(a) Raster Order, Causal Attention
(b) Random Order, Bidirectional Attention

Figure 2: **Autoregressive models with different orders.** (a) A raster-order autoregressive model predicts one next token based on the known ones, implemented using a GPT-like transformer with causal attention. (b) A random-order autoregressive model predicts one or multiple tokens simultaneously given a random order, implemented using a BERT-like transformer with bidirectional attention.

design can affect the performance. Two key design choices are the representation of x, *i.e.*, discrete or continuous, and the generation order, which we elaborate below.

Discrete *vs.* continuous tokens. The goal of an autoregressive model is to estimate $p(x^i \mid x^1, ..., x^{i-1})$. Traditionally, this is done by transforming the image into a set of discrete tokens with a finite vocabulary and then estimating a categorical distribution over the vocabulary. The training objective is to minimize the cross-entropy loss, and sampling can be efficiently performed using categorical sampling. Most autoregressive image generation models rely on this form of token discretization (Esser et al., 2021; Chang et al., 2022; Tian et al., 2024; Yu et al., 2022).

However, such discretization often leads to a significant loss of information from the image (Figure 3). Recent work (Li et al., 2024) has shown the possibility of applying a small diffusion model to approximate the distribution of each image token in a continuous fashion. This approach eliminates the need for vector quantization, and allows modeling images with continuous tokenizers with much better visual reconstruction quality. In this paper, we explore the scaling behavior of autoregressive image models on both discrete and continuous tokens.

Raster Order + GPT vs. Random Order + BERT. In autoregressive image generation, there are two primary generation orders: raster and random. As illustrated in Figure 2, raster order generates tokens sequentially from left to right, top to bottom. This fixed-order generation is well-suited for a GPT-like transformer architecture, which predicts the next token in a causal manner. In contrast, random order allows multiple tokens to be generated in each step. The selection of these tokens can either be completely random or based on a sampling mechanism that prioritizes tokens with higher predicted confidence scores (Chang et al. (2023); Li et al. (2024)).

Each generation order has its pros and cons. Raster order models with GPT-like transformer support fast inference via key-value (kv) caching. However, this causal structure can also introduce performance degradation. On the other hand, random order generation is usually achieved with a BERT-like bidirectional attention mechanism. While this approach prevents the usage of kv-caching, it enables the model to decode multiple tokens at each autoregressive step, allowing global editing. Despite their individual strengths, it remains unclear in the literature which generation order scales better for text-to-image generation tasks. In this work, we compare the performance and scaling behaviors of raster-order and random-order autoregressive models.

4 IMPLEMENTATION

The overall framework of our text-to-image model training is straightforward. An image tokenizer first converts the original image into tokens. These tokens are then partially masked, and a transformer is trained to reconstruct the masked tokens conditioned on the text. Below, we provide a detailed description of each component of our framework.

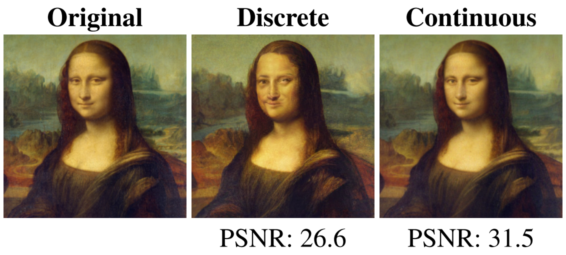


Figure 3: **Reconstruction quality of the tokenizers we used.** The discrete tokenizer is significantly worse than the continuous tokenizer.

Image Tokenizer. We use a pre-trained image tokenizer to encode 256×256 images into a token space. Such a tokenizer can be either discrete or continuous, facilitating different training objectives of the autoregressive model. In our experiments, the discrete tokenizer is an VQ-based model pre-trained on the WebLI dataset (Chen et al., 2022). We follow Muse (Chang et al., 2023) to encode each image into 16×16 discrete tokens with a vocabulary size of 8192. For the continuous tokenizer, we adopt a widely-used one from Stable Diffusion (Rombach et al., 2022b), which encodes the image into 32×32 continuous tokens, each containing 4 channels. To be consistent in sequence length with the discrete tokenizer, we further group each 2×2 patch of continuous tokens into a single token, resulting in a final sequence length of 256, with each token containing 16 channels. As shown in Figure 3, the continuous tokenizer can achieve notably better reconstruction quality than the discrete one.

Text Encoder. The raw text (maximum length of 128) is tokenized by SentencePiece tokenization (SentencePiece, 2023), and embedded with a pre-trained T5-XXL encoder (Raffel et al., 2020), which has 4.7B parameters and is frozen during training. To further align the text embeddings for image generation, we add a small bidirectional text aligner consisting of six trainable transformer blocks on top of the T5 embeddings to extract the final text representation.

Transformer. After encoding the original image into a token sequence using an image tokenizer, we use a standard decoder-only transformer model (Vaswani et al., 2017) for autoregressive generation. Each block consists of three consecutive layers – self-attention, cross-attention, and MLP. The self-attention and MLP layers are only applied to visual tokens, while the cross attention layer takes visual and textual tokens as queries and keys, respectively. As shown in Figure 2, for raster-order models, the transformer predicts the next token based on previous tokens using causal attention for the self-attention block, similar to GPT. In random-order models, unknown tokens are masked by a learnable token, and the transformer predicts these masked tokens using bidirectional attention, similar to BERT.

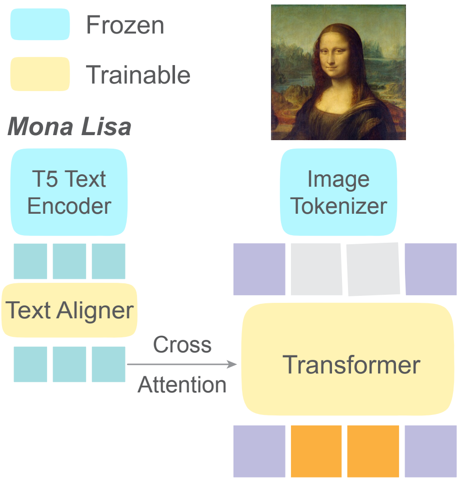


Figure 4: **Our text-to-image generation framework.** We use a pre-trained image tokenizer to convert the image into either discrete or continuous tokens. The corresponding text is embedded using a pre-trained T5 encoder, followed by a trainable text aligner. The transformer then takes cross-attention from the text embeddings to predict the missing tokens.

Output head. For *discrete* tokens, we follow the common practice with autoregressive models. The outputs are transformed into categorical distributions by softmax following a linear layer, whose weights are reused from the input embedding layer. For *continuous* tokens, we apply a six layer light-weight MLP as the diffusion head (Li et al., 2024) to model the per-token distribution. The embedding dimension of this head is the same as the backbone transformer. The per-token diffusion process follows (Nichol & Dhariwal, 2021; Li et al., 2024). The noise schedule has a cosine shape, with 1000 steps at training time; at inference time, it is resampled to 100 steps.

5 EXPERIMENTS

Dataset. We use a subset of the WebLI (Web Language Image) dataset (Chen et al., 2022) as our training set, which consists of image-text pairs from the web with high scores for both image quality and alt-text relevance. By default, the images are center-cropped and resized to 256×256, and the text is the corresponding alt-text of each image.

Training. Unless otherwise specified, we use the AdamW optimizer ($\beta_1 = 0.9, \beta_2 = 0.95$) (Loshchilov & Hutter, 2019) with a weight decay of 0.02 to train each model for 1M steps with a batch size of 2048. This is equivalent to approximately 3 epochs on our dataset. We employ a constant learning rate schedule with a 65K-step linear warmup and a maximum learning rate of 1×10^{-4}. For training the random-order models, we randomly sample the masking ratio from [0, 1] following a cosine schedule, similar to MaskGIT (Chang et al., 2022), to mask each image. For all models, exponential moving average of the weights are gathered by a decay rate of 0.9999 and then used for evaluation.

Inference. We follow the practices established by VQGAN (Esser et al., 2021), Muse (Chang et al., 2023), and MAR (Li et al., 2024) to generate images from text prompts. For random-order models, we use 64 steps for image generation with a cosine schedule (Chang et al., 2022). To further enhance generation performance, we apply temperature and classifier-free guidance, as is commonly practiced.

Evaluation. We evaluate the scaling behavior of different autoregressive model variants both quantitatively and qualitatively. Quantitatively, we evaluate the validation loss on 30K images from the MS-COCO 2014 training set, as well as two widely-adopted metrics: zero-shot Frechet Inception Distance (FID) on MS-COCO, and the GenEval score (Ghosh et al., 2024). Inference hyper-parameters, such as temperature and classifier-free guidance, are optimized for each evaluation metric. FID is computed over 30K randomly selected image-text pairs from the MS-COCO 2014 training set, providing a metric that evaluates both the fidelity and diversity of generated images. The GenEval benchmark, on the other hand, measures the model's ability to generate images that accurately reflect the given prompt. For qualitative evaluation, we generate images from several prompts using each model and compare the visual quality of the generated images.

5.1 SCALING BEHAVIORS

In this section, we explore how two key design choices in autoregressive image generative models—token representation and generation order—affect performance and scaling behavior. We construct models with different combinations of these two design choices, resulting in four distinct variants of autoregressive image generation models. We also explore the generalizability of these models across different data and evaluation metrics. Our experiments reveal several intriguing properties.

Validation losses consistently scale with model size. In Figure 5, we examine the scaling behavior of the four autoregressive variants in terms of validation loss. We observe consistent reductions in validation loss as model size increases from 150 million to 3 billion parameters, which aligns with the findings in Henighan et al. (2020). This demonstrates that the improvements in training loss resulting from increased model size generalize well to validation loss on data different from the training data.

Continuous tokens and large models are crucial for visual quality. In Figure 7, we compare the visual quality of images generated by the four autoregressive variants. The visual quality of models using discrete tokens is significantly worse than that of models using continuous tokens, *e.g.*, the eyes of the corgi is asymmetric for discrete token based models and scaling up can not solve this problem. This limitation is largely because of the discrete tokenizer, which introduces substantial information loss. For instance, even with 3B parameters, the discrete token models cannot generate an accurate Mona Lisa due to poor reconstruction quality of the tokenizer (Figure 3). In contrast, models with continuous tokens produce much higher-quality images.

Additionally, larger models show consistent improvements in both visual quality and image-text alignment. For example, a random-order model with 0.2B parameters struggles to generate "an

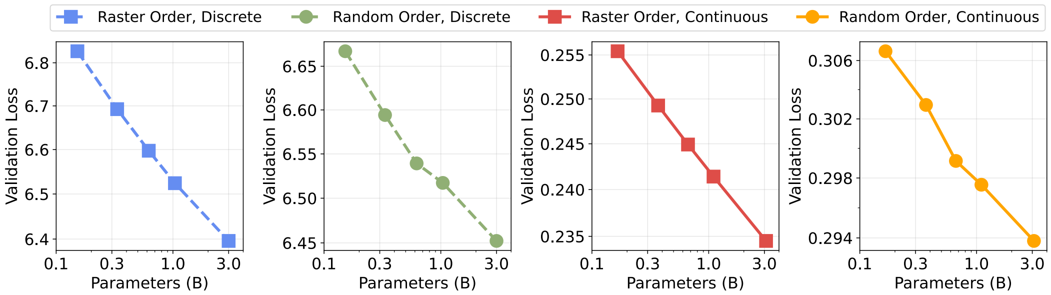


Figure 5: **Validation loss scales as a power-law with model size.** The validation loss is evaluated on 30K images randomly sampled from the MS-COCO 2014 training set. The x and y axes are in log-scale. The change in y is relatively small for each plot, making the log-scale alike linear-scale.

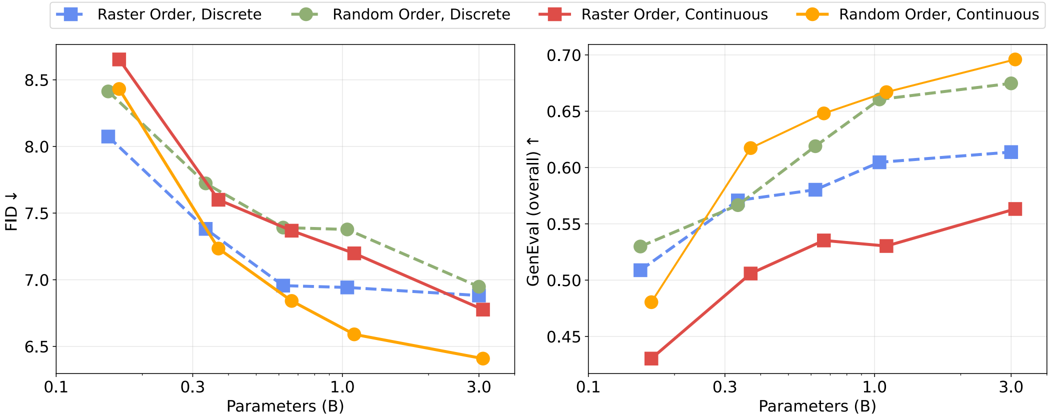


Figure 6: **Random-order models using continuous tokens (orange) achieve the best performance on evaluation metrics.** FID (lower is better) is evaluated on 30K images randomly sampled from the MS-COCO 2014 training set, while the GenEval overall score (higher is better) is assessed using the 553 prompts provided by the official benchmark, with four images generated for each prompt. Among all models, random-order models on continuous tokens consistently show an improvement in evaluation metrics as model size increases and achieve the best FID and GenEval scores.

angry duck doing heavy weightlifting at the gym", while the same model with 3B parameters can generate the corresponding images successfully. This demonstrates that modeling continuous tokens and increasing model size are crucial for achieving high visual quality in autoregressive image generation models.

Random-order models with continuous tokens scale best in evaluation scores. In Figure 6, we analyze the scaling behavior of the four autoregressive variants in terms of FID and GenEval overall scores. We find that the improvements observed in validation loss do not always translate to better evaluation metrics. For example, raster-order models with discrete tokens (blue line) reach a plateau in both FID and GenEval scores around 1B parameters. Among the four variants, random-order models with continuous tokens (*i.e.*, Fluid) show consistent improvements in evaluation metrics up to 3B parameters, achieving the best overall performance. Therefore, we focus on further investigating the scaling behavior of this model in the remaining section.

Random-order models with continuous tokens scale with training computes. In Figure 8, we plot validation loss, FID, and GenEval scores as functions of total training steps and compute for different model sizes of Fluid. We observe consistent improvements in both validation loss and evaluation performance with increased training steps and compute. However, the benefits from additional training steps saturate around 1M steps, indicating that training smaller models for more

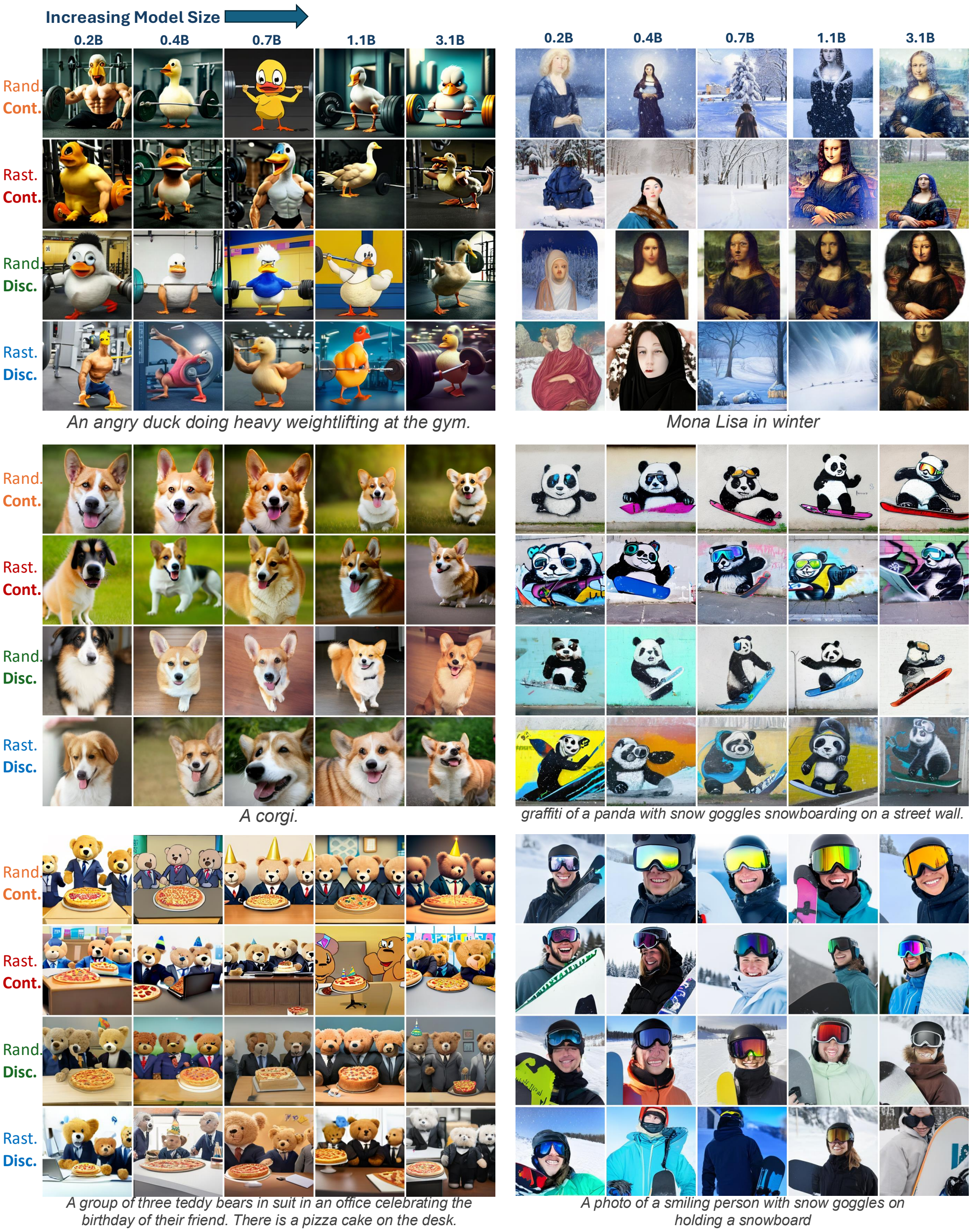

Increasing Model Size
0.2B
0.4B
0.7B
1.1B
3.1B
Rand. Cont.
Rast. Cont.
Rand. Disc.
Rast. Disc.
An angry duck doing heavy weightlifting at the gym.
Mona Lisa in winter
A corgi.
graffiti of a panda with snow goggles snowboarding on a street wall.
A group of three teddy bears in suit in an office celebrating the birthday of their friend. There is a pizza cake on the desk.
A photo of a smiling person with snow goggles on holding a snowboard

Figure 7: **Visual quality and image-text alignment improves with increasing model size.** Random-order (Rand.) models with continuous tokens (Cont.) achieve the highest visual quality and best image-text alignment (top row). Best viewed with zoom-in for details.

steps is less compute-efficient compared to training larger models for fewer steps. This behavior aligns with observations in language models, highlighting the potential for scaling up model sizes with sufficient training.

Strong correlation between validation loss and evaluation metrics. In Figure 9, we plot FID and GenEval scores against validation loss for different model sizes of Fluid and observe a strong correlation. To quantify this, we fit the data points using linear regression. The Pearson correlation

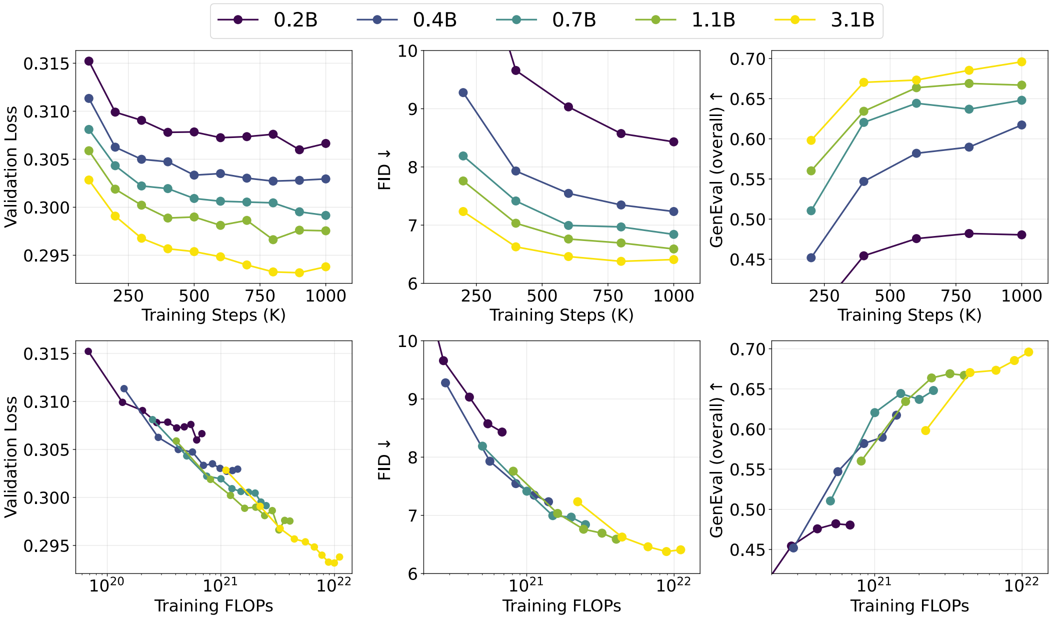


Figure 8: **Validation losses and evaluation performance scale with increasing training steps and computes.** We use random-order models with continuous tokens. Results for other autoregressive variants are included in the appendix. The training compute is computed as model GFLOPs×batch size×training steps×3, where the factor of 3 accounts for the backward pass being approximately twice as compute-intensive as the forward pass.

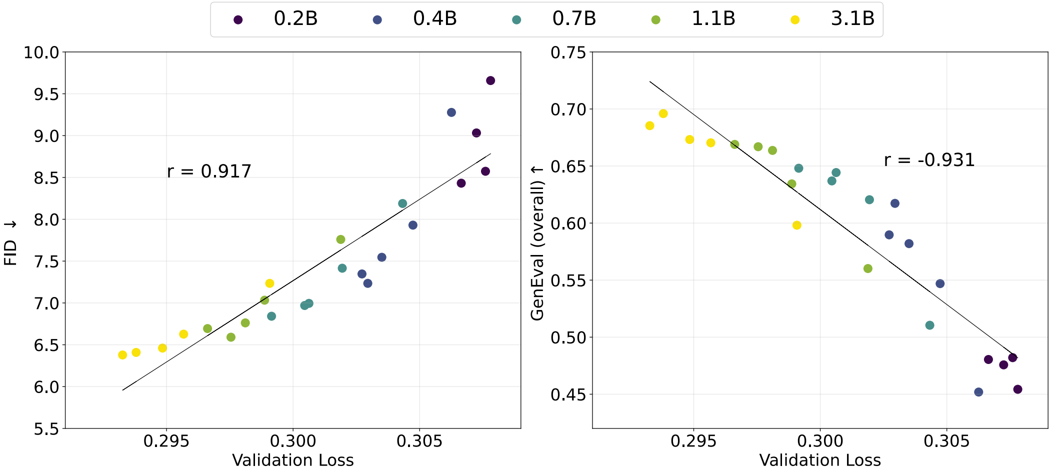


Figure 9: **Validation loss and evaluation metrics are highly correlated.** We use random-order models with continuous tokens. The Pearson correlation coefficients for FID and GenEval scores are 0.917 and -0.931, respectively. We also observe that the linear correlation slightly weakens and becomes less pronounced for the 3.1B model.

coefficients for FID and GenEval scores are 0.917 and -0.931, respectively, indicating a nearly linear relationship between validation loss and these evaluation metrics across model sizes ranging from 150M to 3B[1]. Encouraged by this positive trend, we trained a model with 10.5B parameters and a batch size of 4096 for 1M steps, achieving state-of-the-art text-to-image generation performance, as discussed in the next section.

---

[1] Since both FID and GenEval scores have lower/upper bounds, this linear relationship cannot hold indefinitely. Future work should explore the limits of this correlation.

Table 1: **System-level comparison.** Fluid achieves leading results on both MS-COCO zero-shot FID-30K and GenEval benchmark (Ghosh et al., 2024). †: CM3Leon result is reported without retrieval.

		MS-COCO	GenEval						
	#params	FID-30K↓	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall
diffusion model									
LDM	1.4B	12.64	0.92	0.29	0.23	0.70	0.02	0.05	0.37
DALL-E 2	4.2B	10.39	0.94	0.66	0.49	0.77	0.10	0.19	0.52
DALL-E 3	-	-	0.96	0.87	0.47	0.83	0.43	0.45	0.67
Imagen	3B	7.27	-	-	-	-	-	-	-
SD3	8B	-	0.98	0.84	0.66	0.74	0.40	0.43	0.68
Transfusion	7.3B	6.78	-	-	-	-	-	-	0.63
RAPHAEL	3B	6.61	-	-	-	-	-	-	-
autoregressive model									
CM3Leon†	7B	10.82	-	-	-	-	-	-	-
Show-o	1.3B	9.24	0.95	0.52	0.49	0.82	0.11	0.28	0.53
Muse	3B	7.88	-	-	-	-	-	-	-
Parti	20B	7.23	-	-	-	-	-	-	-
Fluid (our work)	369M	7.23	0.96	0.64	0.53	0.78	0.33	0.46	0.62
	665M	6.84	0.96	0.73	0.51	0.77	0.42	0.51	0.65
	1.1B	6.59	0.96	0.77	0.61	0.78	0.34	0.53	0.67
	3.1B	6.41	0.98	0.83	0.60	0.82	0.41	0.53	**0.70**
	10.5B	**6.16**	0.96	0.83	0.63	0.80	0.39	0.51	0.69

5.2 BENCHMARKING WITH PREVIOUS SYSTEMS

In this section, we compare our Fluid, *i.e.*, continuous random-order autoregressive model, with leading text-to-image generation systems in Table 1 (Rombach et al., 2022b; Ramesh et al., 2022; Betker et al., 2023; Saharia et al., 2022; Esser et al., 2024; Zhou et al., 2024; Xue et al., 2024; Yu et al., 2023; Xie et al., 2024; Chang et al., 2023; Yu et al., 2022). The Fluid smallest model, with 369M parameters, achieves a zero-shot FID of 7.27 on MS-COCO and a GenEval overall score of 0.62, matching the performance of many state-of-the-art models with several billion parameters. The Fluid largest model, with 10.5B parameters, further improves the zero-shot FID on MS-COCO to 6.16 and increases the GenEval overall score to 0.69[2], with a speed of 1.571 seconds per image per TPU (evaluated on 32 TPU v5 with batch size 2048). Detailed model configurations and generation speed results are included in the appendix. We hope these strong results and promising scaling behavior provide valuable insights and support for the scalability of autoregressive models in visual generative modeling.

6 DISCUSSION

In this paper, we present an empirical study on the scaling behavior of autoregressive models for text-to-image generation. We investigate two critical design factors: random order versus raster order, and discrete tokens versus continuous tokens. Our results show that random-order models with continuous tokens achieve the best performance and scaling behavior across various evaluation metrics and in terms of visual quality. Building on these findings, we scale up the random-order model with continuous tokens, namely Fluid, to 10.5B parameters, and achieves state-of-the-art text-to-image generation performance. We hope that our findings and promising results could provide valuable insights into the scaling behavior of autoregressive models for image generation and help bridge the gap between the scaling performance of vision models and language models.

Reproducibility Statement. To aid reproducibility, we have provided the implementation details of our framework in Section 4, training hyper-parameters in Section 5, and model configurations in the Appendix. For the diffusion loss used for continuous tokens, we have strictly followed the open-sourced code of Li et al. (2024).

---

[2] We observe that the GenEval scores plateau for the 10.5B Fluid compared to the 3.1B Fluid; however, it continues to show consistent improvements in visual quality and FID.

REFERENCES

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.

Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. In *CVPR*, 2024.

Jason Baldridge, Jakob Bauer, Mukul Bhutani, Nicole Brichtova, Andrew Bunner, Kelvin Chan, Yichang Chen, Sander Dieleman, Yuqing Du, Zach Eaton-Rosen, et al. Imagen 3. *arXiv preprint arXiv:2408.07009*, 2024.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *OpenAI*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. MaskGIT: Masked generative image Transformer. In *CVPR*, 2022.

Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative Transformers. In *ICML*, 2023.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. PixelSNAIL: An improved autoregressive generative model. In *ICML*, 2018.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *ICML*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. *arXiv preprint arXiv:2401.08541*, 2024.

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

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *NeurIPS*, 2024.

Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. Deep autoregressive networks. In *ICML*, 2014.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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

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

Midjourney. https://www.midjourney.com/, 2023.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021.

OpenAI. Dall-e-2, https://openai.com/product/dall-e-2, 2023.

OpenAI. Sora: Creating video from text. `https://openai.com/sora`, 2024.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image Transformer. In *ICML*, 2018.

William Peebles and Saining Xie. Scalable diffusion models with Transformers. In *ICCV*, 2023.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

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.

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

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

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022.

SentencePiece. Unsupervised text tokenizer for neural network-based text generation. `https://github.com/google/sentencepiece`, 2023.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022.

Mingxing Tan. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024.

Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *NeurIPS*, 2016a.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016b.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

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.

Zeyue Xue, Guanglu Song, Qiushan Guo, Boxiao Liu, Zhuofan Zong, Yu Liu, and Ping Luo. Raphael: Text-to-image generation via large mixture of diffusion paths. *NeurIPS*, 2024.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022.

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

A More Scaling Results

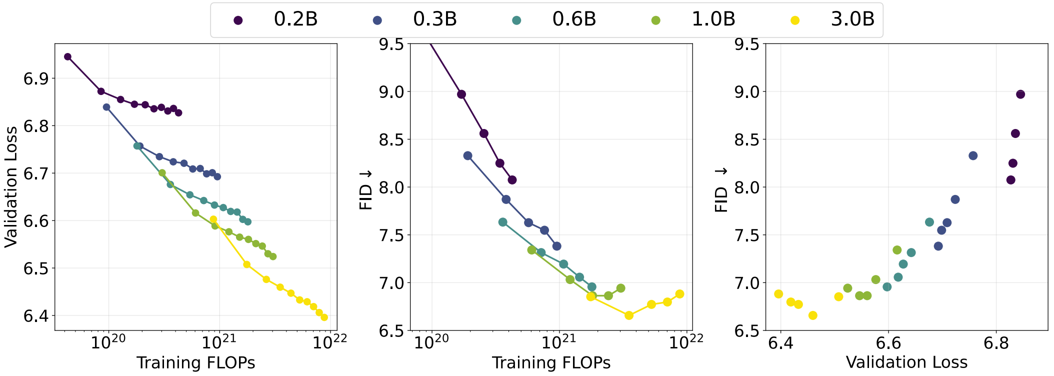


Figure A1: Validation loss and FID w.r.t. training FLOPs for raster-order models with discrete tokens.

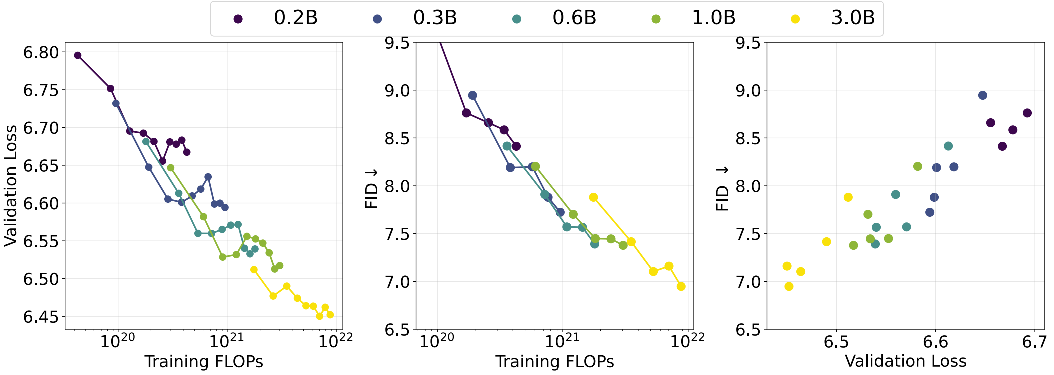


Figure A2: Validation loss and FID w.r.t. training FLOPs for random-order models with discrete tokens.

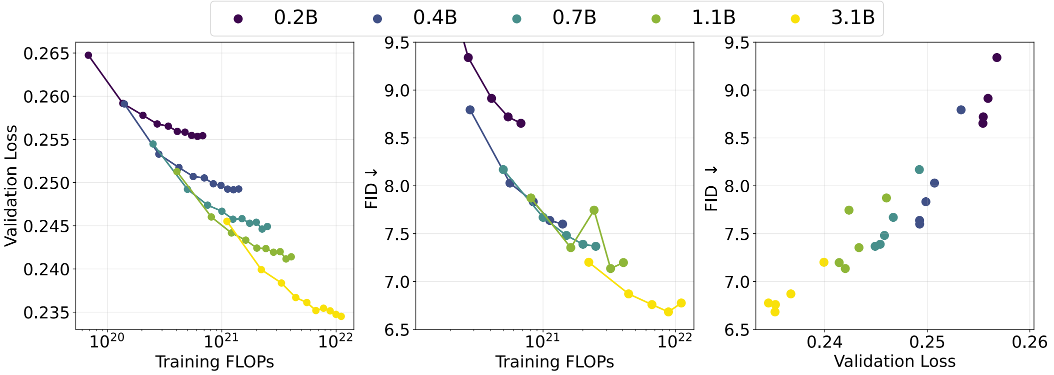


Figure A3: Validation loss and FID w.r.t. training FLOPs for raster-order models with continuous tokens.

A.1 Scaling behavior w.r.t training FLOPs.

In Figure A1 – A3, we present the relationship between validation loss and FID with respect to training FLOPs for the other three autoregressive variants. As shown, all three variants exhibit consistent scaling behavior in validation loss, but the FID gains start to level off for the 3B raster-order model that uses discrete tokens. This hints that simply using a GPT-like language model for images in a straightforward way may not scale well. To improve scaling for visual data, further adaptations such as continuous tokens and random-order generation are needed.

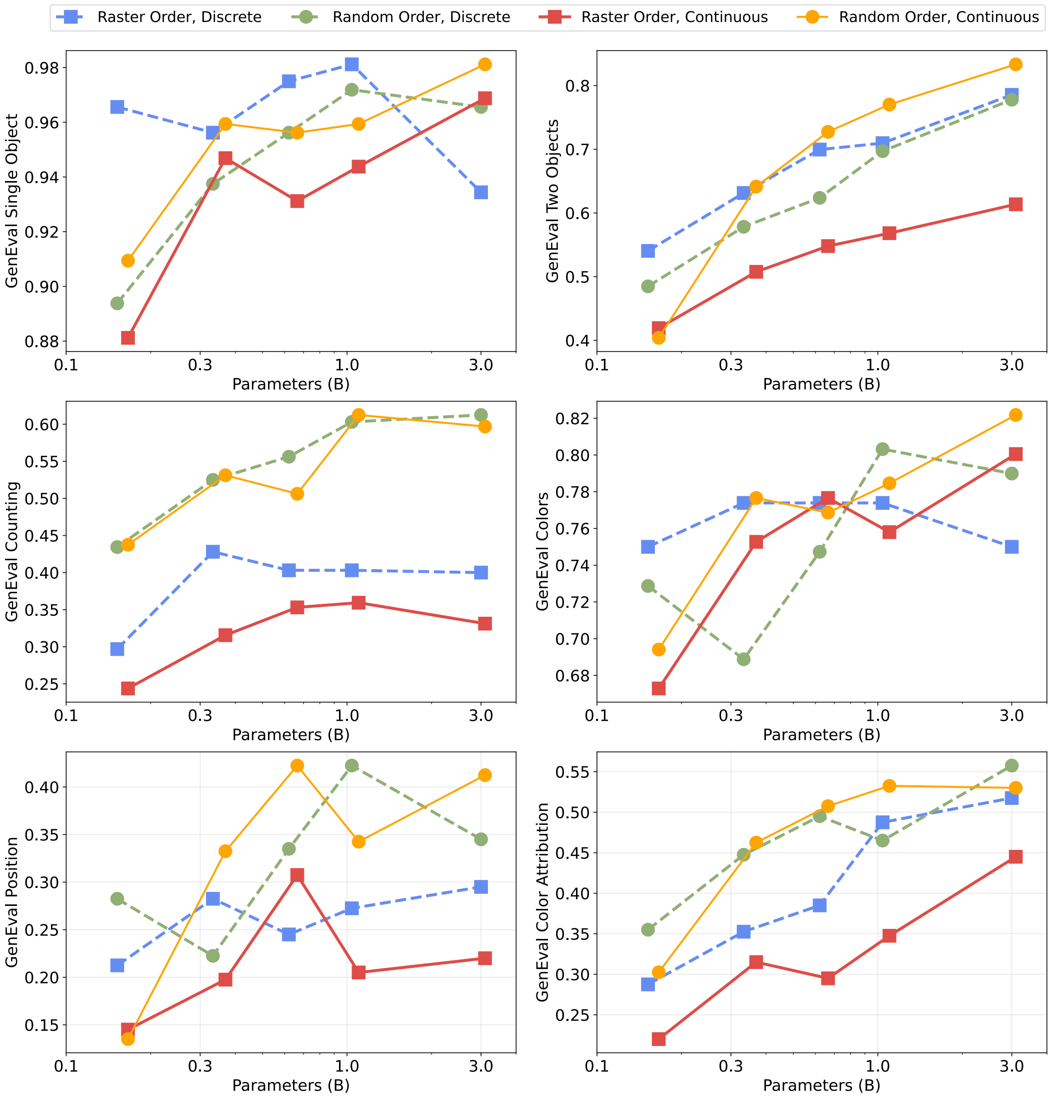


Figure A4: Scaling behavior on fine-grained GenEval scores across different model setups.

A.2 FINE-GRAINED GENEVAL SCORES.

In Figure A4, we present the performance of the four autoregressive variants across all metrics in the GenEval benchmark. As shown, all models perform well in single-object scenarios. The performance on other metrics also consistently improves as model size increases. Additionally, we observe that random-order models significantly outperform raster-order models in metrics related to counting and position, both of which require a better global generation structure—an area where random-order models have an advantage.

A.3 SCALING COMPARISON WITH SD3

The scaling of diffusion-based models, and later flow-based models, has been studied in several works in the literature. Most notably, in the SD3 paper (Esser et al., 2024) (one of the current state-of-the-art flow-based models), they scaled the model from 0.8B to 8B parameters and observed consistent scaling behavior. In Figure A5, we compare the scaling behavior of SD3 and Fluid in terms of the GenEval score. Surprisingly, despite various differences in model architecture, training data, and implementations, the scaling rates of these two models are quite similar within the same GenEval score range (0.6–0.7). Moreover, Fluid performs even better than SD3 at the same number of parameters. However, we also observe that the GenEval scores plateau for the 10.5B Fluid model

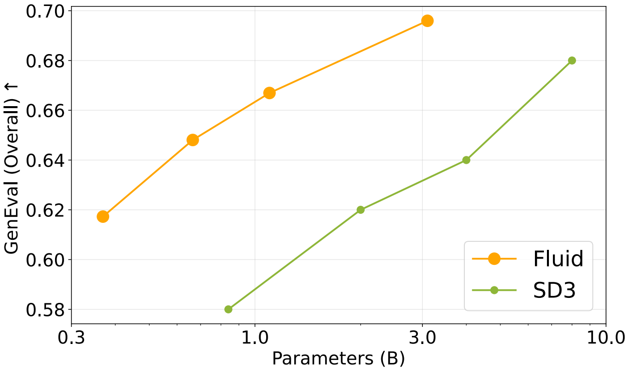


Figure A5: Scaling comparison between our Fluid model and SD3 across different backbone sizes on GenEval benchmark.

compared to the 3.1B Fluid model. We hypothesize that this is due to the inconsistency between the scaling of the validation loss and the validation metrics. The scaling behavior of SD3 beyond 10B parameters would also be interesting to explore, and we leave it for future work.

B MORE IMPLEMENTATION DETAILS

Model Configurations. In Table A1, we provide the detailed configurations of our models across different sizes. The MLP ratio is fixed at 4 for all models. The text aligner consistently consists of 6 transformer blocks, with the same channel size as the image transformer. The DiffLoss MLP also contains 6 MLP layers, with channels matching those of the image transformer. The generation speed is evaluated on 32 TPU v5 with a batch size of 2048, and we report the time needed to generate one image per TPU.

Table A1: Model configurations of our random-order models on continuous tokens.

#Params	#Blocks	#Channels	#Heads	Speed (sec/img)
166M	12	768	12	0.047
369M	16	1024	16	0.078
665M	20	1280	16	0.110
1.1B	24	1536	16	0.180
3.1B	32	2304	24	0.483
10.5B	34	4096	64	1.571

CFG and Temperature. To enable CFG (Ho & Salimans, 2022), we randomly replace the text condition with a dummy vacant text string for 10% of the samples. During inference, the model is run with the given text condition and the vacant text, yielding two outputs for each token.

For discrete tokens, the two outputs are conditional logit ℓ_c and unconditional logit ℓ_u. Then the final logit ℓ_g with a guidance scale of ω is $\ell_g = (1+\omega)\cdot\ell_c - \omega\cdot\ell_u$. Afterwards, the final logit is divided by the temperature τ, which controls the diversity of the samples.

For continuous tokens, they are conditional vector z_c and unconditional vector z_u for the diffusion loss head. The predicted noise ϵ is then extrapolated as: $\epsilon = \epsilon_\theta(x_t|t, z_u) + \omega\cdot(\epsilon_\theta(x_t|t, z_c) - \epsilon_\theta(x_t|t, z_u))$, where ω is the guidance scale. To control sample diversity via temperature τ, Dhariwal & Nichol (2021) suggests to either divide ϵ_θ by τ or scale the noise with τ. We follow MAR (Li et al., 2024) to adopt the latter option.

Table A2: Optimal guidance scale ω and temperature τ for FID.

Model variants	ω	τ
random order, continuous token	5	0.975
raster order, continuous token	4.5	0.975
random order, discrete token	1.6	1.05
raster order, discrete token	2.5	0.95

We conduct a sweep over the guidance scale ω and temperature τ to determine the optimal combination for each model variant. This sweep is performed for models with 160M and 360M parameters, and we find that the optimal parameters remain consistent across these two scales. Therefore, we apply the same parameters, as shown in Table A2, for models with sizes up to 3B parameters.

Visualization Details of Figure 1. To generate Figure 1, we first used our 10.5B random-order model with continuous tokens to generate 256×256 images conditioned on the text prompts. We then applied an in-house super-resolution model to upscale the images to 1024×1024 for improved visual quality (only for Figure 1). The prompts used, from left to right and top to bottom, are: "Close up photo of a knight", "A baseball bat", "An origami bird made of paper is perched on a branch of an evergreen tree", "A wise old mushroom wearing spectacles and reading a book under a tree", "photo of an eagle with a golden crown resting upon its head", "A beautiful castle beside a waterfall in the woods by Josef Thoma, matte painting, trending on artstation HQ", "A photorealistic image of a beautiful mountain range reflected perfectly on the still surface of a crystal-clear lake, surrounded by lush green meadows and vibrant wildflowers", "Oil painting of a vibrant landscape of rolling hills covered in wildflowers, with a quaint farmhouse nestled in the distance under a bright blue sky", "Mona Lisa on a cake", "A grumpy-looking lemon wearing a raincoat and holding an umbrella, standing in the rain", "A hyperrealistic close-up of an eye, with the iris reflecting a vast and detailed landscape, complete with mountains, rivers, and forests", "A section of the Great Wall in the mountains, detailed charcoal sketch", "A present with a blue ribbon under a Christmas tree".

Ablation study on Text Aligner. Table A3 presents a pilot study on the effect of the trainable text aligner. We trained the smaller model with 277M parameters using random order continuous tokens with a T5-XL text encoder for 500k steps. The FID score shows improvements as more layers are added to the text aligner. We choose to use 6 layers in our Fluid models to balance performance and efficiency.

Table A3: FID vs number of layers in the text aligner.

#layers	FID
0	9.38
3	8.61
6	8.42

C ADDITIONAL QUALITATIVE RESULTS

C.1 IMAGES FROM 10.5B FLUID MODEL

In Figure A6 and A7 we show additional images generated from our 10.5B Fluid model.

C.2 COMPARISON BETWEEN 3.1B AND 10.5B FLUID MODEL

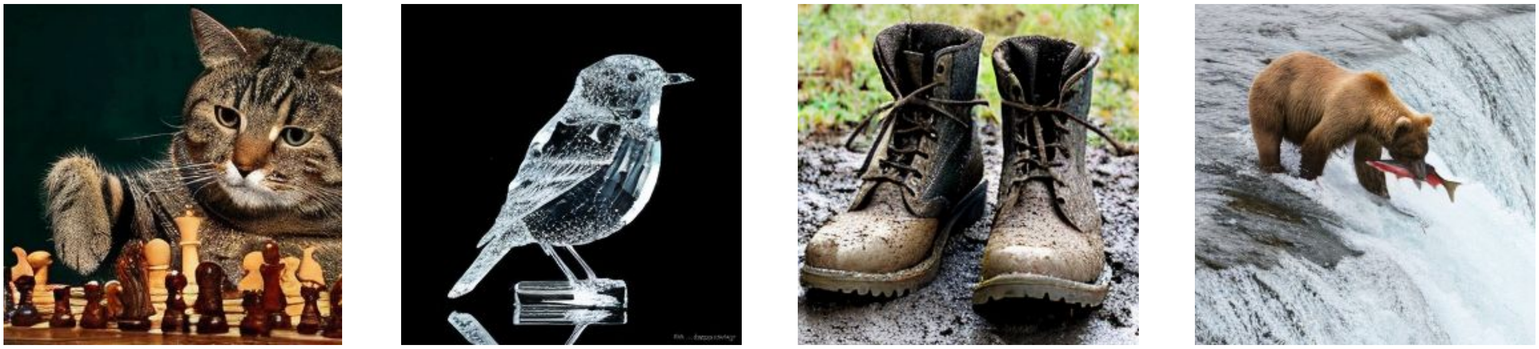
We also show more qualitative comparisons between the images generated by the 3.1B and 10.5B Fluidmodel in Figure A8 and A9. While the 3.1B model performs well in most cases, the 10.5B model demonstrates better ability in generating text and finer image details, and creating images that better align with the corresponding texts.

C.3 COMPARISON BETWEEN FLUID ON 256X256 AND 512X512

To enhance image detail and evaluate our model's performance at higher resolutions, we trained a Fluid 3.1B model to generate 512x512 resolution images. This was achieved by replacing the learned positional embeddings with 2D Rotary Position Embeddings and initially training the model at 256x256 resolution. The 256x256 model served as the initialization for the 512x512 model, with the positional embeddings extended to support the higher resolution. Fine-tuning was conducted for 500k iterations at 512x512 resolution using a reduced learning rate of 1e-5 (0.1x the base learning rate). Figures A10, A11 and A12 compare the 512x512 Fluid 3.1B model with the original 256x256 Fluid 3.1B and 10.5B models, demonstrating that the 512x512 model generates sharper, more detailed images with significantly improved visual quality, while Fluid 10.5B model can generate better text and numbers.

C.4 COMPARISON BETWEEN FLUID AND LLAMAGEN

Figures A13 and A14 present a qualitative comparison between our Fluid 3.1B and 10.5B models and LlamaGen (Sun et al. (2024)). LlamaGen employs a two-stage training process for text-to-

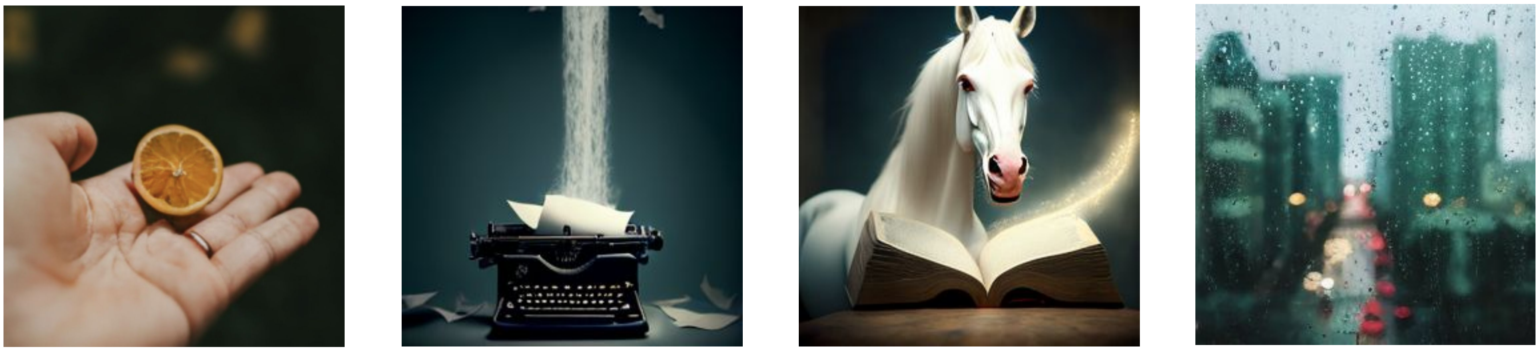


A photo of a cat playing chess.

A bird made of crystal

A pair of old boots covered in mud.

Photo of a bear catching salmon.

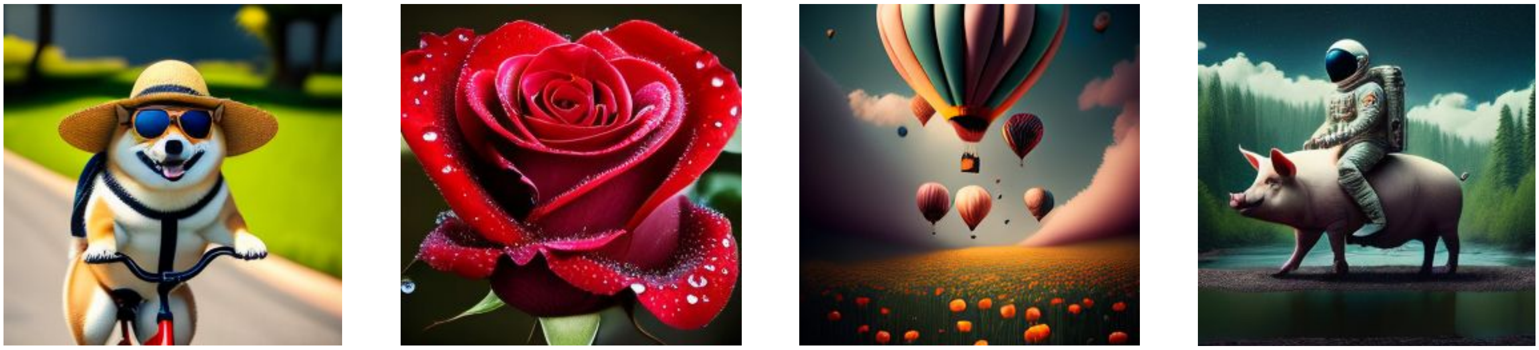


High quality, a close up photo of a human hand

A vintage typewriter with paper spewing out like a waterfall.

A white horse reading a book, fairytale.

A window with raindrops trickling down, overlooking a blurry city.

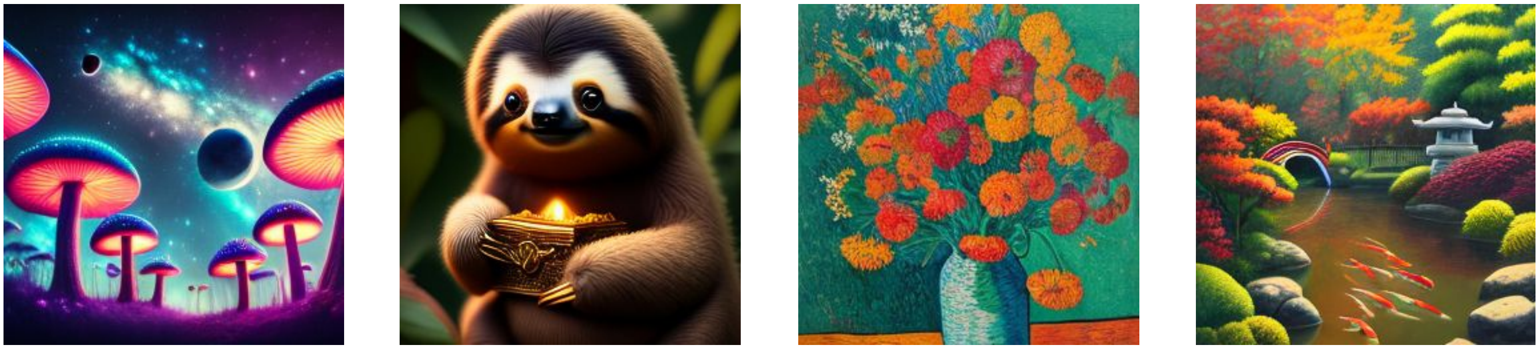


A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.

A close-up photo of a bright red rose, petals scattered with some water droplets, crystal clear.

Hot air balloons and flowers, collage art, photorealism, muted colors, 3D shading beautiful eldritch, mixed media, vaporous

an astronaut rides a pig through in the forest. next to a river, with clouds in the sky



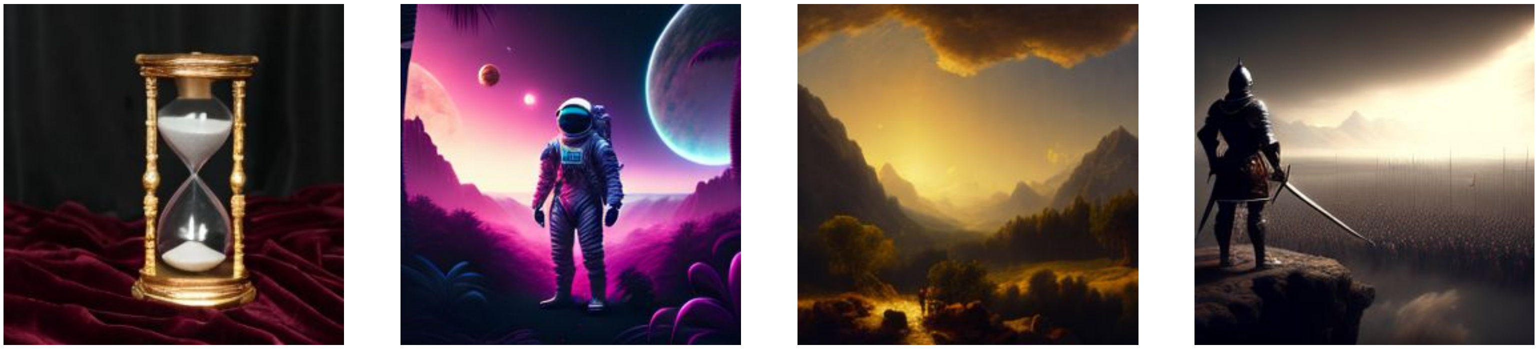
An otherworldly forest of giant glowing mushrooms under a vibrant night sky filled with distant planets and stars, creating a dreamlike, cosmic landscape

A close-up photo of a baby sloth holding a treasure chest. A warm, golden light emanates from within the chest, casting a soft glow on the sloth's fur and the surrounding rainforest foliage.

A still life of a vase overflowing with vibrant flowers, painted in bold colors and textured brushstrokes, reminiscent of van Gogh's iconic style.

A tranquil scene of a Japanese garden with a koi pond, painted in delicate brushstrokes and a harmonious blend of warm and cool colors.

Figure A6: Additional images generated from our 10.5B Fluid model.

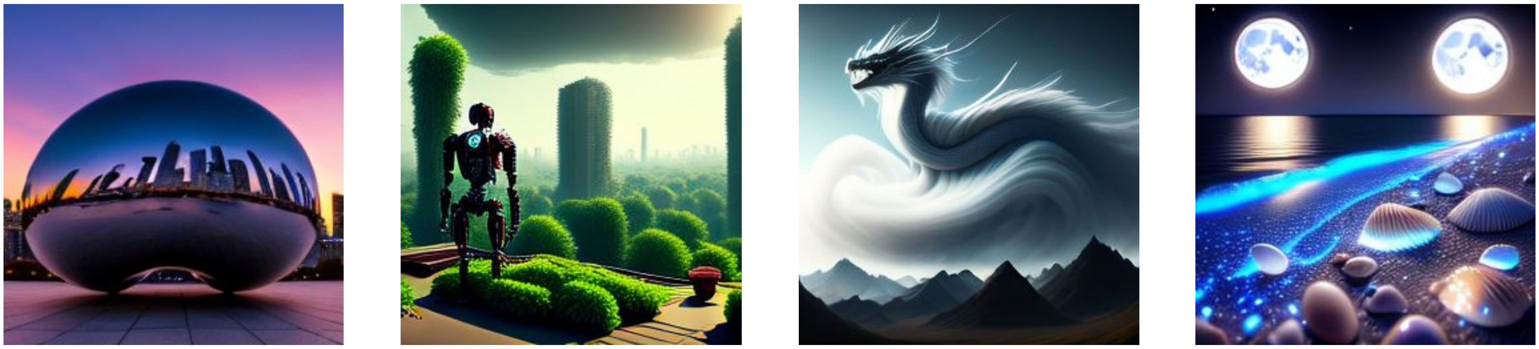


A golden hourglass, half-filled with flowing silver sand, placed on a rich velvet cloth.

A space explorer discovering an alien jungle planet under a purple sky.

A gorgeous mountain landscape at sunset. Masterful painting by Rembrandt

A medieval knight standing on a cliff overlooking a vast battlefield.

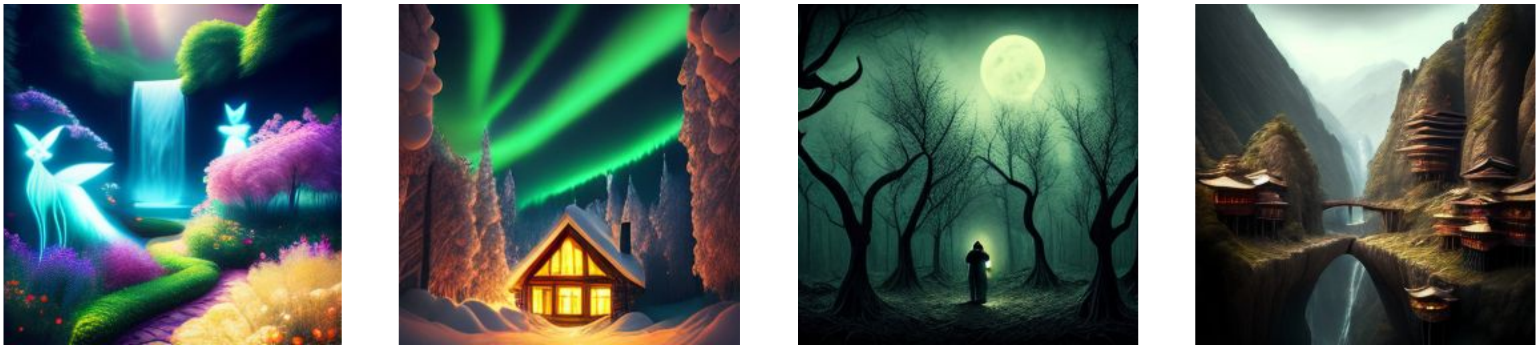


An image of a chrome sphere reflecting a vibrant city skyline at sunset.

A post-apocalyptic city overtaken by vines, with a robot tending a rooftop garden.

A cloud dragon flying over mountains, its body swirling with the wind

A moonlit beach with glowing seashells and two moons reflected on the water.

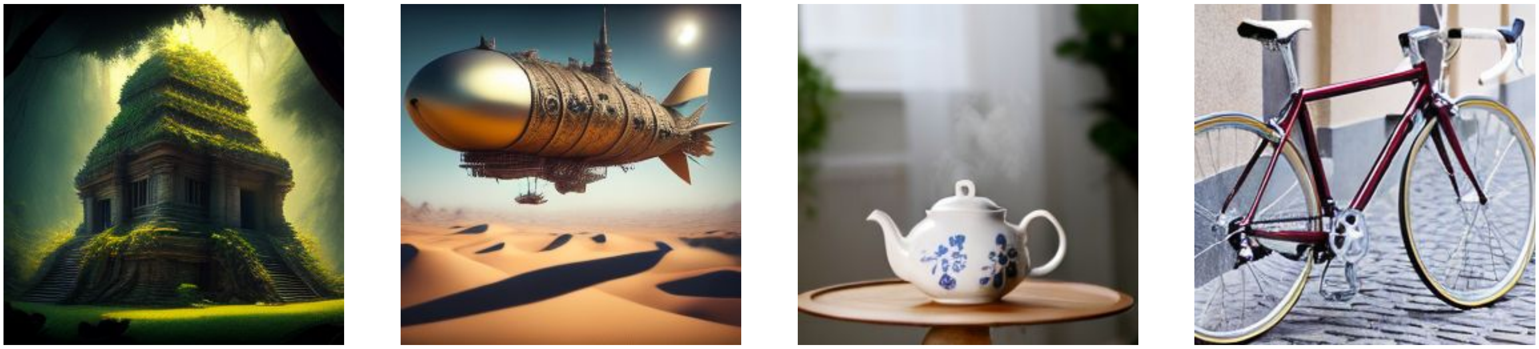


An enchanted garden where every plant glows softly, and creatures made of light and shadow flit between the trees, with a waterfall flowing in the background

A cozy cabin in the middle of a snowy forest, surrounded by tall trees with lights glowing through the windows, a northern lights display visible in the sky.

A dark forest under a full moon, with twisted, gnarled trees, shadows lurking behind every branch, and a lone figure holding a glowing lantern.

A mountain village built into the cliffs of a canyon, where bridges connect houses carved into rock, and waterfalls flow down into the valley below.



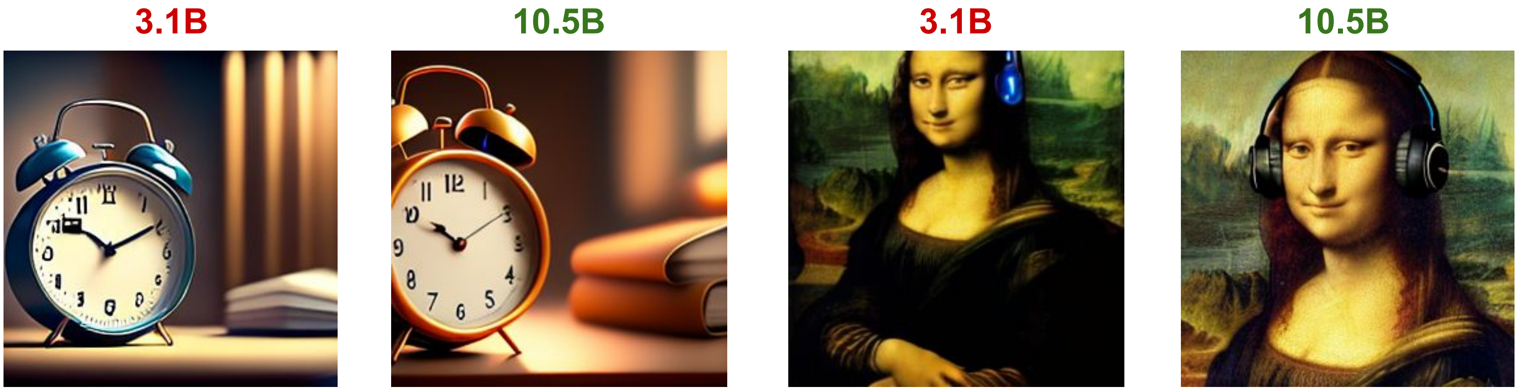
An ancient, overgrown temple hidden deep within a jungle, with vines crawling up its stone walls, and golden light filtering through the thick canopy above.

A steampunk airship soaring over a desert landscape, with mechanical wings and gears turning, casting shadows on the sand dunes below

A pristine white teapot with delicate blue floral designs, steaming with hot tea, sitting on a round wooden table.

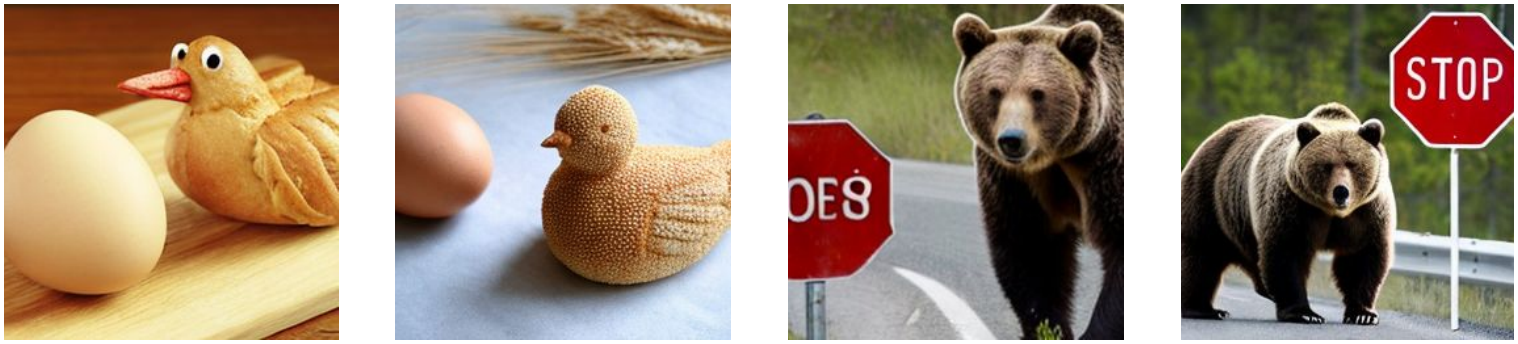
A sleek modern bicycle, leaning against a cobblestone alley, with shiny chrome details and a deep red paint job.

Figure A7: Additional images generated from our 10.5B Fluid model.



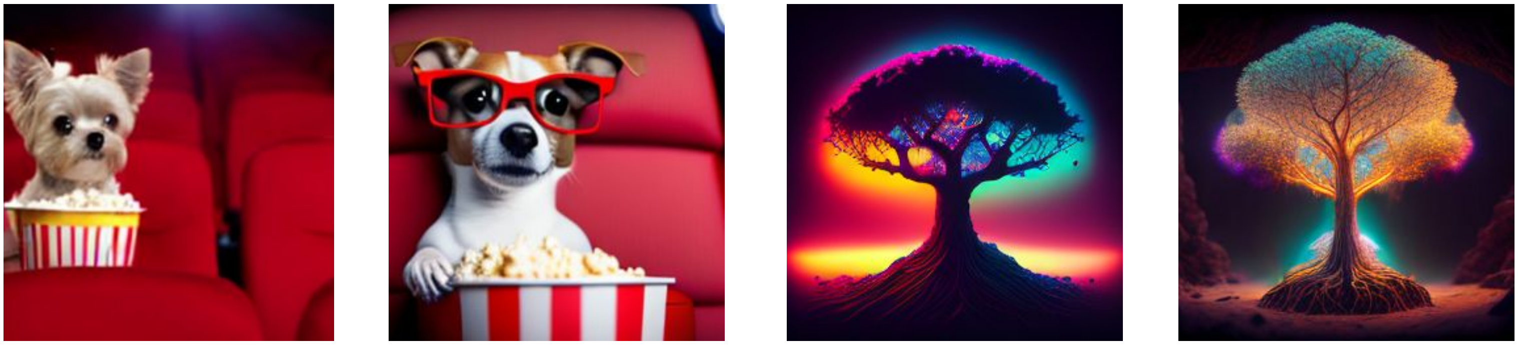
a clock on a desk, cartoon style

Mona Lisa wearing headphone



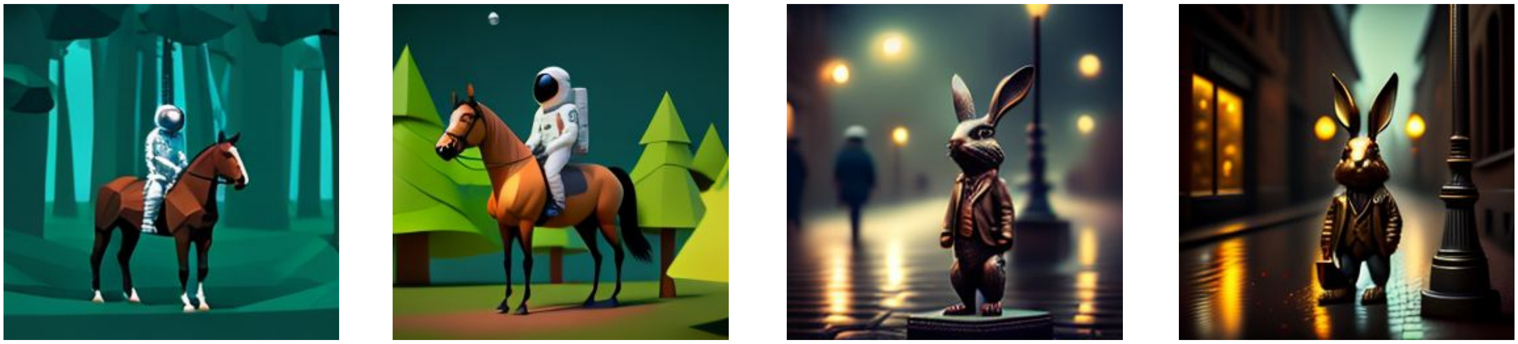
an egg and a bird made of wheat bread

a photo of a bear next to a STOP sign



Cute small dog sitting in a movie theater eating popcorn watching a movie

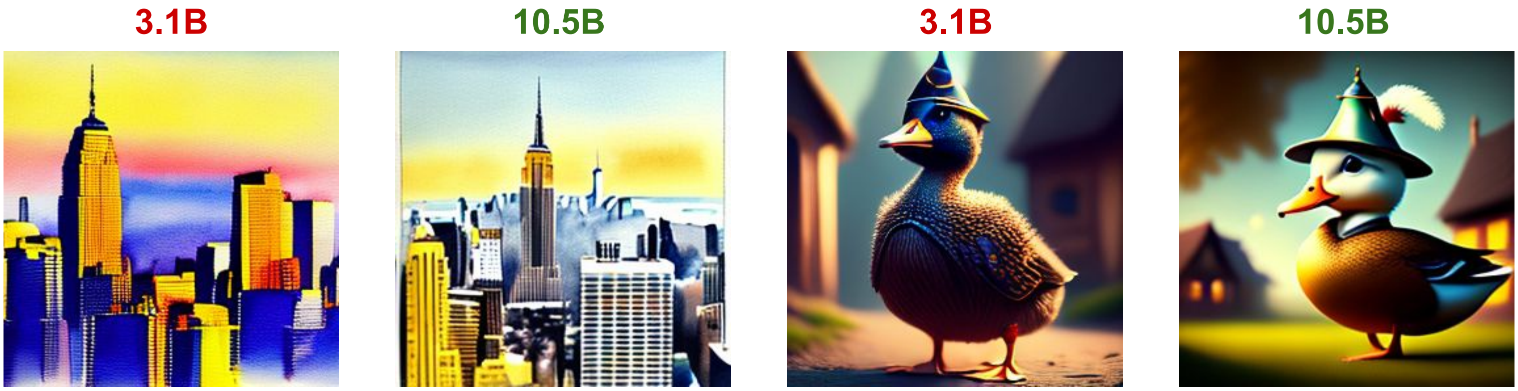
dark high contrast render of a psychedelic tree of life illuminating dust in a mystical cave



Astronaut riding a horse in a forest, low poly

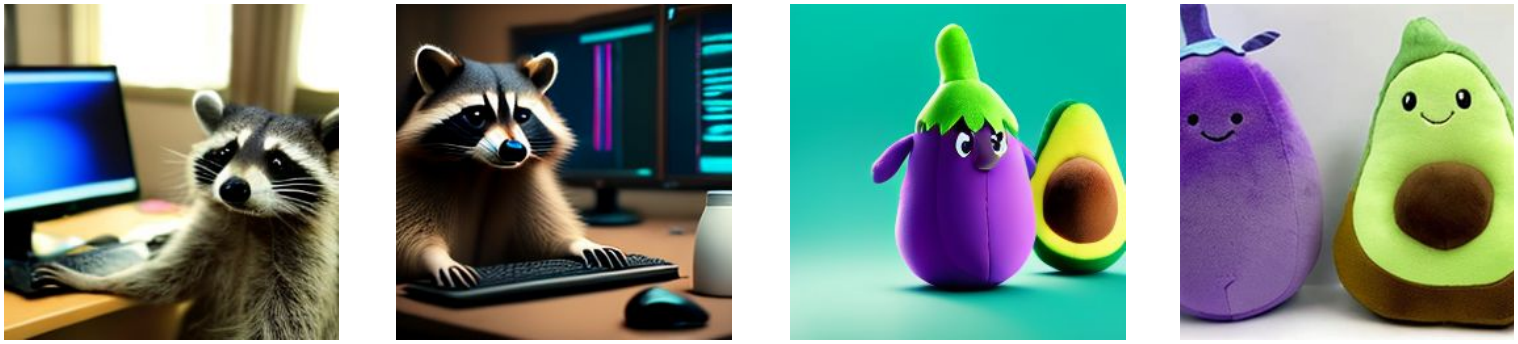
Metal statue of a rabbit detective standing under a street light near a brick lined street on a rainy night. Bokeh

Figure A8: Additional qualitative comparisons between images generated from 3.1B and 10.5B Fluid model.



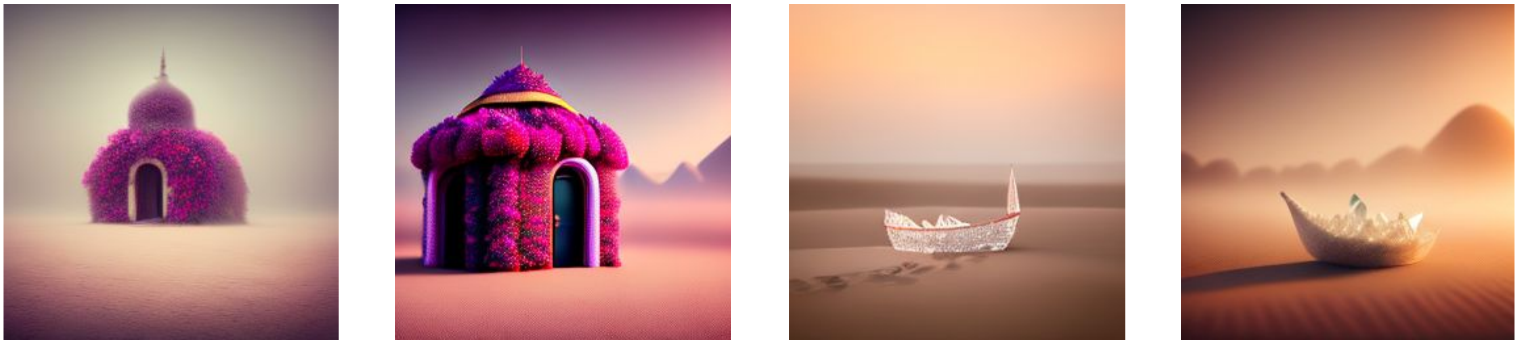
Downtown New York City at sunrise. detailed ink wash

medieval duck that lives in a village, 4k digital art



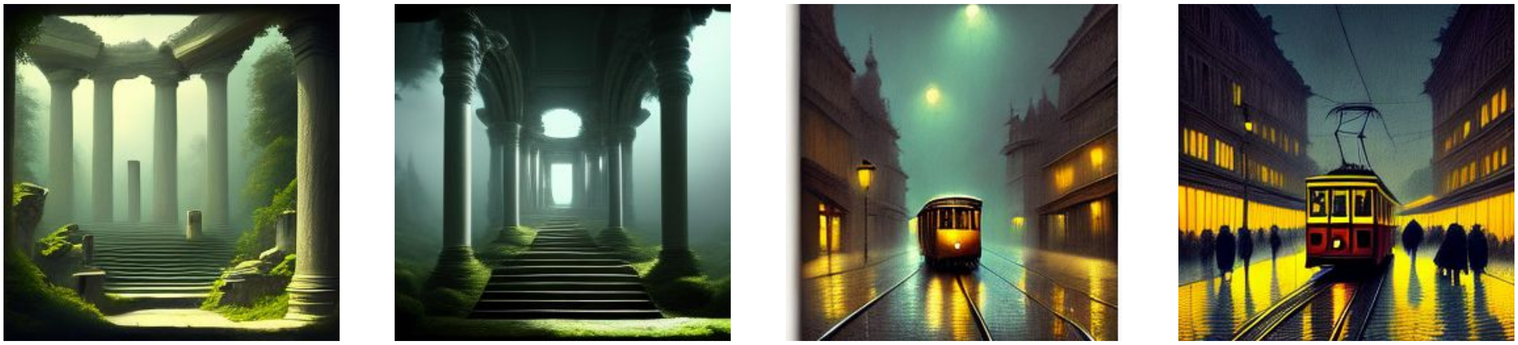
A photo of a confused racoon in computer programming class

an eggplant and an avocado, stuffed toys next to each other



A building made up of flowers, in the middle of the empty sand, pink and purple, dreamy, foggy, photograph

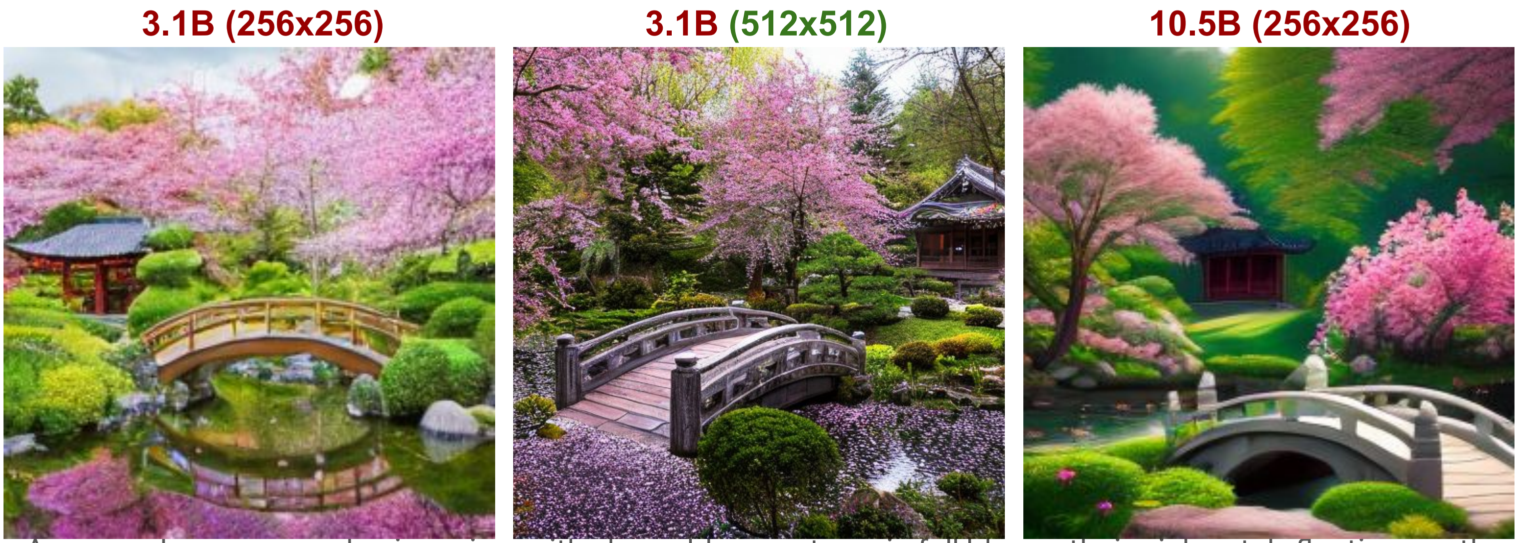
A boat made up of crystals, in the middle of the empty sand, cream and orange, dreamy, foggy, photograph



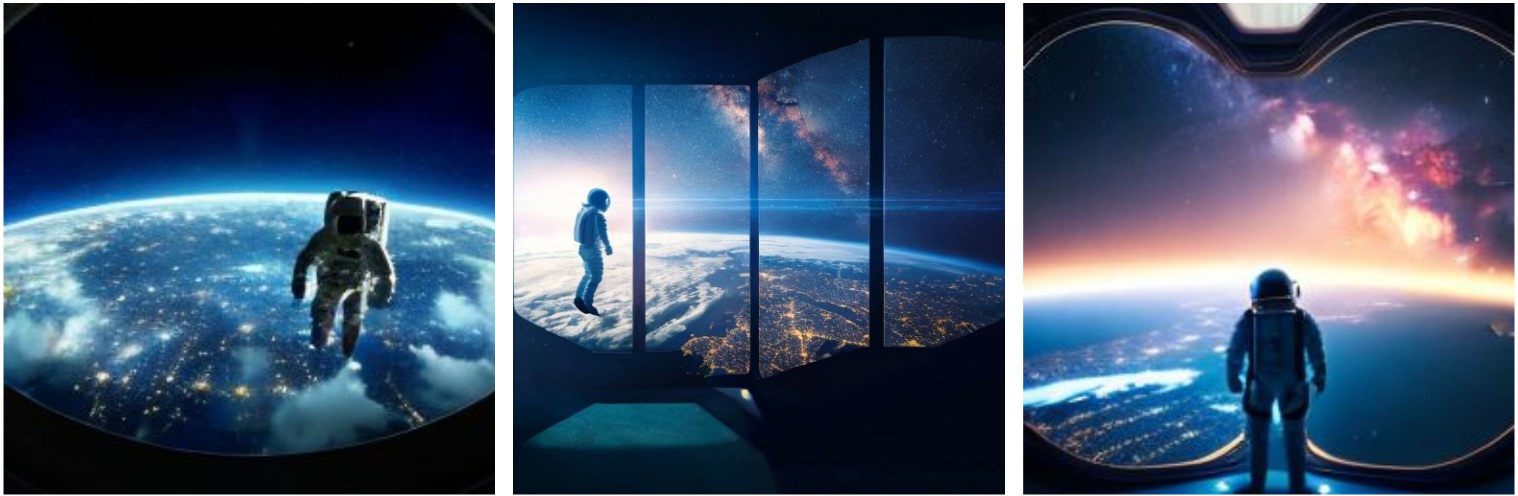
Temple in ruins, epic, forest, stairs, columns, cinematic, detailed, atmospheric, epic, concept art, matte painting, background, mist, photo-realistic, concept art, volumetric light, cinematic epic, 8k

A futuristic street train a rainy street at night in an old European city. Painting by David Friedrich, Claude Monet and John Tenniel

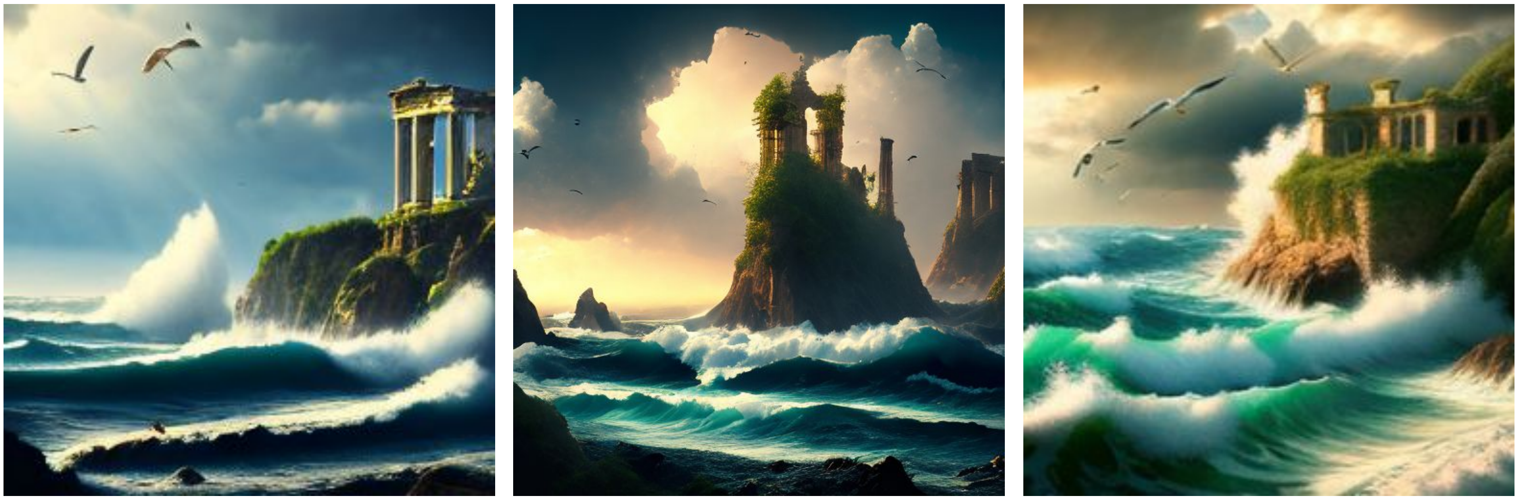
Figure A9: Additional qualitative comparisons between images generated from 3.1B and 10.5B Fluid model.



A serene Japanese garden in spring, with cherry blossom trees in full bloom, their pink petals floating gently in the breeze. A stone bridge arches over a tranquil koi pond, and a traditional tea house sits in the background, surrounded by perfectly manicured greenery.

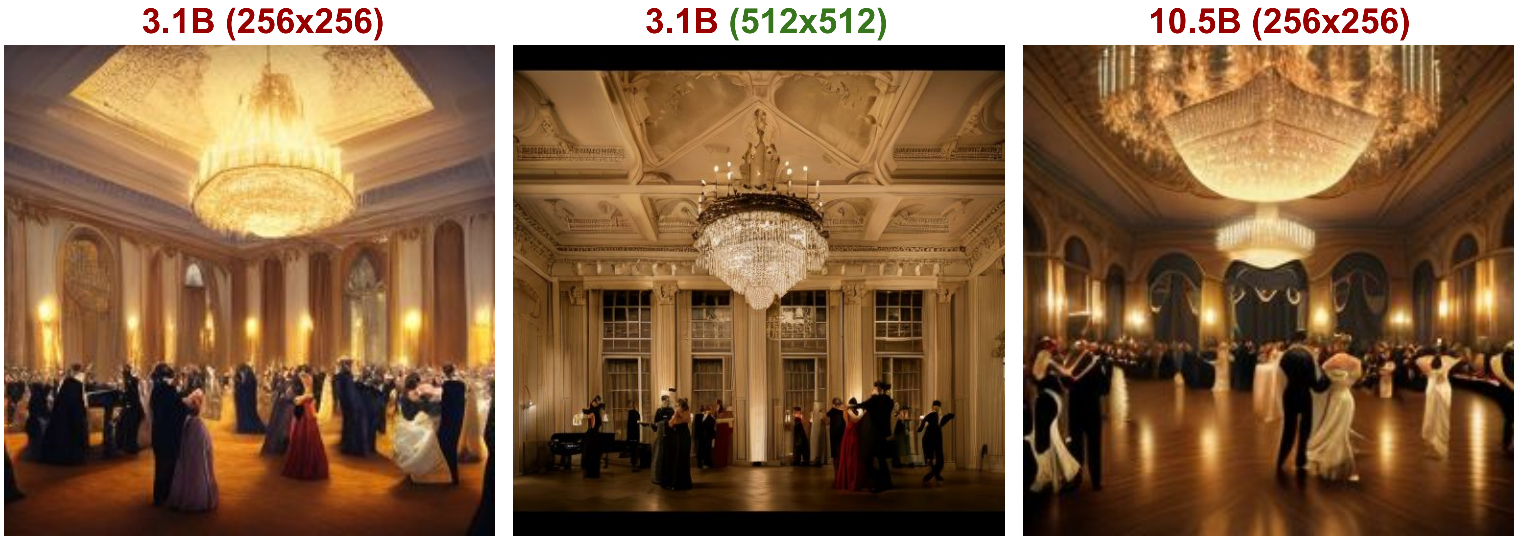


An astronaut floating weightlessly inside a space station observatory, gazing out at a panoramic view of Earth with swirling clouds, city lights, and the vast expanse of stars and galaxies beyond.

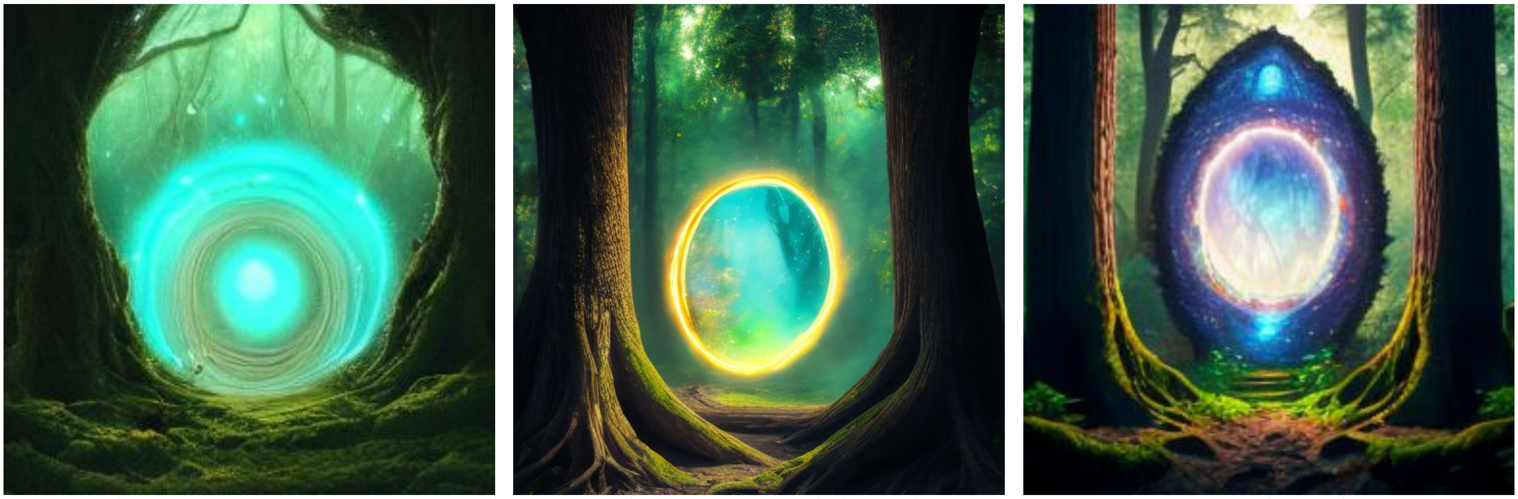


A dramatic seascape at dawn, where towering waves crash against jagged cliffs topped with ancient ruins overgrown with vines, seagulls soaring above, and the first rays of sunlight breaking through stormy clouds.

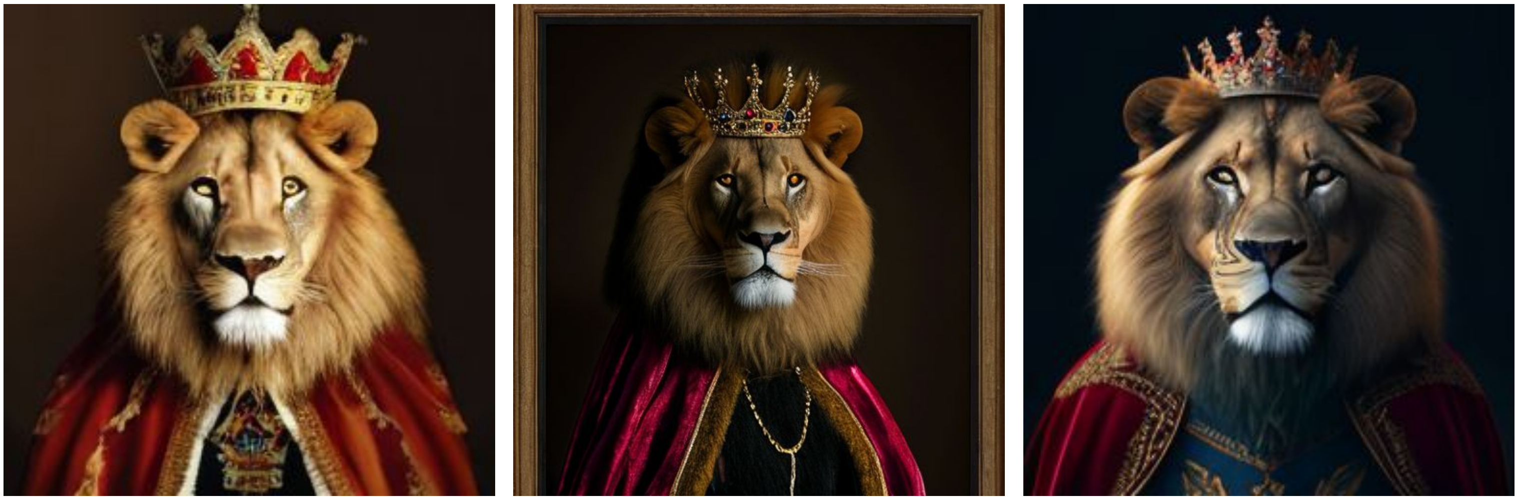
Figure A10: Qualitative comparisons between images generated from 3.1B and 10.5B Fluid model with 256x256 resolution, and 3.1B model with 512x512 resolution.



An elegant ballroom from the 1920s, filled with dancers in vintage attire, a grand chandelier casting warm light, a jazz band playing lively tunes, and Art Deco decorations adorning the walls.

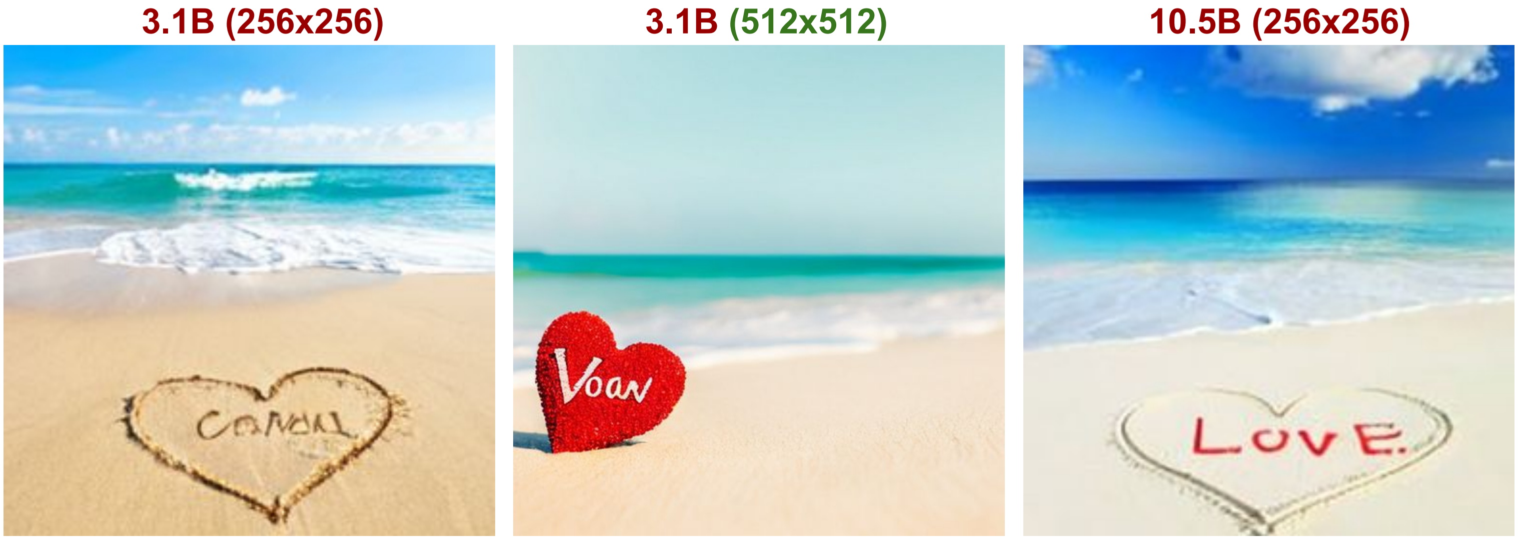


A mystical portal between two ancient trees in a forest, swirling with ethereal energy, revealing glimpses of a fantastical realm beyond.

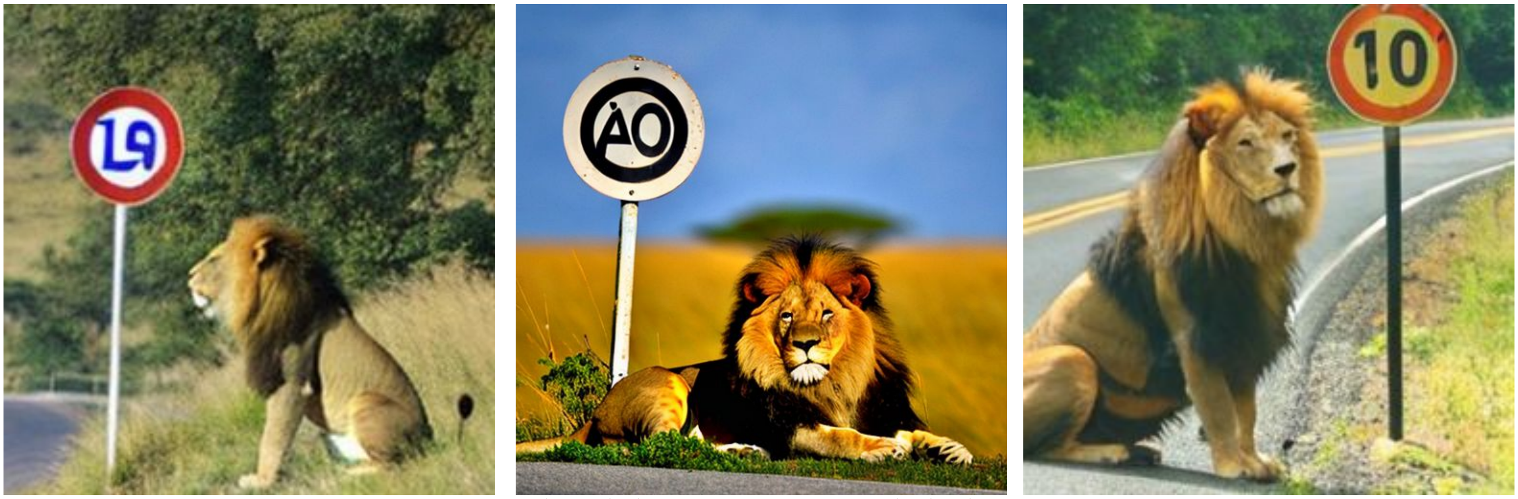


A photo of a lion wearing a regal cape and crown

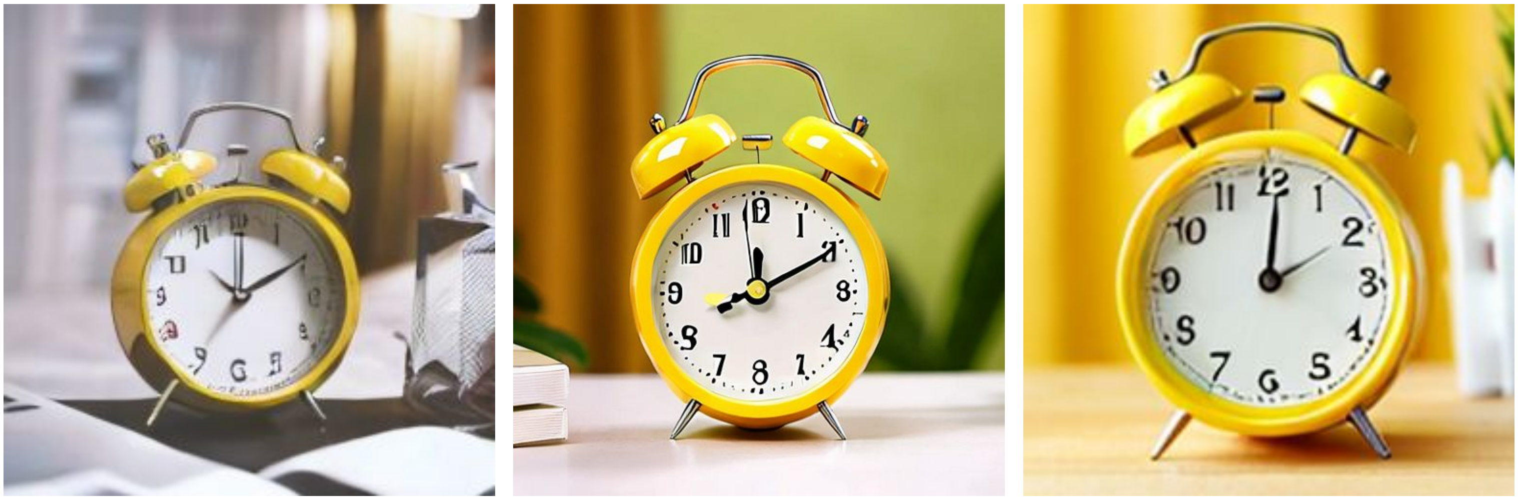
Figure A11: Qualitative comparisons between images generated from 3.1B and 10.5B Fluid model with 256x256 resolution, and 3.1B model with 512x512 resolution.



A heart with “Love” written in it, on the beach



A lion sitting under a speed limit sign



retro style yellow alarm clock with a white clock face

Figure A12: Qualitative comparisons between images generated from 3.1B and 10.5B Fluid model with 256x256 resolution, and 3.1B model with 512x512 resolution.

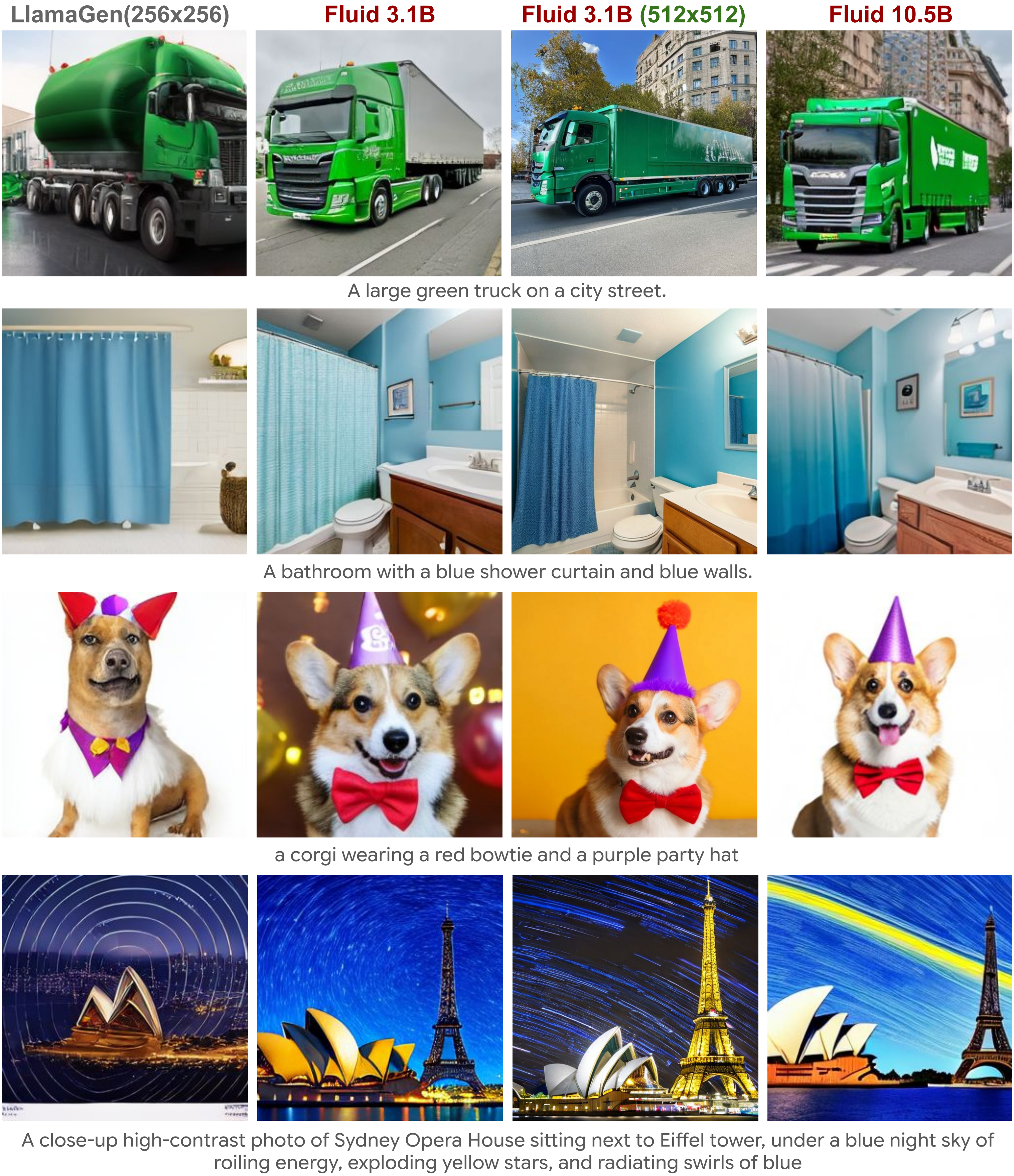

LlamaGen(256x256)
Fluid 3.1B
Fluid 3.1B (512x512)
Fluid 10.5B
A large green truck on a city street.
A bathroom with a blue shower curtain and blue walls.
a corgi wearing a red bowtie and a purple party hat
A close-up high-contrast photo of Sydney Opera House sitting next to Eiffel tower, under a blue night sky of roiling energy, exploding yellow stars, and radiating swirls of blue

Figure A13: Qualitative comparisons between images generated from 3.1B and 10.5B Fluid model and LlamaGen(Sun et al. (2024)) stage 1 model.

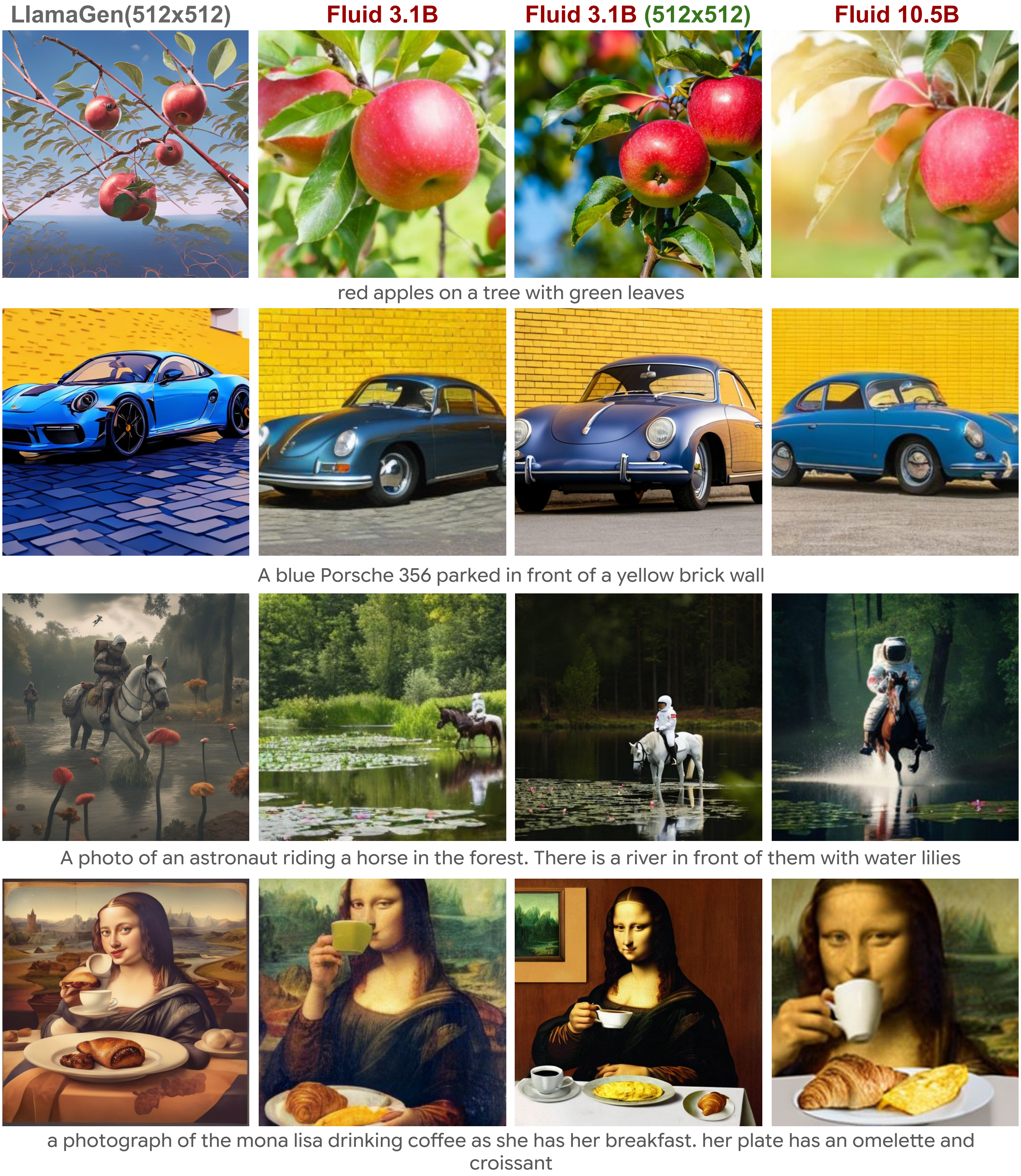
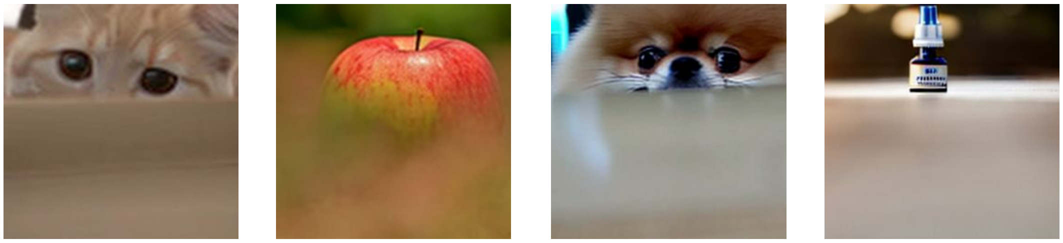

LlamaGen(512x512)
Fluid 3.1B
Fluid 3.1B (512x512)
Fluid 10.5B
red apples on a tree with green leaves
A blue Porsche 356 parked in front of a yellow brick wall
A photo of an astronaut riding a horse in the forest. There is a river in front of them with water lilies
a photograph of the mona lisa drinking coffee as she has her breakfast. her plate has an omelette and croissant

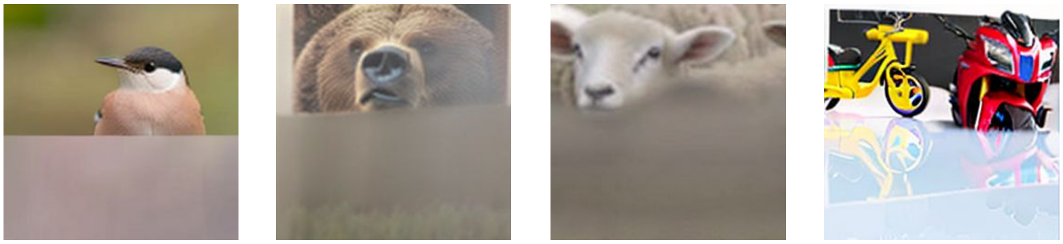
Figure A14: Qualitative comparisons between images generated from 3.1B and 10.5B Fluid model and LlamaGen(Sun et al. (2024)) stage 2 model.

image generation. In Figure A13, we compare our models with LlamaGen's stage 1 model, and in Figure A14, with its stage 2 model. The comparisons show that our Fluid models achieve significantly better alignment with textual descriptions, while LlamaGen struggles to capture certain concepts accurately. For instance, the stage 1 model generates an incorrect dog breed for ”corgi” and ignores the Eiffel Tower mentioned in the text. Similarly, the stage 2 model fails to generate the correct Porsche model, a water lily, and Mona Lisa, demonstrating notable limitations in its concept understanding.

D FAILURE CASES.

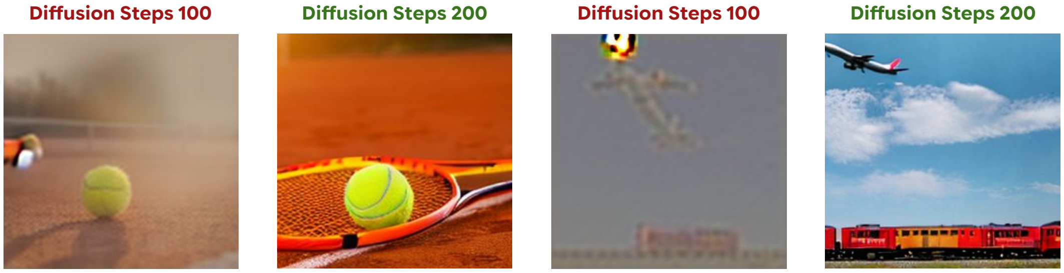


A photo of a cat | A photo of an apple | A photo of a dog | A photo of a bottle

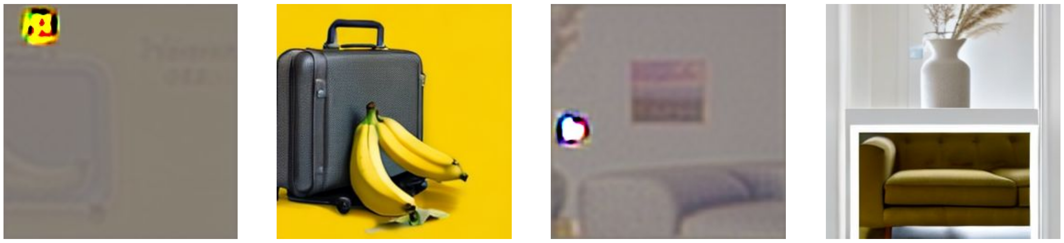


A photo of a bird | A photo of a bear | A photo of a sheep | a photo of a yellow bicycle and a red motorcycle

Figure A15: Failure cases for raster order generation with continuous tokens.



a photo of an orange tennis racket and a yellow sports ball | a photo of a train below an airplane



a photo of a suitcase left of a banana | a photo of a couch below a vase

Figure A16: Failure cases for random order generation with continuous tokens. In very rare cases, an abnormal bright spots can overshadow other tokens. Increasing the diffusion steps solves this issue.

Figure A15 illustrates failure cases for raster order generation using continuous tokens. The model occasionally generates gray tokens in the lower part of the images, and once it begins, it keeps generating gray tokens and rarely recovers. This results in incomplete contents in the generated images. We suspect this issue is because the learned positional embedding for raster order generation struggles to capture the discontinuity between the end token of one line and the start token of the next. This could potentially be addressed by using 2D positional embeddings.

Figure A16 shows a failure case for random order generation with continuous tokens, *i.e*. Fluid, where the model produces abnormal bright spots. This rarely happens, and this issue can be addressed by increasing the diffusion steps, *e.g*., from 100 to 200.

REFERENCES

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv:2207.12598*, 2022.

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

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