arXiv:2610.26863v2 [cs.LG] 20 Oct 2026

TEXT-TO-IMAGE GENERATIVE MODELING WITHOUT LANGUAGE PRETRAINING

Keshigeyan Chandrasegaran[1,2] **Yue Zhao**[1] **Armin W. Thomas**[2]
Ehsan Adeli[1] **Jiajun Wu**[1] **Juan Carlos Niebles**[1] **Li Fei-Fei**[1] **Michael Poli**[1,2]
[1]Stanford University [2]Radical Numerics

ABSTRACT

Modern text-to-image (T2I) systems use internet-scale language pretraining for text-conditioned image generation. *We ask whether language pretraining is necessary for strong T2I generation.* We introduce `Halcyon`, a family of early-fusion multimodal autoregressive models pretrained entirely from scratch on GPIC, a dataset of 100M image-text pairs. `Halcyon`-0.52B outperforms LlamaGen-XL/8 (**48.0** vs. 63.3 gFD) with **18.2**× less pretraining compute. Further, `Halcyon`-1.6B outperforms JiT-T2I-XXL/8++ (**35.3** vs. 41.0 gFD) with **275**× less pretraining compute. Controlled experiments show that text-only pretraining does not improve T2I generation. Halcyon also learns image captioning by replacing caption dropout with I2T supervision, maintaining competitive T2I generation quality under the same pretraining budget. We show that strong T2I systems can be trained from scratch without internet-scale language pretraining.

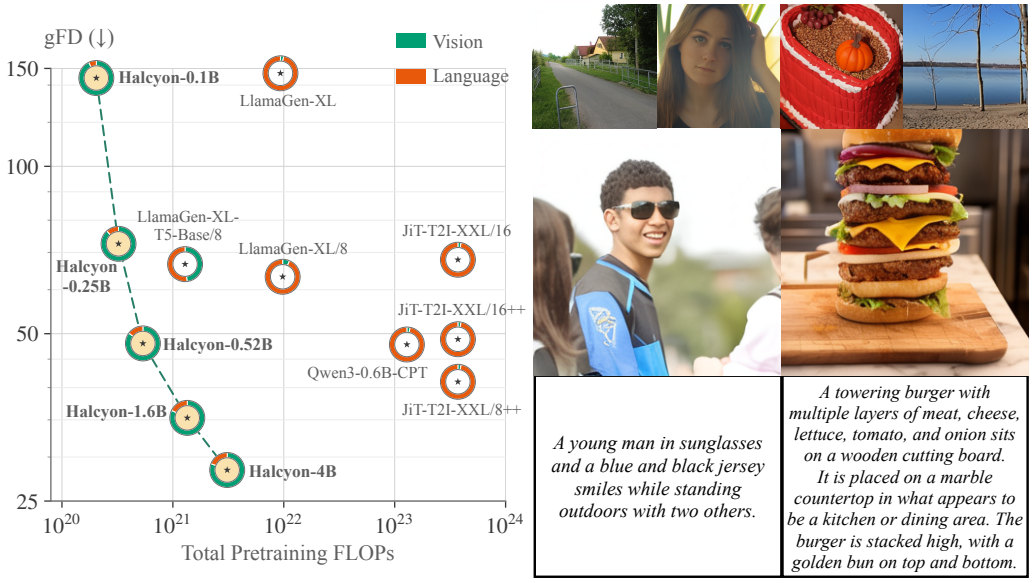


Figure 1: **Text-to-image generation without internet-scale language pretraining. Left:** Generation quality on GPIC (Chandrasegaran et al., 2026) (gFD, log scale) vs. total pretraining FLOPs required to build each T2I system, including language model, image tokenizer, and image generator pretraining. Each ⋆ marks total pretraining FLOPs and gFD, while the donut breaks down language and vision compute. **Halcyon (Ours), trained entirely from scratch, achieves lower gFD with substantially lower total pretraining compute** compared to: *(i) LlamaGen (Sun et al., 2024)*, late-fusion autoregressive models with 0.78–0.90B image generators and pretrained FLAN-T5 encoders (0.11–1.22B); *(ii) JiT-T2I (Li & He, 2026; Ma et al., 2026)*, late-fusion diffusion models with a 1.10B image generator and pretrained Qwen3-1.7B; and *(iii) Qwen3-0.6B-CPT*, an autoregressive model continually pretrained on image–text pairs, initialized from a pretrained Qwen3-0.6B language model (Yang et al., 2025a). **Right:** 256×256 samples generated by Halcyon-4B with prompts shown for two samples. Best viewed in color.

Correspondence to `keshik@cs.stanford.edu`. Code and models are available here.

1 INTRODUCTION

Text-to-image (T2I) generation maps natural-language descriptions to images. Modern T2I systems have an ingrained reliance on internet-scale language pretraining [1] for semantic conditioning of image generators. This dependence is realized via two designs, distinguished by how and when text tokens interact with image tokens: (1) *Late fusion*, the predominant design, uses two separate models: a large model is pretrained on text, and the resulting text embeddings are used to condition a separate image generator through cross- or joint attention (Saharia et al., 2022; Chen et al., 2024; Black Forest Labs, 2025; Zhao et al., 2026); (2) *Early fusion* removes the separate text encoder while retaining language pretraining within a single multimodal model trained on text-only and image-text data, with captions used as a prefix to condition image generation (Team et al., 2024a; Wang et al., 2026; Zhou et al., 2025; Liu et al., 2026). Across both designs, language pretraining dominates the compute budget, accounting for over 80% of total pretraining FLOPs in representative T2I systems (Sun et al., 2024; Liu et al., 2026). This raises two issues. First, its contribution to T2I generation has not been systematically quantified. Second, language pretraining cost is often excluded, understating the total cost of building a T2I system.

In this work, we show that strong T2I generation can be achieved without language pretraining. We introduce `Halcyon`, a family of early-fusion autoregressive models that learn text conditioning directly from image–text pairs using next-token prediction. We quantify the contribution of language pretraining to T2I generation using GPIC (Chandrasegaran et al., 2026), a dataset of 100M image–text pairs, through three experimental axes: (1) **From-scratch pretraining:** we train all components from scratch, including the text and image tokenizers and the generative model; (2) **Data composition and scheduling:** under a fixed pretraining budget, we control the mixture and schedule of text-only, T2I, and I2T examples and measure their effect on T2I generation quality; and (3) **Compute comparisons:** we pretrain representative T2I baselines on GPIC and compare generation quality as a function of total pretraining compute, decomposed into language and vision. Through these experiments, we quantify the utility of language pretraining for T2I generation.

Our results show that `Halcyon` achieves strong T2I generation without language pretraining, with generation Fréchet distance (gFD) consistently improving as model size scales (Figure 1). Particularly, `Halcyon`-0.52B achieves 48.0 gFD, outperforming LlamaGen-XL/8 (FLAN-T5-XL 1.22B + 0.90B image generator; 63.3 gFD) using **18.2× less** total pretraining compute. `Halcyon`-1.6B achieves 35.3 gFD, outperforming JiT-T2I-XXL/8++ (Qwen3-1.7B + 1.10B image generator; 41.0 gFD) using **275× less** total pretraining compute. Our data composition and scheduling experiments further show that external text-only pretraining does not improve T2I generation. Further, we show that `Halcyon` can learn image captioning by replacing caption-dropped T2I examples with I2T sequences, while maintaining competitive T2I generation quality under the same pretraining budget. We provide a simple recipe for pretraining T2I models without language pretraining. Our work prompts rethinking the role of generic language pretraining for multimodal generation.

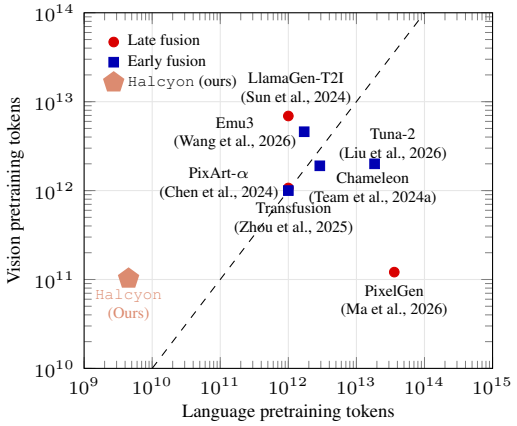


Figure 2: **Language vs. vision pretraining tokens.** Language tokens count text-only pretraining. Emu3 and Tuna-2 values are estimated.

2 RELATED WORK

Modern T2I systems use large-scale language pretraining through two designs. In *late fusion*, the predominant paradigm, a language model is pretrained separately on text-only data, and its embeddings condition a visual generator. Imagen (Saharia et al., 2022), Fluid (Fan et al., 2025) and PixArt-α (Chen et al., 2024) use T5-XXL (Chung et al., 2024), LlamaGen-T2I (Sun et al., 2024) uses T5-XL, SANA (Xie et al., 2025) uses Gemma-2B (Team et al., 2024b) and PixelGen (Ma et al., 2026) uses Qwen3-1.7B. In *early fusion*, the separate text encoder is removed, but text-only

[1] Throughout this paper, *language pretraining* refers to pretraining on web-scale text-only corpora, excluding image-paired captions.

pretraining is incorporated into multimodal pretraining. Chameleon (Team et al., 2024a), Transfusion (Zhou et al., 2025), and Emu3 (Wang et al., 2026) train jointly on text-only and image–text data, while Tuna-2 (Liu et al., 2026) and SimpleAR (Wang et al., 2025) initialize from pretrained Qwen models. Figure 2 compares the text and vision tokens used to pretrain these T2I systems. Systems with language pretraining consume $10^{12} \sim 10^{14}$ text tokens, often exceeding the vision budget. In contrast, `Halcyon` uses no text-only language pretraining; its only language supervision comes from caption tokens paired with images, more than two orders of magnitude lower than the language-pretraining budgets shown. We note that early T2I systems such as DALL-E, CogView, and GLIDE learned text conditioning directly from paired image–text data, without language pretraining (Ramesh et al., 2021; Ding et al., 2021; Nichol et al., 2022). However, the role of language pretraining has been debated: Imagen (Saharia et al., 2022) found benefits from scaling pretrained text encoders, whereas Parti (Yu et al., 2022) found that text-encoder pretraining only marginally reduced T2I generation loss. Similarly, Zhang et al. (2024) found limited benefits from initializing autoregressive T2I models with pretrained language models. Nevertheless, modern T2I systems have largely shifted toward pretrained language models for text conditioning. We quantify the role of language pretraining and show that T2I models trained entirely from scratch, without language pretraining, can outperform strong T2I baselines relying on language pretraining.

3 Approach

`Halcyon` is a decoder-only Transformer with a joint text–image vocabulary. We model text and image tokens as a single sequence using next-token prediction, following prior multimodal early-fusion works (Team et al., 2024a; Wang et al., 2026). All components, including the text tokenizer, image tokenizer, and multimodal generative model are trained from scratch. Below, we describe the tokenizers, sequence format, training objective, and pretraining data composition and scheduling.

3.1 Tokenization

Images. We train a discrete image tokenizer on GPIC images, following the encoder–decoder design of Cosmos Tokenizer (Agarwal et al., 2025) with finite scalar quantization (FSQ) (Mentzer et al., 2024). The tokenizer downsamples images by a factor of 8 along each spatial dimension, with a codebook size of 64,000. Each 256×256 image is represented as $N = 1{,}024$ image tokens $\mathbf{x} = (x_1, \ldots, x_N)$ obtained by flattening the 32×32 grid in raster order.

Text. We train a byte-level BPE tokenizer (Sennrich et al., 2016) from scratch on GPIC captions and OpenWebText Gokaslan & Cohen (2019), with a vocabulary size of 16,384. Each caption is represented as $\mathbf{c} = (c_1, \ldots, c_L)$.

Joint vocabulary. Text tokens, image tokens, and special tokens form a single vocabulary of size 80,640, comprising 16,384 text tokens, 64,000 image tokens, and 256 special tokens. The model uses a shared embedding table and output head. We use `[SOV]` and `[EOV]` to mark the beginning and end of an image, `[EOS]` to mark the end of a sequence, and `[PAD]` for sequence padding.

3.2 Sequence Format and Training Objective

We use four sequence formats for pretraining shown in Figure 3. Text-only sequences are packed, while image-containing sequences are padded to a fixed length with `[PAD]` tokens. We do not pack multiple image–caption pairs into a sequence. For image-only (I) sequences, image tokens are predicted from `[SOV]`. For image-to-text (I2T) sequences, image tokens are predicted first, followed by caption tokens conditioned on the image. For text-to-image (T2I) sequences, image tokens are predicted conditioned on the caption. For classifier-free guidance (CFG) (Ho & Salimans, 2022), we adopt the standard practice of dropping the caption from a fraction p_{drop} of T2I examples to train the unconditional branch. At inference, the conditional and unconditional image-token logits are combined as $\boldsymbol{\ell}_{\text{cfg}} = \boldsymbol{\ell}_{\text{u}} + \gamma(\boldsymbol{\ell}_{\text{c}} - \boldsymbol{\ell}_{\text{u}})$, where γ is the guidance scale. We note that I2T sequences share the same image prefix as unconditional T2I sequences, and therefore also train the unconditional image branch used for CFG. We use weighted next-token cross-entropy, $\mathcal{L} = (\mathcal{L}_{\text{text}} + \lambda_{\text{img}}\mathcal{L}_{\text{img}})/M$, where $\mathcal{L}_{\text{text}} = -\sum_{i \in \mathcal{T}} \log p_\theta(s_i \mid s_{<i})$, $\mathcal{L}_{\text{img}} = -\sum_{i \in \mathcal{I}} \log p_\theta(s_i \mid s_{<i})$, and $M = |\mathcal{T}| + |\mathcal{I}|$. Here, $\mathcal{T}$ and $\mathcal{I}$ denote the non-padding text and image tokens respectively, with special tokens included in $\mathcal{T}$. We set $\lambda_{\text{img}} = 0.5$ to prevent image tokens dominating the training loss similar to Emu3 (Wang et al., 2026).

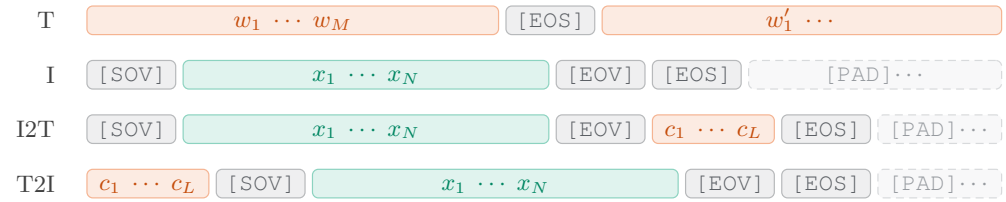


Figure 3: **Sequence formats.** Text-only pretraining tokens, caption tokens, and image tokens occupy one joint vocabulary alongside special tokens. The top two formats are unimodal, while the bottom two represent paired image-caption examples. Rows I and I2T share the same `[SOV]` x `[EOV]` prefix, which is also obtained when dropping the caption from a T2I sequence used for classifier-free guidance. Image-caption sequences are padded to a fixed length, with `[PAD]` positions excluded from the training loss. Box widths are schematic and not proportional to token counts.

3.3 Pretraining Data Composition and Scheduling

To measure the utility of language pretraining for T2I generation, we control the composition and scheduling of text-only, T2I, and I2T examples under a **fixed** pretraining budget. Figure 4 illustrates the five configurations. Unless otherwise specified, caption dropout is applied to T2I examples during training at rate p_{drop}.

- **Baseline T2I.** We train exclusively on T2I examples.
- **Staged: Language → T2I.** We allocate the first α fraction of training to text-only pretraining, followed by T2I for the remaining $1 - \alpha$ fraction. We choose α such that the text-only pretraining tokens exceed the Chinchilla-optimal token budget for the model size (Hoffmann et al., 2022). This setup studies language-first pretraining schedules used in early-fusion systems such as Tuna-2 (Liu et al., 2026). Separately, we evaluate continual pretraining recipe from an internet-scale pretrained LLM by expanding its vocabulary with image tokens and continuing training on paired image–text data.
- **Staged: I2T → T2I.** We allocate the first β fraction of training to I2T, followed by T2I for the remaining $1 - \beta$ fraction.
- **Mixing: T2I + I2T.** We interleave T2I and I2T examples throughout training, allocating a fraction η to I2T and $1 - \eta$ to T2I.
- **Mixing: T2I without caption dropout + I2T.** We set $\eta = p_{\text{drop}}$ and disable caption dropout. This preserves the baseline's conditional and unconditional image-training tokens while additionally providing I2T supervision.

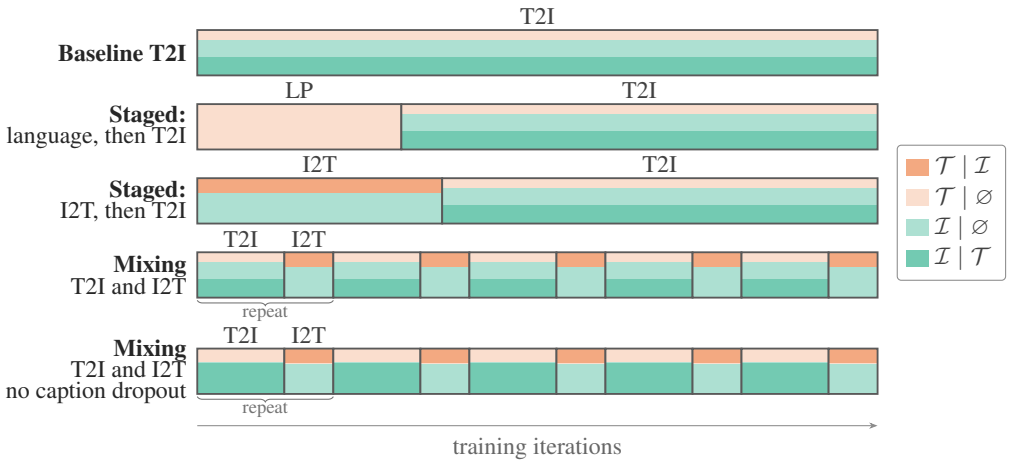


Figure 4: **Pretraining data composition and scheduling described in Section 3.3.** The four token types are: image tokens conditioned on text ($\mathcal{I} \mid \mathcal{T}$); image tokens without text conditioning ($\mathcal{I} \mid \varnothing$); text tokens conditioned on images ($\mathcal{T} \mid \mathcal{I}$); and text-only tokens ($\mathcal{T} \mid \varnothing$). Vertical boundaries denote training phases, while horizontal bands show the token composition within each phase. Bands represent interleaved examples. Band height and width are schematic.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Dataset and pretraining protocol. We pretrain on GPIC (Chandrasegaran et al., 2026), a dataset of approximately 100M image–text pairs, at 256×256 resolution. All configurations are trained for one epoch on GPIC with a global batch size of 1,024 and an identical warmup-stable-decay (WSD) learning-rate schedule (Hu et al., 2024a). For language pretraining, we use ClimbMix (Diao et al., 2025) packed without padding. Model details and training details are provided in Appendix A.

Tokenizer training and evaluation. • *Image tokenizer.* We train our image tokenizer for one epoch using reconstruction and perceptual losses, followed by an additional half epoch with adversarial loss using a PatchGAN discriminator. Compared to Cosmos-DI-8x8 (Agarwal et al., 2025), we increase the decoder width from 128 to 192. To assess reconstruction quality, we compare our tokenizer with pretrained LlamaGen-VQ-16x16 and Cosmos-DI-8x8 tokenizers. We include Real-vs-Real scores as a reference. Our tokenizer outperforms Cosmos-DI-8x8 across all metrics (Table 1). As with prior discrete tokenizers, reconstruction fidelity is lower for small text and fine glyphs (Team et al., 2024a; Sun et al., 2024; Agarwal et al., 2025).

Table 1: **Reconstruction results on GPIC Test-50K (256×256).** Our tokenizer, trained from scratch on GPIC, outperforms Cosmos-DI-8x8 on all metrics.

Tokenizer	rFD ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LlamaGen-VQ-16x16	129.87	20.18	0.560	0.146
Cosmos-DI-8x8	37.93	25.38	0.776	0.153
`Halcyon`	**18.86**	**25.61**	**0.792**	**0.062**
Real-vs-Real	7.59	–	–	–

• *Text tokenizer.* Despite having a $9.3\times$ smaller vocabulary than Qwen3, our tokenizer achieves comparable compression on GPIC test captions (4.774 vs. 4.763 bytes/token), outperforming the FLAN-T5 tokenizer (4.152 bytes/token; Table 2). On held-out OpenWebText, our tokenizer achieves 4.086 bytes/token. OpenWebText is used only to train the BPE tokenizer, not the multimodal generative models. Full tokenizer training details are provided in Table 14.

Table 2: **Text tokenizer comparison on GPIC test captions.** Higher bytes/token and lower tokens/caption indicate better compression.

Tokenizer	Vocab.	Bytes/token ↑	Tok./caption ↓
FLAN-T5	32,100	4.152	51.1
Qwen3	151,669	4.763	44.6
`Halcyon`	**16,384**	**4.774**	**44.5**

Model configurations. We train `Halcyon` at five sizes: 0.1B, 0.25B, 0.52B, 1.6B, and 4B parameters. Model architecture details are provided in Table 11. We use a sequence length of 1280 for all experiments. We use $p_{drop} = 0.1$ unless otherwise specified.

Late-fusion methods. Late fusion is the predominant design choice for modern T2I systems. Selecting baselines for this study is non-trivial: existing systems differ in pretraining objective, language conditioning, image representation, and training setup, while our study requires public pretraining code and models that enable controlled interventions to study the role of language pretraining. We select two representative families, LlamaGen (Sun et al., 2024) and JiT-T2I (Li & He, 2026; Ma et al., 2026). They cover (i) autoregressive and diffusion-based generation, (ii) encoder-decoder and decoder-only language models of different sizes, and (iii) image sequence lengths of 256 and 1,024. For both families, we use the authors' official codebases and follow their training configurations unless stated otherwise. Below, we describe our setup and summarize them in Table 3.

• **LlamaGen (Sun et al., 2024).** This method uses a FLAN-T5-XL text encoder for language modeling, a discrete image tokenizer, and an autoregressive image generator.

- **Default (LlamaGen-XL).** We use the original LlamaGen configuration: FLAN-T5-XL, the pretrained image tokenizer producing 256 image tokens from a codebook of 16,384 entries, and the 0.78B image generator.
- **Use `Halcyon` image tokenizer (LlamaGen-XL/8).** Starting with the default configuration, we replace the image tokenizer with our GPIC-trained tokenizer. This increases the visual vocabulary size by $4\times$, the image sequence length from 256 to 1,024, and the image generator size from 0.78B to 0.90B.
- **Use smaller text encoder (LlamaGen-XL-T5-Base/8).** Starting with LlamaGen-XL/8, we replace FLAN-T5-XL (1.22B) with FLAN-T5-base (0.11B) to study the effect of language model size.

Table 3: **Experiment configurations for late-fusion methods.** LlamaGen and JiT-T2I configurations, varying the language model, image sequence length, and perceptual supervision. LlamaGen and `Halcyon` are autoregressive; JiT is diffusion-based. Pretraining tokens report the total language and vision tokens used during training; `Halcyon` uses no pretrained language model, and all its language tokens come from GPIC captions. We report encoder parameters for FLAN-T5.

	Language	Params		Pretraining tokens		Perceptual
Configuration	model	Language	Vision	Language	Vision	loss
LlamaGen-XL	FLAN-T5-XL	1.22B	0.78B	∼1T	25.6B	✓
LlamaGen-XL/8	FLAN-T5-XL	1.22B	0.90B	∼1T	102.4B	✓
LlamaGen-XL-T5-Base/8	FLAN-T5-Base	0.11B	0.90B	∼1T	102.4B	✓
JiT-T2I-XXL/16	Qwen3	1.70B	1.10B	36T	25.6B	
JiT-T2I-XXL/16 ++	Qwen3	1.70B	1.10B	36T	25.6B	✓
JiT-T2I-XXL/8 ++	Qwen3	1.70B	1.10B	36T	102.4B	✓
`Halcyon` (Ours)	—	0.1–4B		4.5B	102.4B	✓

• **JiT-T2I (Li & He, 2026; Ma et al., 2026).** This method uses pretrained Qwen3-1.7B (Yang et al., 2025a) as the language model and a pixel-space flow-matching image generator.

- **Default (JiT-T2I-XXL/16).** We use the original JiT-T2I-XXL/16 configuration: Qwen3-1.7B and no image tokenizer, with 16×16 image patches producing 256 image tokens.
- **Add perceptual loss (JiT-T2I-XXL/16 ++).** Starting with the default configuration, we add LPIPS loss (Zhang et al., 2018) to train the image generator. This tests the effect of perceptual supervision used to also train the image tokenizers in `Halcyon` and LlamaGen.
- **Use 8×8 patches (JiT-T2I-XXL/8 ++).** Starting with JiT-T2I-XXL/16 ++, we reduce the image patch size from 16×16 to 8×8, increasing the image sequence length from 256 to 1,024, matching `Halcyon` and the modified LlamaGen configurations.

Early-fusion methods. Existing early-fusion models, including Chameleon (Team et al., 2024a) and Emu3 (Wang et al., 2026), incorporate language pretraining into multimodal generative modeling. We study the role of language pretraining in two controlled settings. First, we vary the composition and scheduling of text-only, T2I, and I2T sequences within `Halcyon` under a fixed pretraining budget (Section 3.3). Second, we study continual pretraining by initializing from Qwen3-0.6B-Base, pretrained on 36T text tokens, expanding its vocabulary with image tokens, and continuing pretraining on GPIC using the baseline T2I recipe. For comparison, we train the same architecture and vocabulary from scratch under an identical T2I pretraining budget. We disable weight decay for embedding parameters when vocabulary size is large (216K).

Evaluation protocol. We evaluate all models at 256×256 resolution on three benchmarks: • **GPIC (primary metrics).** We follow the GPIC evaluation protocol using the GPIC Test-50K captions[2], reporting generation Fréchet distance (gFD), Precision, Recall, Density, Coverage, and MMD. • **GenEval (Ghosh et al., 2023).** We evaluate compositional generation using the rephrased prompts from PixelGen (Ma et al., 2026)[3], removing generic phrases such as "an image of" and "the image depicts". • **DPG-Bench (Hu et al., 2024b).** We evaluate complex prompt adherence on DPG-Bench using the original prompts. • **Token-level NLL.** For `Halcyon`, we report negative log-likelihood (NLL) on GPIC Test-50K for four settings: text-conditioned images ($\mathcal{I} \mid \mathcal{T}$), unconditional images ($\mathcal{I} \mid \varnothing$), image-conditioned text ($\mathcal{T} \mid \mathcal{I}$), and unconditional text ($\mathcal{T} \mid \varnothing$). • **Image captioning.** We evaluate image–text alignment using Qwen3-VL-Reranker-8B scores over GPIC Test-50K images and generated captions, treating the image as the query and the caption as the document.

For each model, we sweep guidance scales on GPIC Validation-50K[4] and use the scale achieving the lowest gFD for all three benchmarks. We report the sampling and guidance settings for each model in Appendix A.

[2] `hf.co/datasets/stanford-vision-lab/gpic/blob/main/gpic_test_50k.jsonl`
[3] `github.com/Zehong-Ma/PixelGen/blob/main/evaluations/geneval/`
[4] `hf.co/datasets/stanford-vision-lab/gpic/blob/main/gpic_val_50k.jsonl`

4.2 RESULTS: COMPARISON WITH LATE-FUSION T2I METHODS

We report quantitative results in Table 4, including compositional generation benchmarks (GenEval and DPG-Bench) in Table 5. We include generated samples in Figure 5 and Appendix D. Figure 1 compares generation quality as a function of total pretraining FLOPs, with the exact compute breakdown reported in Table 15. Below, we discuss our key findings:

Table 4: **Comparison of `Halcyon` with late-fusion T2I methods on GPIC.** Without language pretraining, `Halcyon` achieves stronger T2I generation than the late-fusion baselines: `Halcyon`-0.52B outperforms LlamaGen-XL/8 (48.0 vs. 63.3 gFD), while `Halcyon`-1.60B outperforms JiT-T2I-XXL/8++ (35.3 vs. 41.0 gFD). Vision parameter counts refer to the generative models, excluding image tokenizers (72M for LlamaGen and 136M for `Halcyon`). The CFG scale for each model is reported in Appendix Table 16.

Model	Language	Vision	gFD ↓	Prec. ↑	Rec. ↑	Dens. → 1	Cov. ↑	MMD ↓
LlamaGen-XL	1.22B	0.78B	147.01	0.914	0.715	0.925	0.822	0.174
LlamaGen-XL/8	1.22B	0.90B	63.29	0.948	0.827	1.058	0.925	0.041
LlamaGen-XL-T5-Base/8	0.11B	0.90B	66.69	0.941	0.822	1.016	0.924	0.045
JiT-T2I-XXL/16	1.70B	1.10B	67.92	0.934	0.822	0.979	0.923	0.038
JiT-T2I-XXL/16 ++	1.70B	1.10B	48.86	0.947	0.861	1.022	0.939	0.022
JiT-T2I-XXL/8 ++	1.70B	1.10B	40.97	0.949	0.880	1.049	0.952	0.016
`Halcyon` (Ours)	0.10B		144.16	0.924	0.636	1.029	0.853	0.131
	0.25B		72.55	0.946	0.796	1.050	0.922	0.050
	0.52B		48.02	0.953	0.858	1.064	0.938	0.027
	1.60B		35.28	0.955	0.891	1.071	0.955	0.019
	4.00B		28.43	0.951	0.905	1.034	0.960	0.014

Effect of image representation and perceptual supervision. For LlamaGen-XL, replacing its original image tokenizer with our GPIC-trained tokenizer increases the image sequence length from 256 to 1,024 and reduces gFD (147.0 → 63.3). For JiT-T2I-XXL/16, adding LPIPS perceptual supervision reduces gFD (67.9 → 48.9). Reducing the image patch size from 16×16 to 8×8 increases the image sequence length from 256 to 1,024 and further reduces gFD (48.9 → 41.0). We use LlamaGen-XL/8 and JiT-T2I-XXL/8++ for subsequent comparisons with `Halcyon`.

Effect of language encoder size. We study whether a larger pretrained language encoder improves T2I generation. For LlamaGen, replacing FLAN-T5-XL with FLAN-T5-base reduces the language encoder size from 1.22B to 0.11B while retaining the same image tokenizer and 0.90B image generator. The resulting gFD changes only marginally (63.3 → 66.7), despite a 13.5× reduction in language-model training compute[5]. This provides initial evidence that an order-of-magnitude increase in language pretraining compute yields only marginal improvements in T2I generation. We next investigate whether strong T2I generation can be achieved without language pretraining altogether.

Table 5: **Comparison of `Halcyon` with late-fusion T2I methods on GenEval and DPG-Bench.** We also report GPIC gFD for comparison. Per-category breakdowns of GenEval and DPG-Bench are provided in Sec. C.1.

Model	GPIC gFD ↓	GenEval ↑	DPG-Bench ↑
LlamaGen-XL	147.01	30.03	65.40
LlamaGen-XL/8	63.29	30.48	67.08
LlamaGen-XL-T5-Base/8	66.69	28.48	64.67
JiT-T2I-XXL/16	67.92	36.27	72.69
JiT-T2I-XXL/16 ++	48.86	41.79	76.58
JiT-T2I-XXL/8 ++	40.97	43.17	76.85
`Halcyon`-0.10B	144.16	27.31	66.36
`Halcyon`-0.25B	72.55	35.72	72.66
`Halcyon`-0.52B	48.02	44.91	76.84
`Halcyon`-1.60B	35.28	53.48	77.90
`Halcyon`-4.00B	28.43	56.39	77.90

T2I generation without language pretraining. We now address the central question of this work: *can a T2I model trained entirely from scratch match or surpass strong late-fusion systems that rely on pretrained language models?* `Halcyon`-0.52B, a single multimodal model trained entirely from scratch, achieves 48.0 gFD, outperforming LlamaGen-XL/8 with pretrained FLAN-T5-XL (63.3) and approaching JiT-T2I-XXL/8++ with pretrained Qwen3-1.7B (41.0). Increasing `Halcyon` to 1.6B parameters further reduces gFD to 35.3, surpassing both late-

[5]Pretraining and finetuning the full FLAN-T5-XL encoder–decoder model requires 9.06×10^{21} FLOPs vs. 6.69×10^{20} FLOPs for FLAN-T5-base (Chung et al., 2024)

fusion methods. At 4B parameters, gFD decreases to 28.4. On compositional generation benchmarks, Halcyon-0.52B outperforms JiT-T2I-XXL/8++ on GenEval (44.9 vs. 43.2) and matches its DPG-Bench score (76.84 vs. 76.85). At 1.6B, Halcyon surpasses JiT-T2I-XXL/8++ on both GenEval (53.5 vs. 43.2) and DPG-Bench (77.9 vs. 76.9). Moreover, Halcyon-1.6B achieves these results with 275× less total pretraining compute than JiT-T2I-XXL/8++. Together, these results demonstrate that strong T2I generation capabilities can be learned from paired image–text data alone, without language pretraining, while substantially reducing total pretraining compute.

> **Takeaway.** Halcyon demonstrates that T2I models trained entirely from scratch, without internet-scale language pretraining, can outperform strong late-fusion methods while requiring substantially less total pretraining compute.

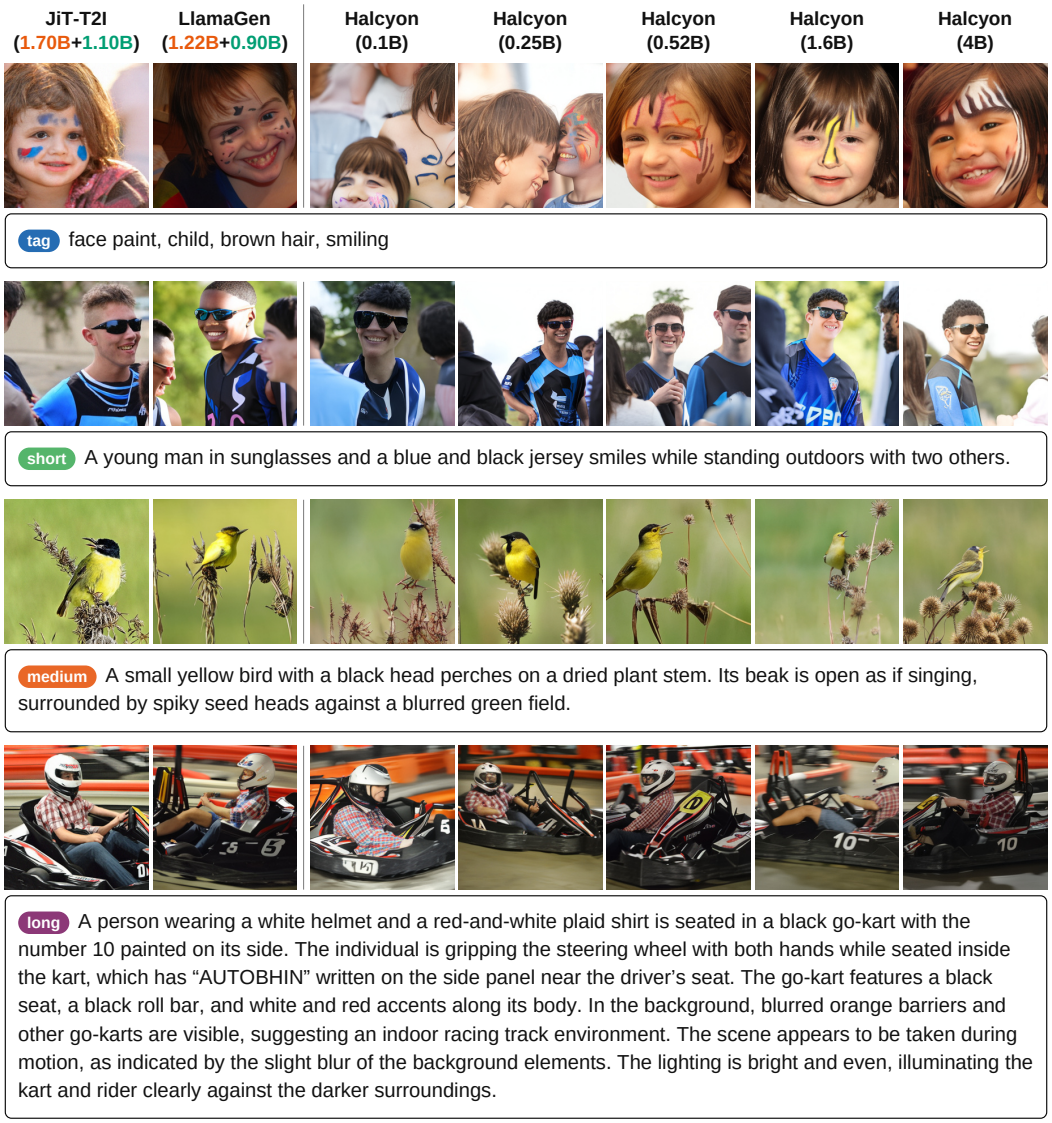


Figure 5: **Comparison of generated images.** From left to right: JiT-T2I-XXL/8++, LlamaGen-XL/8, and Halcyon models from 0.1B to 4B parameters. For the baselines, we report the sizes of the **language model** and **image generator** separately. Image quality improves as Halcyon scales up. All prompts are from GPIC Test-50K. Additional samples are provided in Appendix D.

4.3 RESULTS: EFFECT OF LANGUAGE PRETRAINING ON T2I GENERATION

External language pretraining does not improve T2I generation. We investigate this through two controlled interventions. First, under a fixed pretraining budget, allocating α=$0.1, 0.2, 0.5$ to text-only pretraining reduces paired image–text training and worsens gFD from 48.0 to 51.6, 52.8, and 65.1, respectively; conditional image-token NLL also increases (Table 6). To isolate the effect of language pretraining from reduced T2I training, we additionally evaluate language pretraining followed by a full T2I epoch ($1.5\times$ total compute), but gFD does not improve (48.0 vs. 49.2). Second, we investigate continual pretraining from Qwen3-0.6B-Base, pretrained on 36T text tokens. Despite this extensive language pretraining, Qwen3-0.6B-CPT performs similar to the same architecture trained from random initialization (47.8 vs. 46.6 gFD). Reducing the peak learning rate from 7.5×10^{-4} to 1×10^{-4} for continual pretraining also fails to improve T2I generation (Table 7). Together, these experiments show that external text-only pretraining provides no benefit to T2I generation under the evaluated settings.

Table 6: **Effect of language pretraining on `Halcyon-0.52B`.** Under a fixed pretraining budget, we measure the impact of allocating compute to external text-only pretraining on T2I generation. Each language-pretraining stage exceeds the Chinchilla reference of 20 text tokens per parameter. The † configuration preserves a full T2I epoch after additional language pretraining ($1.5\times$ total pretraining compute), but does not improve gFD. Token counts exclude padding.

		Text tokens (B)		Image tokens (B)		NLL ↓				
Configuration	α	$\mathcal{T} \mid \varnothing$	$\mathcal{T} \mid \mathcal{I}$	$\mathcal{I} \mid \varnothing$	$\mathcal{I} \mid \mathcal{T}$	$\mathcal{T} \mid \varnothing$	$\mathcal{T} \mid \mathcal{I}$	$\mathcal{I} \mid \varnothing$	$\mathcal{I} \mid \mathcal{T}$	gFD ↓
`Halcyon`-0.52B	–	4.05	0	10.24	92.16	**1.433**	**18.414**	**8.428**	**8.366**	**48.02**
Staged: Language → T2I	0.10	16.44	0	9.22	82.94	1.447	19.098	8.442	8.382	51.56
	0.20	28.83	0	8.19	73.73	1.451	17.197	8.445	8.388	52.76
	0.50	65.80	0	5.12	46.08	1.454	22.850	8.505	8.451	65.13
Staged: Language → T2I†	0.33	68.03	0	10.24	92.16	1.445	18.809	**8.428**	8.371	49.22

Table 7: **Continual pretraining from a language model.** Qwen3-0.6B-CPT is initialized from Qwen3-0.6B-Base (pretrained on 36T text tokens), while Qwen3-0.6B is randomly initialized. Both share the same Transformer architecture as `Halcyon`-0.52B but use a larger vocabulary (216k vs. 80,640 for `Halcyon`). We additionally evaluate continual pretraining with a lower peak learning rate (*: 1×10^{-4} vs. 7.5×10^{-4}) to test whether aggressive weight updates diminish the benefits of pretrained language weights. Since vocabularies differ, we report bits per byte (bpb) for both text and images. Neither configuration improves T2I generation over random initialization. All models are trained for one epoch on GPIC and evaluated with CFG 6.0. LP denotes language pretraining tokens. Token counts exclude padding.

		Text tokens (B)		Image tokens (B)		Text bpb ↓		Image bpb ($\times 10^{-2}$) ↓		
Model	LP	$\mathcal{T} \mid \varnothing$	$\mathcal{T} \mid \mathcal{I}$	$\mathcal{I} \mid \varnothing$	$\mathcal{I} \mid \mathcal{T}$	$\mathcal{T} \mid \varnothing$	$\mathcal{T} \mid \mathcal{I}$	$\mathcal{I} \mid \varnothing$	$\mathcal{I} \mid \mathcal{T}$	gFD ↓
`Halcyon`-0.52B	0	4.05	0	10.24	92.16	0.433	5.565	6.333	6.286	48.02
Qwen3-0.6B	0	4.05	0	10.24	92.16	0.434	5.436	**6.324**	**6.278**	**46.61**
Qwen3-0.6B-CPT	36T	4.05	0	10.24	92.16	0.430	**4.974**	6.326	6.283	47.81
Qwen3-0.6B-CPT *	36T	4.05	0	10.24	92.16	**0.424**	6.743	6.409	6.367	63.62

4.4 RESULTS: EFFECT OF I2T SUPERVISION ON T2I GENERATION

I2T supervision as an additional task does not improve T2I generation. We investigate whether adding I2T supervision as a separate task through staged or mixed training improves T2I generation under a fixed pretraining budget (Table 8). At an I2T allocation of 0.1, both schedules achieve gFD comparable to the T2I baseline (48.4 and 48.2 vs. 48.0). Increasing the allocation to 0.5 reduces conditional image tokens by half (92.2B → 46.1B), with staged training outperforming mixing (54.2 vs. 76.0 gFD), although neither improves upon the baseline. Conversely, mixing substantially reduces conditional text NLL compared to staging (1.009 vs. 20.437), indicating better image captioning.

These results suggest that allocating part of the pretraining budget to I2T as a separate task does not benefit T2I generation, despite improving captioning.

Table 8: **Effect of I2T supervision and scheduling on T2I generation for `Halcyon-0.52B`.** Under a fixed pretraining budget, staged and mixed training use identical token counts at each I2T allocation, differing only in scheduling. At 10%, both schedules preserve comparable T2I generation, while at 50%, staged training achieves substantially lower gFD. However, only mixing substantially reduces captioning NLL, namely $\mathrm{NLL}(\mathcal{T} \mid \mathcal{I})$. NLL and gFD are evaluated on GPIC Test-50K; token counts exclude padding.

		Text tokens (B)		Image tokens (B)		NLL ↓				
Configuration	β (or η)	$\mathcal{T} \mid \varnothing$	$\mathcal{T} \mid \mathcal{I}$	$\mathcal{I} \mid \varnothing$	$\mathcal{I} \mid \mathcal{T}$	$\mathcal{T} \mid \varnothing$	$\mathcal{T} \mid \mathcal{I}$	$\mathcal{I} \mid \varnothing$	$\mathcal{I} \mid \mathcal{T}$	gFD ↓
Baseline	–	4.05	0	10.24	92.16	**1.433**	18.414	8.428	8.366	**48.02**
Staged: I2T → T2I	$\beta = 0.1$	3.65	0.45	19.46	82.94	1.440	18.415	8.426	8.366	48.35
	$\beta = 0.5$	2.03	2.25	56.32	46.08	1.435	20.437	8.421	8.370	54.22
Mixing: T2I + I2T	$\eta = 0.1$	3.65	0.45	19.46	82.94	1.443	1.125	**8.414**	**8.362**	48.18
	$\eta = 0.5$	2.03	2.25	56.32	46.08	1.475	**1.009**	8.453	8.417	75.97

Replacing caption dropout with I2T supervision enables image captioning under the same pretraining budget. Rather than allocating a separate training budget to I2T, we replace caption-dropped T2I examples ($p_{\text{drop}} = 0.1$) with I2T examples ($p_{\text{drop}} = 0$, $\eta = 0.1$). Since I2T sequences share the $[\texttt{SOV}]\, x_1 \cdots x_N\, [\texttt{EOV}]$ prefix, this preserves the conditional and unconditional image-token counts under the same pretraining budget. Across all model sizes, conditional text NLL decreases by more than an order of magnitude (e.g., 18.414 → 1.102 at 0.52B), while gFD, GenEval, and DPG-Bench remain comparable to the baseline (Table 9). Caption alignment also increases with model size (0.090 → 0.559), with qualitative examples from `Halcyon-4.00B` shown in Figure 6. Although caption alignment remains below that of GPIC ground truth captions, its consistent improvement with model size demonstrates the potential to learn additional multimodal capabilities while maintaining competitive T2I generation under the same pretraining budget.

Table 9: **Replacing caption dropout with I2T supervision across model sizes.** Replacing caption-dropped T2I examples with I2T preserves the pretraining budget and image-token counts, enabling image captioning while maintaining comparable T2I generation. We report mean Qwen3-VL-Reranker-8B alignment scores over 50K generated image–caption pairs from GPIC Test-50K, using the image as the query and caption as the document. GPIC Test-50k image-caption pairs obtain a mean alignment score of 0.891.

			NLL ↓				T2I generation			Captioning
Size	η	p_{drop}	$\mathcal{T} \mid \varnothing$	$\mathcal{T} \mid \mathcal{I}$	$\mathcal{I} \mid \varnothing$	$\mathcal{I} \mid \mathcal{T}$	gFD ↓	GenEval ↑	DPG-Bench ↑	Alignment ↑
0.10B	–	0.1	1.601	19.066	8.774	8.727	144.16	27.31	66.36	–
	0.1	0	**1.598**	**1.410**	**8.771**	**8.724**	**144.14**	**27.66**	**68.89**	0.090
0.25B	–	0.1	1.497	21.472	8.566	8.509	72.55	35.72	72.66	–
	0.1	0	**1.491**	**1.218**	**8.559**	**8.503**	**70.44**	**36.54**	**73.07**	0.207
0.52B	–	0.1	**1.433**	18.414	8.428	8.366	48.02	**44.91**	**76.84**	–
	0.1	0	1.435	**1.102**	**8.419**	**8.358**	**46.48**	44.60	76.14	0.315
1.6B	–	0.1	**1.370**	20.426	8.303	8.235	**35.28**	53.48	77.90	–
	0.1	0	1.372	**0.977**	**8.292**	**8.228**	35.91	**53.51**	**78.21**	0.467
4B	–	0.1	1.338	16.738	**8.193**	**8.122**	**28.43**	**56.39**	77.90	–
	0.1	0	**1.325**	**0.890**	8.196	8.126	29.30	55.33	**78.61**	0.559

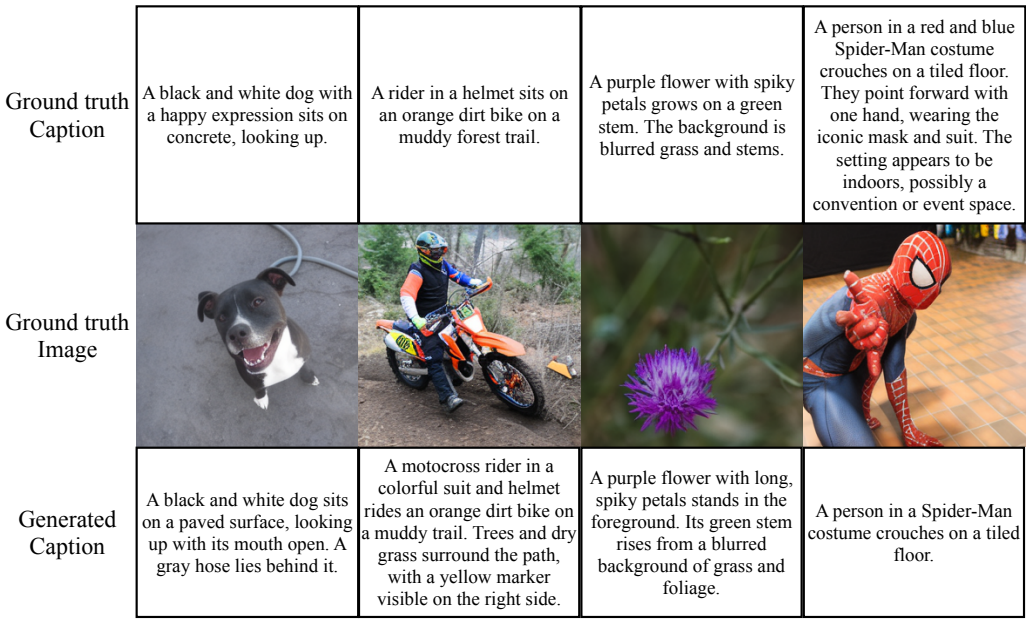


Figure 6: **Image captioning examples from `Halcyon-4B`.** For each example, the ground-truth caption is shown above the input image, and the generated caption is shown below. The model learns image captioning by replacing caption-dropped T2I examples with I2T supervision, using only 10% of the pretraining budget for captioning.

5 Discussion and Conclusion

Model scaling and classifier-free guidance. We examine how `Halcyon` scales from 0.1B to 4B parameters. Figure 7 shows that both image and text validation NLL decrease consistently with model size, with larger models achieving lower NLL throughout pretraining. This demonstrates improvements in modeling both modalities as model capacity increases. We further examine how the optimal CFG scale varies with model size (Figure 8). As model size increases from 0.1B to 4B, the guidance scale minimizing gFD on GPIC Validation-50K decreases from 7.5 to 4.5, while gFD improves from 142.7 to 28.4. This suggests that larger models require less guidance to achieve their best generation quality, although CFG remains beneficial across all evaluated model sizes.

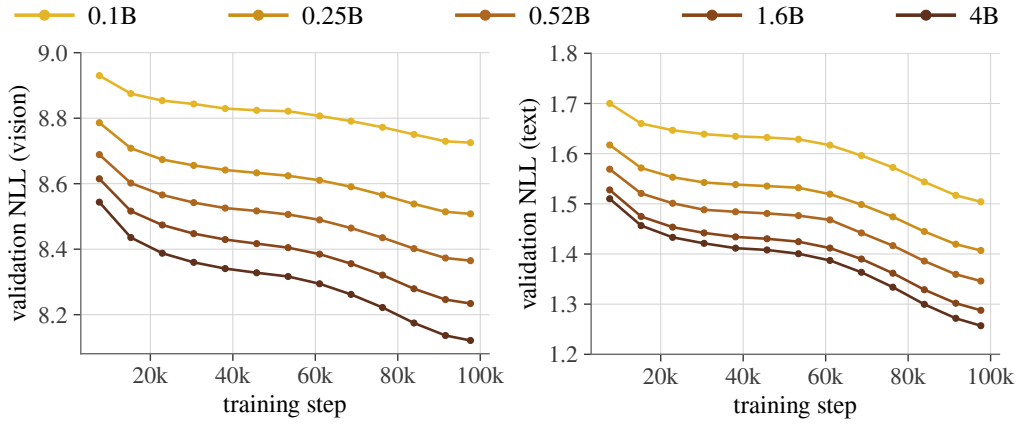


Figure 7: **Scaling behavior of `Halcyon` across model sizes.** Validation negative log-likelihood (NLL) for image tokens (left) and text tokens (right) throughout pretraining on GPIC. As model size increases from 0.1B to 4B, both image and text NLL consistently decrease, with larger models achieving lower NLL throughout training. All models are pretrained from scratch under the same training schedule.

Caption length and text conditioning. Since `Halcyon` learns text conditioning directly from paired image–text data, we investigate how generation quality varies with the length of the caption. We evaluate all five model sizes using short, medium, and long captions from GPIC Test-1M (Table 10). As model size increases from 0.1B to 4B, gFD decreases across all caption lengths. Medium and long captions consistently achieve lower gFD than short captions. These results suggest that more detailed text descriptions improve T2I generation quality, although longer captions do not consistently outperform medium-length captions.

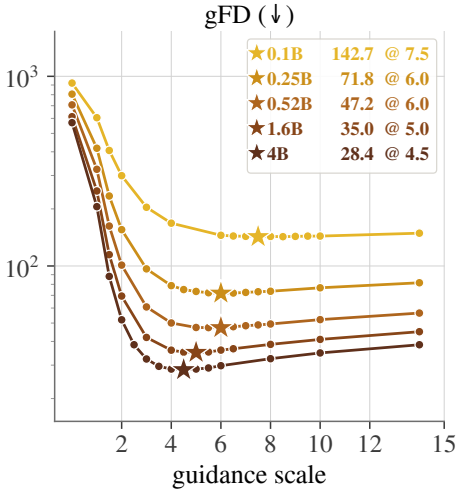


Figure 8: CFG sweep on GPIC validation. The optimal scale drops from 7.5 (0.1B) to 4.5 (4B) as gFD improves from 142.7 to 28.4; these scales are used on Test-50K.

Table 10: **Effect of caption length on T2I generation.** We report gFD (↓) for `Halcyon` across model sizes using 50K examples with short, medium, and long captions from GPIC Test-1M. Test-50K results are included for comparison. Across all model sizes, medium and long captions achieve lower gFD than short captions, with comparable performance between medium and long captions from 0.52B onward. Generation quality consistently improves with model size across all caption lengths.

Model	Test-50K	Short	Medium	Long
0.1B	144.16	149.03	139.27	147.85
0.25B	72.55	77.18	70.14	72.53
0.52B	48.02	51.99	46.12	46.83
1.6B	35.28	38.59	34.21	34.33
4B	28.43	31.48	27.91	27.77

Implications of language pretraining for T2I generation. Our results challenge the prevailing reliance on internet-scale language pretraining for T2I generation. In late-fusion systems, pretrained language models provide text representations, but the image generator must learn to map these representations to images using paired data. On GPIC Test-50K captions, FLAN-T5-XL achieves lower caption NLL than FLAN-T5-Base (2.76 vs. 3.24, averaged over three instruction prompts), with 13.5× more language-pretraining compute (Chung et al., 2024). Yet, using FLAN-T5-XL instead of FLAN-T5-Base only modestly improves LlamaGen gFD (66.7 → 63.3). This result is consistent with Parti (Yu et al., 2022), which reported marginal benefits from text-encoder pretraining. For early-fusion models, text-only pretraining optimizes $p(\mathcal{T} \mid \varnothing)$, whereas T2I generation requires learning $p(\mathcal{I} \mid \mathcal{T})$ with image tokens produced by an independently trained tokenizer. Strong text-only modeling therefore does not guarantee a useful initialization for image generation. Consistent with this view, our randomly initialized Qwen3 model achieves lower vision validation NLL throughout pretraining (Appendix Figure 10) and lower final gFD than its language-pretrained counterpart. In contrast, `Halcyon` learns text representations for image generation jointly from paired image–text data without language pretraining, although our experiments do not isolate the effect of joint training. Since our study focuses on pretraining, we account for all pretraining FLOPs required to build each T2I system, including language-model, image-tokenizer, and image-generator pretraining. We note that our study examines the role of language pretraining in late- and early-fusion systems, not the relative capabilities of autoregressive and diffusion-based image generators. Beyond T2I, our findings motivate reconsidering the role of modality-specific pretraining in other multimodal settings, including biological data (Radical Numerics, 2026; Neiswanger et al., 2026; Lee et al., 2026).

Limitations and future directions. While our experiments use GPIC, which is, to our knowledge, the largest high-quality, centrally hosted, publicly available dataset and benchmark for visual generation, extending these findings to other datasets and higher resolutions remains important. Our controlled data composition and scheduling experiments use `Halcyon`-0.52B, while the pretrained initialization comparison uses Qwen3-0.6B; extensions to larger model sizes and pretraining budgets would help establish the generality of our findings. Although our late-fusion baselines include

diffusion-based generators, our controlled interventions focus on autoregressive models. Extending these experiments to Transfusion-style models (Zhou et al., 2025) is a valuable direction. Finally, text-to-video systems such as Wan (Wan et al., 2025), CogVideoX (Yang et al., 2025b), and Mochi (Team, 2024) also rely on large pretrained language encoders for video generation. Investigating whether T2V models can learn text conditioning directly from paired video–text data, without language pretraining, is a promising direction toward efficient video generation systems.

Conclusion. In this work, we study whether internet-scale language pretraining is necessary for strong T2I generation. We introduced `Halcyon` a family of early-fusion multimodal autoregressive models trained entirely from scratch on paired image–text data (GPIC). Our models outperform strong late-fusion baselines while using significantly less total pretraining compute. Our controlled experiments show that external text-only pretraining does not improve T2I generation under the evaluated settings. Furthermore, replacing caption-dropped T2I examples with I2T supervision enables image captioning while maintaining competitive generation quality under the same pretraining budget. Our results show that strong T2I systems can be trained from scratch without internet-scale language pretraining. Code and models are available at `cs.stanford.edu/~keshik/halcyon`.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing methods. We did not use generative AI tools to propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative or thematic data analysis, or interpret results. Generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, assisting with translation, and cleaning or reformatting datasets were not applicable to our work. Additionally, we used generative AI tools to create or edit software code, search for and identify relevant literature and information, draft and edit parts of the manuscript, and improve readability. We reviewed all AI-assisted work. AI-generated or edited code was inspected and tested for correctness, and information and references suggested by AI tools were verified against the original sources. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work studies text-to-image generative modeling, which can be used to generate realistic visual content and may therefore be misused. By providing a significantly lower-cost recipe for training strong T2I models, our work lowers the barrier to developing capable generative systems, including for malicious use. `Halcyon` and the baseline models are pretrained on GPIC and may inherit biases or limitations present in its images and captions. Our work does not involve human-subject studies or the collection of a new dataset. We do not identify additional ethical risks specific to our method beyond those associated with visual generative models.

REPRODUCIBILITY STATEMENT

We provide the model architecture, tokenization, training objective, data composition and scheduling, baseline modifications, evaluation protocol, and compute accounting in the main paper, with full architecture, optimization, and sampling details in the Appendix. The GPIC dataset and benchmark are publicly available. Our model checkpoints are publicly available.

ACKNOWLEDGMENTS

We thank Radical Numerics for providing compute for this project. We thank Willie Neiswanger, Garyk Brixi, Tiange Xiang, Wenlong Huang, Mark Endo, Federico Berto, Zane Durante, Bailey Trang, Yanchen Wang, and Suzannah Wistreich for feedback on the manuscript. We also thank Suchir Agarwal for valuable discussions, the broader Radical Numerics research team for their feedback, and Alessandro Moro for infrastructure support.

REFERENCES

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

Black Forest Labs. FLUX.2. `https://github.com/black-forest-labs/flux2`, 2025.

Keshigeyan Chandrasegaran, Kyle Sargent, Suchir Agarwal, Michael Jang, Michael Poli, Juan Carlos Niebles, Justin Johnson, Jiajun Wu, and Li Fei-Fei. Gpic: A giant permissive image corpus for visual generation. *arXiv preprint arXiv:2605.30341*, 2026.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *JMLR*, 25(70):1–53, 2024.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, et al. Nemotron-climb: Clustering-based iterative data mixture bootstrapping for language model pre-training. *NeurIPS*, 2025.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *NeurIPS*, 2021.

Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. In *ICLR*, 2025.

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

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

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

Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *CoLM*, 2024a.

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*, 2024b.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Joseph Lee, Yanjing Li, Bo Qiang, Gregory Lau, Allyson Ettinger, Noah A Smith, Rory Donovan-Maiye, David Baker, Hanna Hajishirzi, and Pang Wei Koh. Designing protein structures from natural language. *arXiv preprint arXiv:2610.13917*, 2026.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *CVPR*, 2026.

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.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Improving pixel diffusion with perceptual supervision. *arXiv preprint arXiv:2602.02493*, 2026.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In *ICLR*, 2024.

Willie Neiswanger, Matthew Carter, Anvith Thudi, and Michael Poli. Scaling data mixing for language and dna pretraining, Oct 2026. URL https://www.radicalnumerics.ai/blog/tessera-data-mixing.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

Radical Numerics. A new frontier in generative genomics with omnii, Jun 2026. URL https://www.radicalnumerics.ai/blog/omnii-health-preview.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, 2016.

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.

Chameleon Team, Mingda Chen, Jacob Kahn, and Shang-Wen Li. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024a.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024b.

Genmo Team. Mochi 1. https://github.com/genmoai/models, 2024.

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

Junke Wang, Zhi Tian, Xun Wang, Xinyu Zhang, Weilin Huang, Zuxuan Wu, and Yu-Gang Jiang. SimpleAR: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. *arXiv preprint arXiv:2504.11455*, 2025.

Xinlong Wang, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Zhen Li, Yuqi Wang, et al. Multimodal learning with next-token prediction for large multimodal models. *Nature*, 650(8101):327–333, 2026.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformer. In *ICLR*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025b.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *TMLR*, 2022. URL https://openreview.net/forum?id=AFDcYJKhND.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Yuhui Zhang, Brandon McKinzie, Zhe Gan, Vaishaal Shankar, and Alexander T Toshev. Pre-trained language models do not help auto-regressive text-to-image generation. In *EMNLP*, 2024.

Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, et al. Qwen-image-2.0 technical report. *arXiv preprint arXiv:2605.10730*, 2026.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *ICLR*, 2025.

APPENDIX

A Training Details **17**

B Sampling Details **18**

C Additional Results **18**

C.1 Per-Category Results on GenEval and DPG-Bench 18

C.2 Standard Deviation (Training Seeds) 19

C.3 Standard Deviation (Sampling Seeds) 19

C.4 Inference Cost 19

D Samples **20**

A TRAINING DETAILS

We provide model architecture and training details for all `Halcyon` models in Table 11.

Table 11: **`Halcyon` configurations.**

	0.1B	**0.25B**	**0.52B**	**1.6B**	**4B**
architecture					
backbone	Qwen3 (Yang et al., 2025a), decoder-only				
layers	14	21	28	28	36
hidden dim	512	768	1024	2048	2560
FFN dim	1536	2304	3072	6144	9728
query / KV heads	8 / 4	12 / 6	16 / 8	16 / 8	32 / 8
head dim	128				
non-embedding params	55.1M	185.8M	440.4M	1409.3M	3634M
embedding params	41.3M	61.9M	82.6M	165.2M	206.4M
total params	96.4M	247.7M	523.0M	1574.5M	3840M
vocabulary	80,640 (16,384 text + 64,000 image + 256 special)				
tied embeddings	yes				
image size	256				
image tokens	1,024				
max caption tokens	254				
sequence length	1,280				
training					
training steps	97,632 (1 epoch on GPIC)				
global batch size	1,024				
warmup steps	1,000				
peak learning rate	1e-3	7.5e-4	7.5e-4	7.5e-4	7.5e-4
lr schedule	linear warmup, constant to step 49,595, cooldown to peak$/10$				
optimizer	AdamW, $\beta_1, \beta_2 = 0.9, 0.95$				
weight decay	0.1				
gradient clipping	1.0				
init std	0.02				
image loss weight λ_{img}	0.5				
caption dropout p_{drop}	0.1				
precision	bf16				
sampling					
cfg	7.5	6.0	6.0	5.0	4.5

Table 12: **JiT-T2I configurations.**

	JiT-T2I-XXL/16	JiT-T2I-XXL/16++	JiT-T2I-XXL/8++
language model (frozen)			
text encoder		Qwen3-1.7B	
feature dim		2048	
max caption tokens		300	
image generator			
backbone		JiT-T2I (pixel space, no image tokenizer)	
blocks (image / text)		16 / 4	
hidden dim		1536	
attention heads		24	
patch bottleneck dim		256	
params		1.10B	
patch size	16	16	8
image tokens	256	256	1,024
training			
objective		flow matching, linear schedule	
timestep sampling		logit-normal (μ=-0.8, σ=0.8), t_ϵ=0.05	
LPIPS weight	–	0.1	0.1
LPIPS timestep gate	–	$t \geq 0.3$	$t \geq 0.3$
training steps		390,600	
global batch size		256	
peak learning rate		1e-4	
lr schedule		4,000 warmup, constant, 4,000 cooldown	
optimizer		AdamW, $\beta_1, \beta_2 = 0.9, 0.95$	
weight decay		0	
gradient clipping		1.0	
EMA decay		0.9999	
caption dropout		0.1	
precision		bf16	
sampling			
sampler		Euler (ODE), 50 steps, EMA weights	
CFG scale	6.00	5.00	4.00

B Sampling Details

All models generate one 256×256 image per caption using the GPIC Test-50K set. For `Halcyon`, we sample 1,024 image tokens autoregressively after the caption prefix and `[SOV]`, using temperature 1.0 with top-k and top-p disabled and a fixed seed. CFG is applied only to image-token logits, $\ell_{\text{cfg}} = \ell_u + \gamma(\ell_c - \ell_u)$, where the unconditional branch uses the `[SOV]` prefix without the caption. For each model size, we select γ to minimize gFD on the GPIC validation set (Figure 8).
Baselines. For LlamaGen and JiT-T2I, we use the official sampling code and settings and identify the optimal CFG setting using the GPIC validation set.

C Additional Results

C.1 Per-Category Results on GenEval and DPG-Bench

Tabs. 18 and 19 report per-category GenEval and DPG-Bench results for all models in Tab. 5. GenEval overall scores are reported as mean $\pm$ std. We additionally report `Halcyon` trained with I2T supervision (p_{drop}=0.0, η=0.1) at every scale. On GenEval, I2T supervision matches the standard recipe at all model sizes, with differences in overall scores within one standard deviation. On DPG-Bench, it yields slightly higher overall scores for four of the five model sizes.

Table 13: **LlamaGen configurations.**

	LlamaGen-XL	LlamaGen-XL/8	LlamaGen-XL-T5-Base/8
language model (frozen)			
text encoder	FLAN-T5-XL	FLAN-T5-XL	FLAN-T5-Base
encoder params	1.22B	1.22B	0.11B
feature dim	2048	2048	768
max caption tokens		300	
image tokenizer			
tokenizer	VQ-16 (pretrained)	Ours (GPIC-trained)	
spatial downsampling	16×	8×	
codebook size	16,384	64,000	
image tokens	256	1,024	
image generator			
backbone	GPT-XL (Llama-style, decoder-only)		
layers / hidden dim / heads	36 / 1280 / 20		
FFN dim	3584 (SwiGLU)		
params	0.78B	0.90B	0.90B
sequence length	556	1,324	1,324
training			
training steps	390,600	390,600	390,600
global batch size	256		
peak learning rate	1e-4		
lr schedule	3,906 warmup, then constant		
optimizer	AdamW, $\beta_1, \beta_2 = 0.9, 0.95$		
weight decay	0.05		
gradient clipping	1.0		
dropout / token dropout	0.1 / 0.1		
caption dropout	0.1		
precision	bf16		
sampling			
CFG scale	6.5	6.5	5.5
temperature / top-k / top-p	1.0 / 1000 / off	1.0 / off / off	

C.2 Standard Deviation (Training Seeds)

To estimate variance due to pretraining, we train `Halcyon`-0.10B with I2T supervision (p_{drop}=0.0, η=0.1) three times with different training seeds, keeping all other settings fixed. We evaluate each run with CFG 7.5. Tab. 20 reports per-seed results along with the mean $\pm$ std. The std of gFD across seeds is 5.54, which gives a reference scale for interpreting gFD differences between small models. The sample-quality metrics (precision, recall, density, coverage) remain stable across seeds.

C.3 Standard Deviation (Sampling Seeds)

To measure how much evaluation noise comes from sampling, we generate the 50k evaluation images for each `Halcyon` model three times with different sampling seeds, keeping the checkpoint and CFG scale fixed. Tab. 21 shows that sampling variance is small at every scale: the std of gFD is at most 0.14.

C.4 Inference Cost

We measure inference cost on the GPIC 50k test set. For each model, we generate all 50k images on a single H100 node of 8 GPUs with vLLM (Kwon et al., 2023), splitting prompts evenly across GPUs. We report wall-clock time averaged over 3 runs. Sampling settings match our evaluation settings. We report results in Table 17.

Table 14: **Tokenizer training configurations.**

Image tokenizer	
architecture	Cosmos (Agarwal et al., 2025) encoder–decoder, FSQ (Mentzer et al., 2024)
decoder width	192 (128 in Cosmos)
spatial downsampling	$8\times$
codebook size	64,000 (FSQ levels $[8, 8, 8, 5, 5, 5]$)
Params	136M
tokens per image	1,024 (32×32, raster order)
losses (stage 1)	reconstruction (1.0) + LPIPS (0.1)
losses (stage 2)	+ PatchGAN adversarial (5e-3)
training data	GPIC train images, 1.5 epochs (1.0 stage 1 / 0.5 stage 2)
iterations	242,172
batch size	640
learning rate	1e-4, constant
optimizer	Adam, $\beta_1, \beta_2 = 0.9, 0.999$
weight decay	0.001
gradient clipping	1.0
EMA	yes
hardware	$1\times$ 8×H100 node
Text tokenizer	
algorithm	byte-level BPE (Sennrich et al., 2016)
vocabulary size	16,384
training data	GPIC train captions (21.3 GB) + OpenWebText (11.9 GB)
caption length	mean 44.5, p50 36, p90 64, p99 182 tokens

Table 15: **Total pretraining compute of T2I systems.** We report total pretraining FLOPs, decomposed into language, image-generator, and image-tokenizer compute, for all configurations shown in Fig. 1. Vision compute includes both image-generator and image-tokenizer pretraining. For late-fusion methods, language FLOPs include pretraining of the language model. For `Halcyon`, language FLOPs correspond to the text-token portion of joint multimodal pretraining, without external language pretraining. All image generators are pretrained on GPIC for one epoch at 256×256 resolution. Generation Fréchet distance (gFD) is evaluated on GPIC Test-50K; lower is better.

		Language		Vision			
Method	Total FLOPs	FLOPs	(%)	Generator FLOPs	Tokenizer FLOPs	(%)	gFD ↓
LlamaGen-XL	9.32×10^{21}	9.06×10^{21}	97.17	1.19×10^{20}	1.45×10^{20}	2.83	147.01
LlamaGen-XL/8	9.68×10^{21}	9.06×10^{21}	93.58	4.76×10^{20}	1.45×10^{20}	6.42	63.29
LlamaGen-XL-T5-Base/8	1.29×10^{21}	6.69×10^{20}	51.87	4.76×10^{20}	1.45×10^{20}	48.13	66.69
JiT-T2I-XXL/16	3.67×10^{23}	3.67×10^{23}	99.95	1.69×10^{20}	0	0.05	67.92
JiT-T2I-XXL/16++	3.67×10^{23}	3.67×10^{23}	99.95	1.69×10^{20}	0	0.05	48.86
JiT-T2I-XXL/8++	3.68×10^{23}	3.67×10^{23}	99.82	6.76×10^{20}	0	0.18	40.97
`Halcyon`-0.1B	2.05×10^{20}	1.47×10^{19}	7.15	5.93×10^{19}	1.31×10^{20}	92.85	144.16
`Halcyon`-0.25B	3.21×10^{20}	3.78×10^{19}	11.74	1.53×10^{20}	1.31×10^{20}	88.26	72.55
`Halcyon`-0.52B	5.33×10^{20}	7.97×10^{19}	14.96	3.22×10^{20}	1.31×10^{20}	85.04	48.02
`Halcyon`-1.6B	1.34×10^{21}	2.40×10^{20}	17.90	9.69×10^{20}	1.31×10^{20}	82.10	35.28
`Halcyon`-4B	3.08×10^{21}	5.85×10^{20}	19.00	2.36×10^{21}	1.31×10^{20}	81.00	28.43

D SAMPLES

We compare generated samples across the four GPIC caption types: tag (Figure 11), short (Figure 12), medium (Figure 13), and long (Figure 14). Each row shows JiT-T2I-XXL/8++, LlamaGen-XL/8, and `Halcyon` from 0.1B to 4B for the same caption, sampled with the settings in Appendix B. Sample quality and prompt adherence improve as `Halcyon` scales, with larger models capturing finer details such as clothing, object attributes, and scene layout. Rendering legible text remains challenging for all models at 256×256 resolution.

Table 16: **Classifier-free guidance (CFG) scales for T2I evaluation.** We report the CFG scale selected for each model based on the lowest gFD on GPIC Validation-50K. Sweeps are shown in Figures 8 and 9.

Model	p_{drop}	η	CFG Scale
LlamaGen-XL	0.1	–	6.5
LlamaGen-XL/8	0.1	–	6.5
LlamaGen-XL-T5-Base/8	0.1	–	5.5
JiT-T2I-XXL/16	0.1	–	6.0
JiT-T2I-XXL/16 ++	0.1	–	5.0
JiT-T2I-XXL/8 ++	0.1	–	4.0
`Halcyon`-0.10B	0.1	–	7.5
	0.0	0.1	7.5
`Halcyon`-0.25B	0.1	–	6.0
	0.0	0.1	6.0
`Halcyon`-0.52B	0.1	–	6.0
	0.0	0.1	6.0
`Halcyon`-1.6B	0.1	–	5.0
	0.0	0.1	5.2
`Halcyon`-4B	0.1	–	4.5
	0.0	0.1	4.6

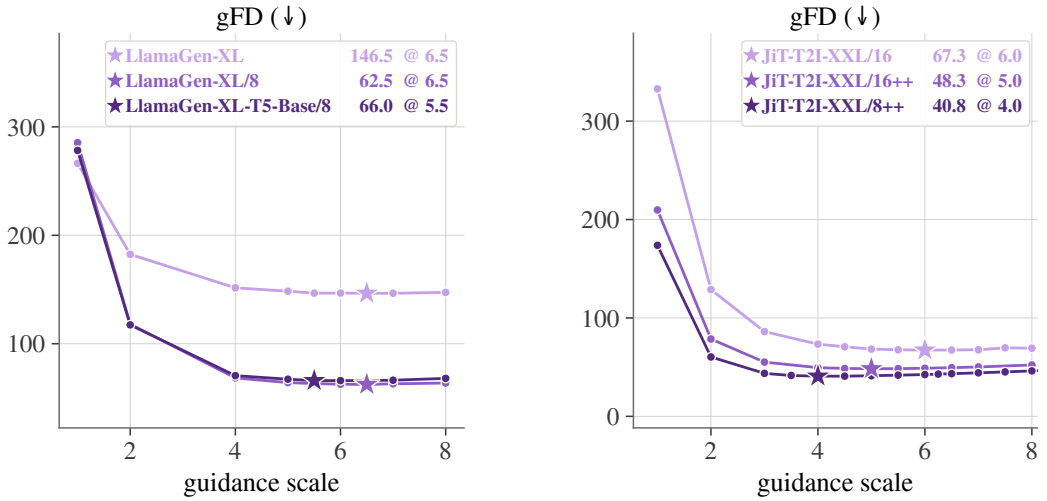


Figure 9: **Classifier-free guidance (CFG) sweeps for late-fusion T2I baselines.** We evaluate LlamaGen variants (left) and JiT-T2I variants (right) across CFG scales on GPIC Validation-50K. Stars indicate the scale achieving the lowest gFD for each configuration, with the corresponding gFD and CFG scale reported in the figure. The selected CFG scales are used for all subsequent evaluations on GPIC Test-50K, GenEval, and DPG-Bench.

Table 17: **Inference cost of Halcyon.** Wall-clock time and throughput for generating 50K images (GPIC Test-50K set) using 8xH100 GPUs with vLLM, averaged over three runs. We use temperature 1.0 without top-k/top-p filtering and each model's CFG scale (two forward passes per image token). Image decoding takes 198.0 ± 5.7 seconds (averaged over 3 runs) for all model sizes and is excluded from the reported results.

Params	Wall time (s) ↓	Throughput (img/s) ↑
0.1B	395.3	126.5
0.25B	507.4	98.5
0.52B	662.8	75.4
1.6B	712.2	70.2
4B	935.5	53.4

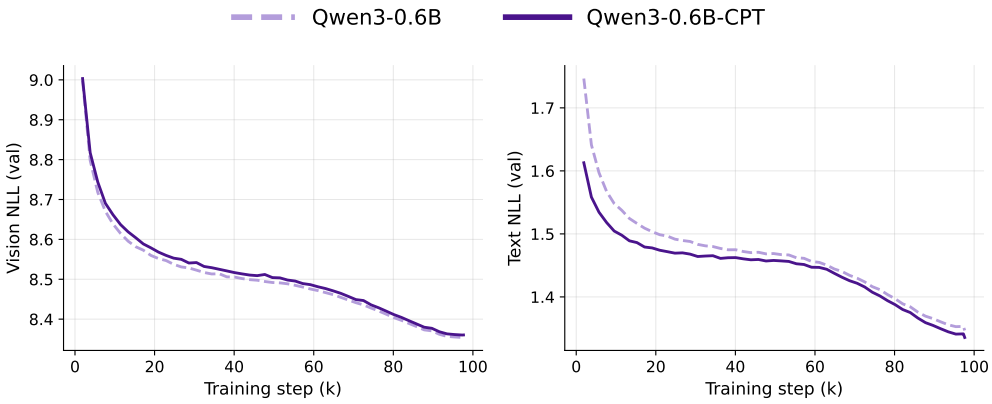


Figure 10: **Effect of language pretraining on validation NLL.** Validation NLL for image tokens conditioned on text ($\mathcal{I} \mid \mathcal{T}$, left) and text tokens without conditioning ($\mathcal{T} \mid \varnothing$, right) during training of Qwen3-0.6B with random vs. language-pretrained initialization. Random initialization consistently achieves lower image NLL, while language-pretrained initialization achieves lower text NLL.

Table 18: **GenEval results by category.** Overall is reported as mean ± std. "+ I2T 10%" denotes the I2T supervision recipe (p_{drop}=0.0, η=0.1).

Model	Overall ↑	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attr.
LlamaGen-XL	30.03 ±1.10	78.44	19.95	20.31	53.46	4.25	3.75
LlamaGen-XL/8	30.48 ±1.74	75.94	23.74	15.94	53.99	9.50	3.75
LlamaGen-XL-T5-Base/8	28.48 ±1.19	78.12	18.18	15.94	48.40	6.75	3.50
JiT-T2I-XXL/16	36.27 ±0.93	75.62	32.32	22.50	56.65	18.50	12.00
JiT-T2I-XXL/16 ++	41.79 ±2.48	85.00	41.41	21.88	61.97	23.25	17.25
JiT-T2I-XXL/8 ++	43.17 ±1.29	83.75	44.95	25.94	60.11	27.25	17.00
`Halcyon`-0.10B	27.31 ±0.61	70.94	26.52	8.12	46.54	7.25	4.50
+ I2T 10%	27.66 ±2.32	76.88	20.71	9.06	47.07	6.50	5.75
`Halcyon`-0.25B	35.72 ±1.68	78.44	35.35	18.75	55.05	13.50	13.25
+ I2T 10%	36.54 ±2.02	78.12	39.39	15.31	56.91	16.25	13.25
`Halcyon`-0.52B	44.91 ±2.29	83.75	50.00	26.56	61.17	29.75	18.25
+ I2T 10%	44.60 ±1.25	81.88	51.52	22.81	60.64	29.50	21.25
`Halcyon`-1.60B	53.48 ±1.01	88.75	67.68	28.12	63.56	42.00	30.75
+ I2T 10%	53.51 ±3.01	85.94	63.13	28.75	66.76	47.75	28.75
`Halcyon`-4.00B	56.39 ±1.33	92.81	68.18	29.06	67.02	51.00	30.25
+ I2T 10%	55.33 ±0.55	87.81	62.88	28.75	71.28	44.00	37.25

Table 19: **DPG-Bench results by category.** "+ I2T 10%" denotes the I2T supervision recipe (p_{drop}=0.0, η=0.1).

Model	Overall ↑	Global	Entity	Attribute	Relation	Other
LlamaGen-XL	65.40	85.85	80.48	79.19	79.92	77.97
LlamaGen-XL/8	67.08	79.09	80.53	79.24	86.21	79.30
LlamaGen-XL-T5-Base/8	64.67	77.34	78.71	80.15	80.88	75.41
JiT-T2I-XXL/16	72.69	81.86	83.69	85.15	88.36	83.28
JiT-T2I-XXL/16 ++	76.58	84.88	87.39	86.46	86.70	87.65
JiT-T2I-XXL/8 ++	76.85	86.14	84.18	84.97	90.21	89.93
Halcyon-0.10B	66.36	78.33	79.02	82.02	81.20	79.56
+ I2T 10%	68.89	75.37	79.85	83.88	82.36	82.06
Halcyon-0.25B	72.66	82.83	83.96	84.75	84.39	84.37
+ I2T 10%	73.07	81.75	81.59	86.03	85.76	85.58
Halcyon-0.52B	76.84	77.51	85.84	86.51	88.72	85.33
+ I2T 10%	76.14	83.59	85.61	86.62	86.35	87.14
Halcyon-1.60B	77.90	85.80	86.83	86.62	87.86	87.19
+ I2T 10%	78.21	91.66	84.76	87.98	86.36	84.36
Halcyon-4.00B	77.90	85.63	83.94	86.03	89.10	87.09
+ I2T 10%	78.61	89.02	86.98	86.20	90.16	84.50

Table 20: **Standard deviation of training runs.** Halcyon-0.10B with 10% I2T supervision, pre-trained using three random seeds and evaluated with CFG 7.5. The last row reports mean ± standard deviation (sample std, n=3).

Seed	gFD ↓	Precision ↑	Recall ↑	Density → 1	Coverage ↑	MMD ↓
1234	144.14	0.926	0.631	1.057	0.841	0.132
2026	154.63	0.926	0.606	1.070	0.848	0.146
7777	152.50	0.927	0.616	1.031	0.850	0.143
Mean ± std	150.42 ±5.54	0.926 ±0.001	0.618 ±0.013	1.053 ±0.020	0.846 ±0.005	0.140 ±0.007

Table 21: **Standard deviation of sampling runs.** Each Halcyon model is evaluated using a fixed checkpoint and three random sampling seeds (50K images per seed). We report mean ± standard deviation (n=3).

Model	CFG	Seed	gFD ↓	Precision ↑	Recall ↑	Density → 1	Coverage ↑	MMD ↓
Halcyon-0.10B	7.5	1234	144.16	0.924	0.636	1.029	0.853	0.131
		2026	143.91	0.927	0.638	1.032	0.863	0.130
		7777	144.13	0.925	0.644	1.042	0.853	0.131
		Avg. ± std	144.07 ±0.14	0.925 ±0.002	0.639 ±0.004	1.034 ±0.007	0.856 ±0.006	0.131 ±0.001
Halcyon-0.25B	6.0	1234	72.55	0.946	0.796	1.050	0.922	0.050
		2026	72.35	0.947	0.804	1.053	0.922	0.049
		7777	72.29	0.949	0.808	1.065	0.923	0.050
		Avg. ± std	72.40 ±0.14	0.947 ±0.002	0.803 ±0.006	1.056 ±0.008	0.922 ±0.001	0.050 ±0.001
Halcyon-0.52B	6.0	1234	48.02	0.953	0.858	1.064	0.938	0.027
		2026	48.03	0.954	0.861	1.067	0.944	0.028
		7777	47.90	0.953	0.859	1.073	0.944	0.028
		Avg. ± std	47.98 ±0.08	0.953 ±0.001	0.859 ±0.002	1.068 ±0.005	0.942 ±0.003	0.028 ±0.001
Halcyon-1.60B	5.0	1234	35.28	0.955	0.891	1.071	0.955	0.019
		2026	35.46	0.954	0.891	1.050	0.952	0.019
		7777	35.46	0.953	0.891	1.057	0.951	0.019
		Avg. ± std	35.40 ±0.11	0.954 ±0.001	0.891 ±0.000	1.059 ±0.011	0.953 ±0.002	0.019 ±0.000
Halcyon-4.00B	4.5	1234	28.43	0.951	0.905	1.034	0.960	0.014
		2026	28.56	0.955	0.910	1.038	0.960	0.014
		7777	28.63	0.956	0.905	1.058	0.966	0.014
		Avg. ± std	28.54 ±0.10	0.954 ±0.003	0.907 ±0.003	1.043 ±0.013	0.962 ±0.003	0.014 ±0.000

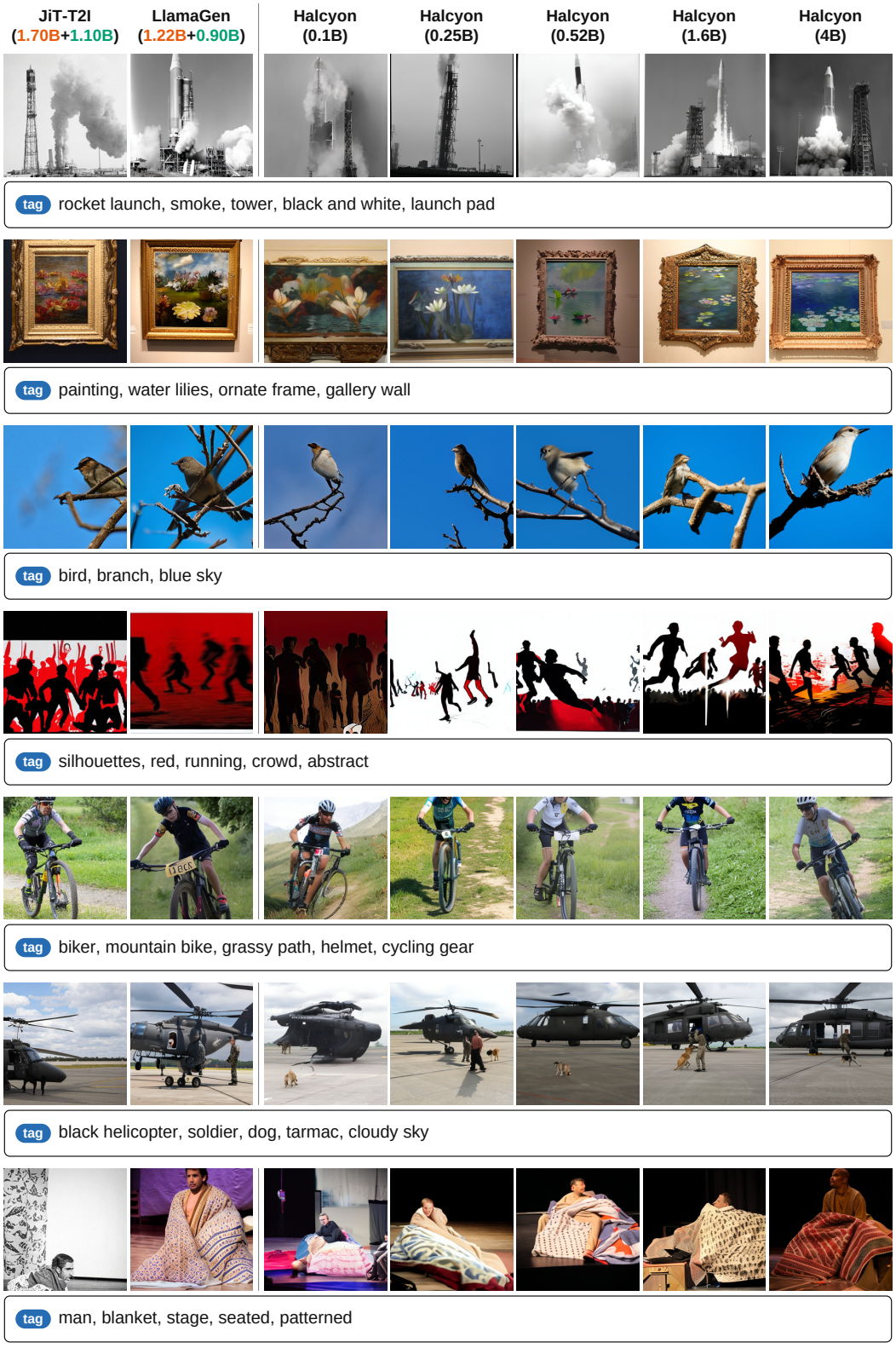


Figure 11: **Comparison of generated images with "tag" captions.** From left to right: JiT-T2I-XXL/8++, LlamaGen-XL/8, and `Halcyon` from 0.1B to 4B. For the baselines, we report the sizes of the **language model** and **image generator** separately. Image quality improves as `Halcyon` scales up. All prompts are from GPIC Test-50K.

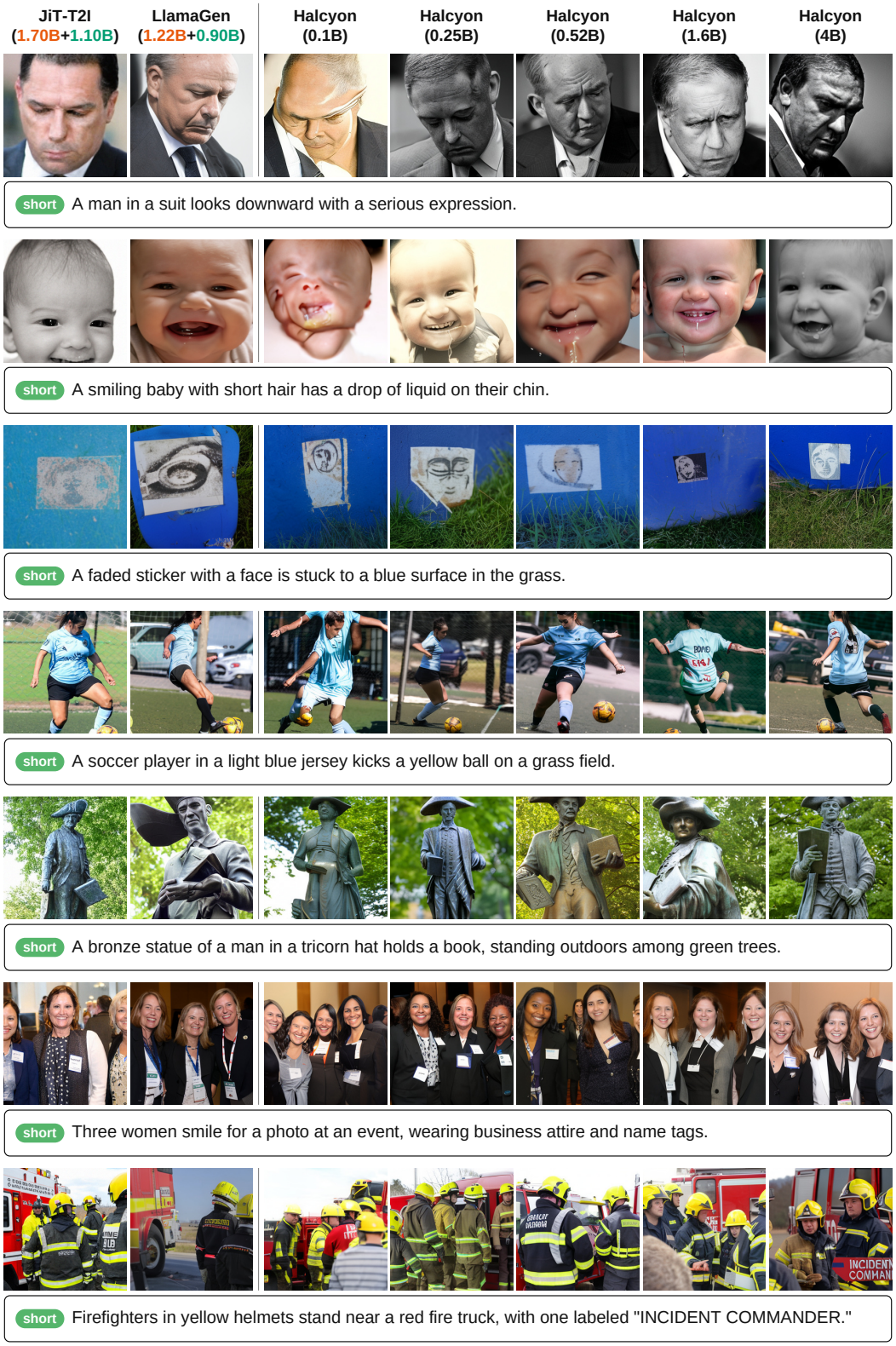


Figure 12: **Comparison of generated images with "short" captions.** From left to right: JiT-T2I-XXL/8++, LlamaGen-XL/8, and `Halcyon` from 0.1B to 4B. For the baselines, we report the sizes of the **language model** and **image generator** separately. Image quality improves as `Halcyon` scales up. All prompts are from GPIC Test-50K.

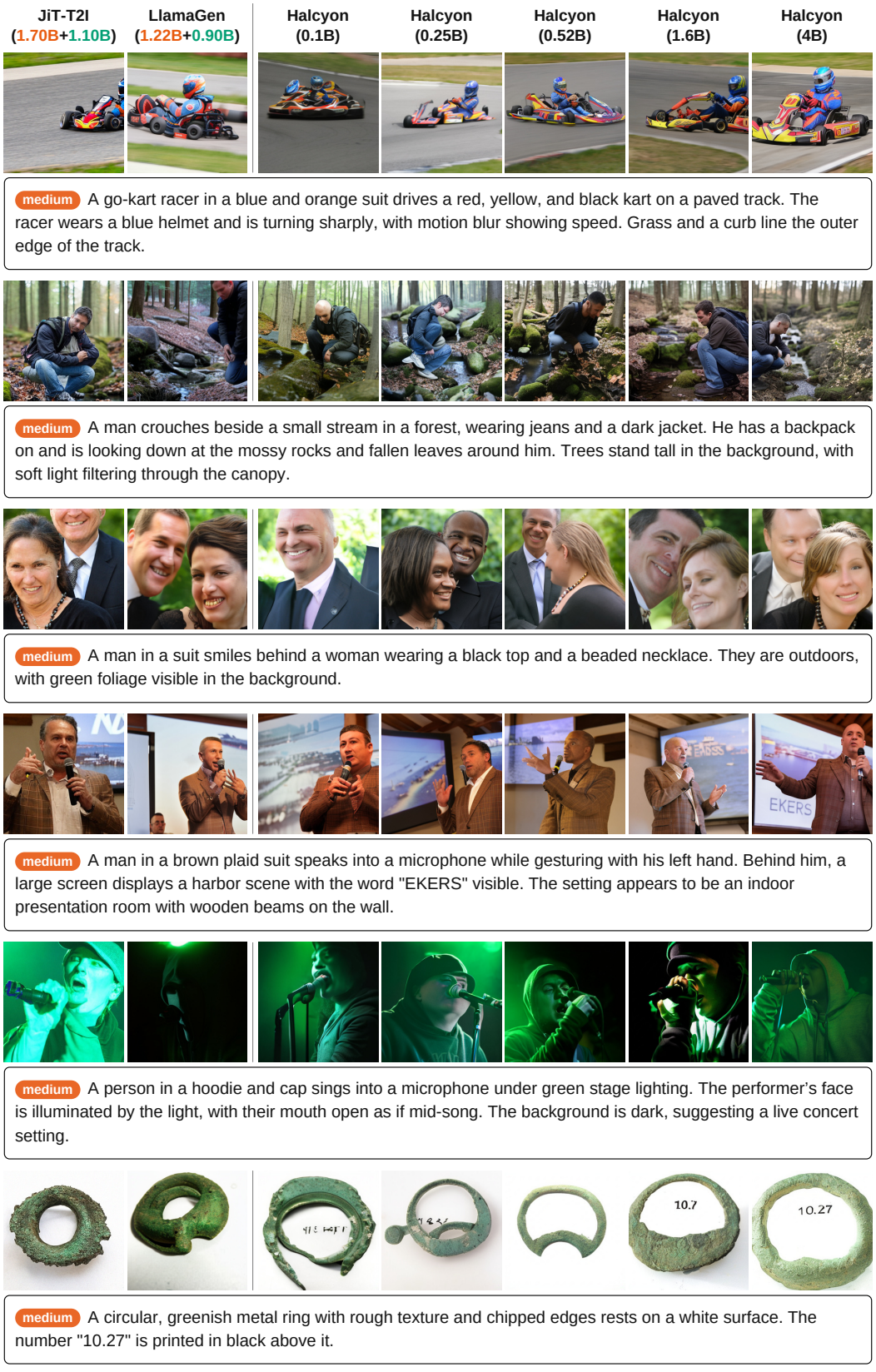


Figure 13: **Comparison of generated images with "medium" captions.** From left to right: JiT-T2I-XXL/8++, LlamaGen-XL/8, and `Halcyon` from 0.1B to 4B. For the baselines, we report the sizes of the **language model** and **image generator** separately. Image quality improves as `Halcyon` scales up. All prompts are from GPIC Test-50K.

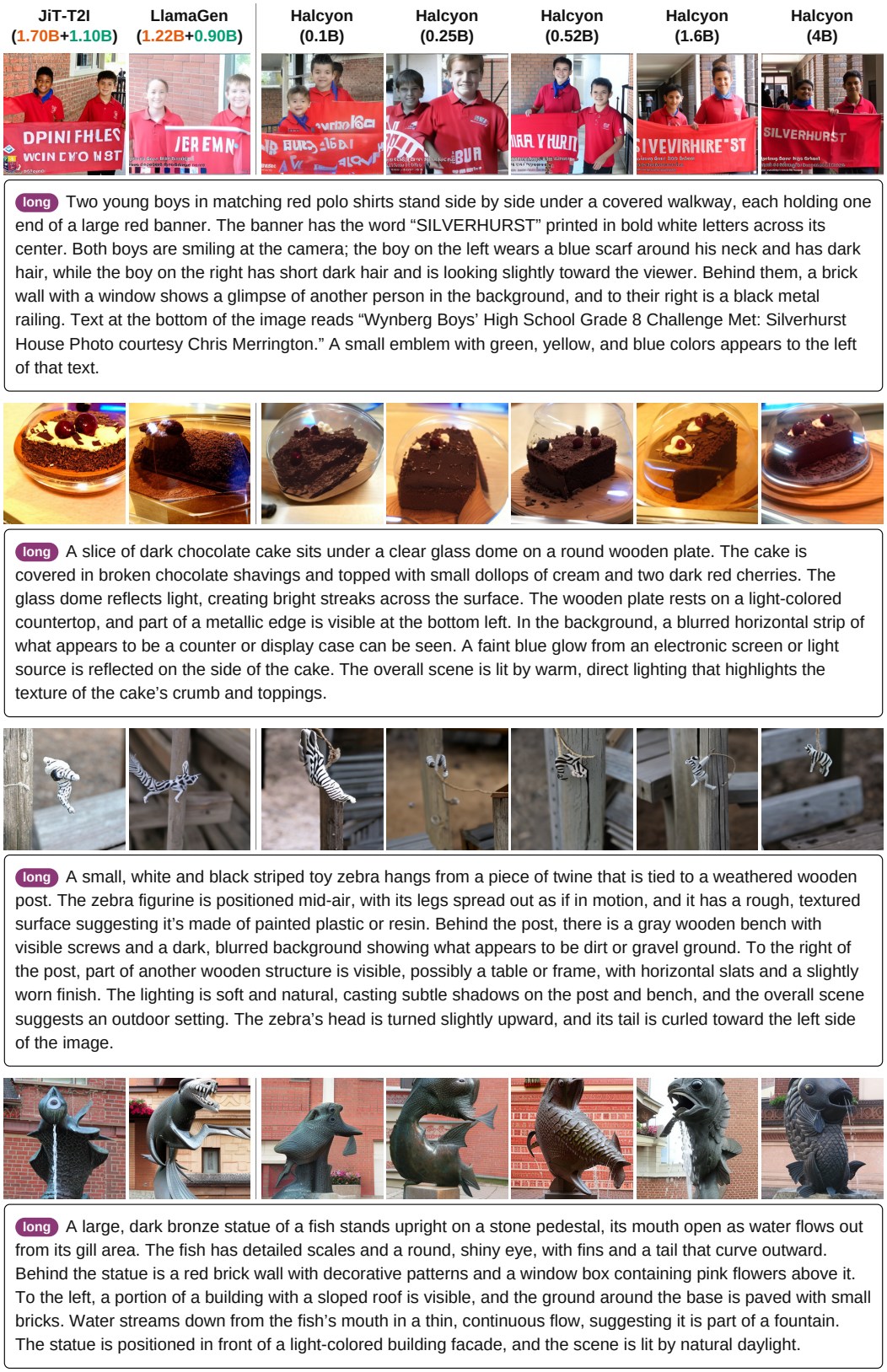


Figure 14: **Comparison of generated images with “long” captions.** From left to right: JiT-T2I-XXL/8++, LlamaGen-XL/8, and `Halcyon` from 0.1B to 4B. For the baselines, we report the sizes of the **language model** and **image generator** separately. Image quality improves as `Halcyon` scales up. All prompts are from GPIC Test-50K.