A LANGUAGE-INFORMED LOSS FOR VISUAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

End-to-end visual generation with pixel-space loss has recently achieved competitive performance without the need for learned image tokenizers. In this work, we study whether language-informed supervision can provide a richer training signal than pixel loss for visual generation. We introduce **LILIE** (pronounced "lily"), a **L**anguage-**I**nformed and **L**ossless **I**mage **E**ncoder. LILIE is an invertible image transform trained such that a vision-language model can recover captions from noised features. The encoder uses a normalizing flow transformer trained jointly with a vision-language model via next token prediction. We then use the encoder to extract a simple language-informed loss for text-to-image generation. We train JiT models with LILIE loss, reducing gFD by 16 points (68.61 $\rightarrow$ 52.73) against a pixel-only baseline on the GPIC dataset. Our code, encoders, and generative models are included in the Supplementary.

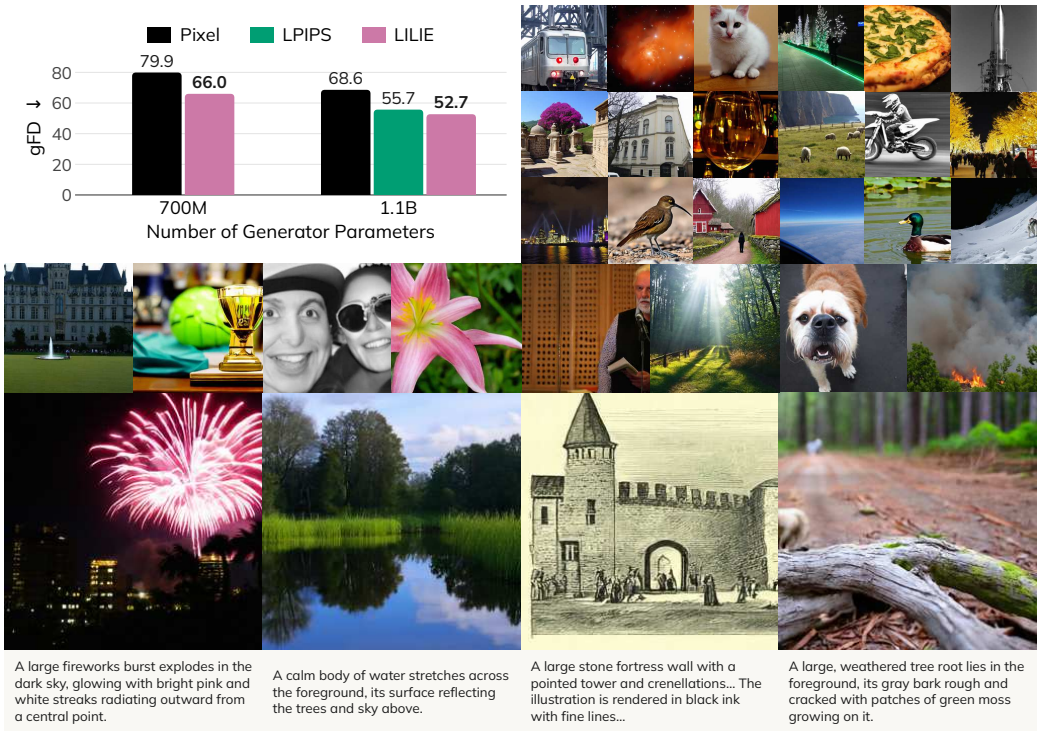

Figure 1: **Generation quality with LILIE.** Upper left: Pixel and LILIE supervision at two generator sizes along with an LPIPS baseline evaluated on GPIC Test-50K with 50 Euler steps. CFG scale is selected using GPIC Validation-50K. All models are trained for one epoch on GPIC Full (Chandrasegaran et al., 2026). Mosaic: selected 256×256 samples from the 1.1B diffusion model with LILIE supervision. Four samples include prompt excerpts. We include randomly selected samples from GPIC Test-50K in Appendix G.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Image generators are typically built in two stages. A tokenizer first compresses images into a learned latent space, and a generative model is then trained in that space. Latent diffusion (Rombach et al., 2022; Peebles & Xie, 2023), autoregressive (Sun et al., 2024), and mixed-modal (Zhou et al., 2025) generators all follow this recipe. By reducing the dimensionality of the modeling problem, tokenization makes generation more tractable and lowers computational cost. However, the lossy tokenizer bounds achievable quality, particularly for fine-grained content (e.g. rendered text and human faces) (Yao et al., 2025; Yang et al., 2026). Recent work removes the first stage entirely. JiT (Li & He, 2026) trains a plain transformer end-to-end on raw pixels and achieves competitive performance with two-stage systems without training a tokenizer at all. Pixels are hence an appealing output space, but we ask whether generation quality can be improved by extracting a richer training signal.

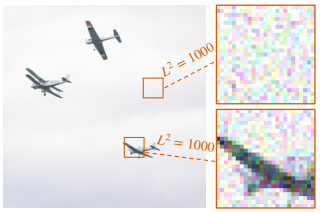

Figure 2: Two perturbed image patches (additive Gaussian noise) with equal pixel Euclidean distance from the original patches. Caption: "Three airplanes fly in the sky against a cloudy backdrop."

Pixel loss measures squared Euclidean distance between predicted and target pixels. However, errors of equal pixel distance can vary in underlying semantic content. Figure 2 shows two perturbations, each with pixel distance of 1000 from the original patches. One perturbation degrades a subject while the other alters a background cloud, but pixel loss treats them the same. We hypothesize that language may help address this gap. Prior work suggests that language provides an inherently semantic signal (Hockett, 1960; Mikolov et al., 2013; Barbieri, 2020). Motivated by the semantic gap between language and vision (Smeulders et al., 2000; Collell & Moens, 2016), we study whether distances in a visual space shaped by language can provide a richer training signal to supervise end-to-end visual generation than pixel loss alone.

We introduce LILIE, a Language-Informed and Lossless Image Encoder. LILIE is an invertible transform from $\mathbb{R}^{H\times W\times 3} \to \mathbb{R}^{H\times W\times 3}$, trained such that a vision-language model can caption the output after it is corrupted by Gaussian noise. We build LILIE on a transformer normalizing flow architecture (Zhai et al., 2025; Lu et al., 2026b), keeping the invertible architecture but replacing the generation objective with a captioning objective. Our method is trained with next token prediction over ground-truth image captions. The encoder learns to arrange its output so that caption-relevant visual information survives the corruption, while invertibility guarantees that no visual information is lost. After training, we discard the VLM and keep LILIE as a simple standalone image transform.

We finally train a family of JiT-T2I (Li & He, 2026; Ma et al., 2026) models on GPIC (Chandrasegaran et al., 2026), a large scale open dataset for text-to-image generation. We first pretrain pixel-space and perceptual loss baselines to provide references on the GPIC corpus. We then study the impact of our training signal with LILIE on generation quality by performing comprehensive ablations. We freeze the encoder, apply it to the predicted and target images, and add squared error in LILIE-space to the standard pixel-space flow-matching loss (Lipman et al., 2023). For our 1.1B parameter JiT-T2I-g, the pixel-only baseline reaches a gFD-DINOv2 (Stein et al., 2023) of 68.61 on the GPIC test set. Replacing the pixel loss entirely with LILIE loss improves this to 55.78, so language supervision alone is already enough to learn a feature space that improves pixel generation. Combining LILIE and pixel losses further improves gFD to 52.73.

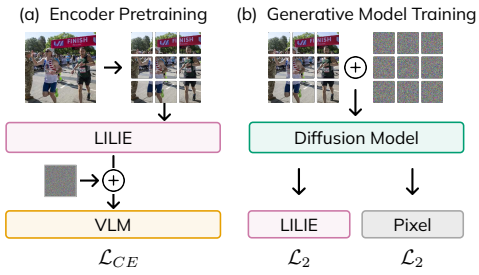

Figure 3: **Training and using LILIE.** (a) We jointly train LILIE and VLM LoRA parameters on noise-corrupted features with next token prediction. (b) The diffusion model trains on Pixel and LILIE L^2 losses, comparing its prediction with the target image in pixel space and through the learned encoder.

To summarize, we propose a language-informed loss that improves pixel-space visual generation quality, and we hope our work encourages the community to adopt language-informed losses for training visual generative models.

We highlight our main contributions below.

- We pretrain language-informed and lossless image encoders.
- We pretrain generative models with perceptual losses (LPIPS, DINOv2, SigLIP 2) on the GPIC corpus to provide baseline references.
- We pretrain generative models with supervision in LILIE-space and achieve a 16-point reduction in gFD over a pixel-only baseline.

2 Related Work

Pixel-space diffusion. Pixel-space diffusion generates images without a learned image tokenizer. Cascaded diffusion models, including Imagen, generate images at progressively higher resolutions (Ho et al., 2022; Saharia et al., 2022), while Simple diffusion and Recurrent Interface Networks support high-resolution generation without cascades (Hoogeboom et al., 2023; Jabri et al., 2023). More recent approaches include cascaded flow matching in PixelFlow (Chen et al., 2025), neural field prediction within image patches in PixNerd (Wang et al., 2026), and joint patch-level and pixel-level processing in PixelDiT (Yu et al., 2026). JiT predicts clean images directly from noisy pixels (Li & He, 2026). Pixel MeanFlow extends clean-image prediction to one-step generation with a MeanFlow objective (Lu et al., 2026a; Geng et al., 2025). We build on JiT, studying the training loss while keeping the generator architecture fixed.

Feature-space losses for generation. Perceptual losses compare images through neural features, providing supervision beyond pixel-wise error (Johnson et al., 2016; Zhang et al., 2018). PixelGen applies this approach by augmenting a pixel-space flow-matching loss with LPIPS and DINOv2 losses (Ma et al., 2026). REPA instead aligns the denoiser's hidden states with features from a pretrained visual encoder (Yu et al., 2025). PixelREPA finds that directly applying this alignment to JiT can reduce sample quality and diversity, and introduces a masked adapter that improves generation (Shin et al., 2026). VA-VAE and representation autoencoders also use pretrained visual features to construct latent spaces for diffusion (Yao et al., 2025; Zheng et al., 2026). In this work, we learn a language-supervised and lossless representation for the loss while keeping generation in pixel space.

Invertible networks. Prior normalizing flow models such as Real NVP and Glow learn invertible transformations that enable exact likelihood evaluation and sampling (Dinh et al., 2017; Kingma & Dhariwal, 2018). TARFlow extends this approach with autoregressive transformers operating on image patches (Zhai et al., 2025). Invertible architectures have also been used for representation learning as in i-RevNet (Jacobsen et al., 2018). IB-INN uses small input perturbations to form an information-bottleneck objective for invertible generative classifiers (Ardizzone et al., 2020). LILIE uses the improved TARFlow architecture introduced in BiFlow (Lu et al., 2026b). We train this bijection through caption prediction from corrupted features and use its learned feature space to measure errors when training a T2I generator.

Language-supervised visual representations. CLIP learns transferable visual features through contrastive image-text supervision (Radford et al., 2021), while SigLIP 2 combines image-text alignment with captioning and self-supervised objectives (Tschannen et al., 2025). VirTex and CapPa demonstrate that caption prediction itself can learn effective visual representations (Desai & Johnson, 2021; Tschannen et al., 2023). Language has also been used to inform image tokenization. QLIP and UniTok combine reconstruction and image-text alignment objectives to learn discrete representations (Zhao et al., 2025; Ma et al., 2025). TexTok conditions image encoding and decoding on captions, improving reconstruction and downstream generation quality (Zha et al., 2025). LILIE uses captions as prediction targets for corrupted image features. Caption cross-entropy is our only training loss.

3 Methodology

Pixel-space losses treat errors across pixels equally, even when they differ in relevance to image content. Feature-space losses instead compare images in a feature space that emphasizes certain

visual properties. Here, we propose a method for learning such a feature space that is lossless and supervised by language.

LILIE learns a bijection $\mathcal{E}_\theta : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^{H\times W\times 3}$ from images to equally-shaped features. We train the encoder such that a finetuned VLM can recover an image's caption from features corrupted with Gaussian noise. This corruption procedure encourages semantic information to remain recoverable under corruption, and invertibility prevents representations from discarding any visual information.

Our complete method has two stages. First, we learn an image encoder $\mathcal{E}_\theta$ from image-caption pairs. We then freeze the encoder and use L^2 in LILIE-space to augment pixel-space loss for training a text-to-image generative model.

3.1 LILIE: A Language-Informed and Lossless Image Encoder

Learning an invertible map. We parametrize $\mathcal{E}_\theta$ using the improved TARFlow architecture from BiFlow (Lu et al., 2026b; Zhai et al., 2025). Unlike prior applications, we do not use this architecture to train a normalizing flow density model. We use it for its invertibility and change the training objective.

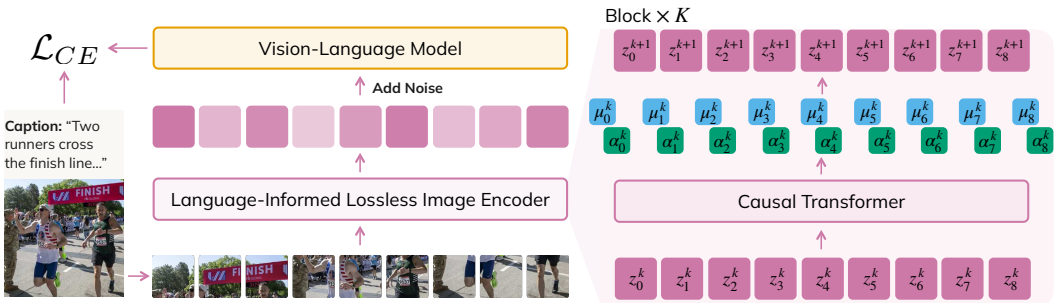

Figure 4: **LILIE training diagram.** We expand on Figure 3(a) to include TARFlow internals. We provide additional details on affine coupling and transformer internals in Appendix A.1.

Figure 4 and Algorithm 1 describe the training procedure to learn $\mathcal{E}_\theta$. Each image is mapped through $\mathcal{E}_\theta$, rescaled to a per-image RMS of $1/2$, corrupted with additive white Gaussian noise, and captioned by a LoRA-finetuned VLM (Hu et al., 2022). The training signal is cross-entropy of the VLM predictions against the ground-truth GPIC caption. We backpropagate to jointly update $\mathcal{E}_\theta$ and the VLM's LoRA parameters. The RMS target of $1/2$ approximately matches the RMS of natural images in the GPIC corpus, keeps LILIE feature scales in-domain for the VLM, and prevents $\mathcal{E}_\theta$ from enlarging all features to compensate for the corruption. We also track an EMA estimate of the raw features' second moment, averaged over all features, which we use for downstream rescaling. After training, we discard the VLM and retain only the encoder and the second moment estimate.

3.2 Training a T2I Generative Model with LILIE

We now freeze the LILIE encoder $\mathcal{E}_\theta$ and use it to supervise pixel-space text-to-image generation.

Base objective. We build on JiT-T2I (Li & He, 2026; Ma et al., 2026; Chandrasegaran et al., 2026), a plain transformer architecture that denoises pixels directly, conditioned on caption embeddings from a frozen Qwen3-1.7B text encoder (Yang et al., 2025). The JiT x-prediction loss is

$$\mathcal{L}_{\text{Pixel}} = \mathbb{E}[w(t)\|x - \hat{x}\|_2^2], \quad w(t) = \frac{1}{\max(1-t, t_{\text{clip}})^2},$$

the velocity-matching objective rewritten in x-prediction form (Li & He, 2026). The caption is dropped for a random 10% of samples to learn the unconditional path and support classifier-free guidance at sampling time (Ho & Salimans, 2022).

Algorithm 1 LILIE training.

Input: Image-caption batches `data`, initial invertible `encode`, pretrained `vlm` with LoRA, noise level `sigma`, EMA decay `beta`.
Output: Trained `encode` and second moment `m2`.

```
for image, caption in data:
   x = preprocess(image) # Appendix A
   z = encode(x)
   u = z / (2 * rms(z)) # per image
   y = u + sigma * randn_like(u)
   loss = cross_entropy(vlm(y), caption)
   update_params(loss, encode, vlm.lora)

   # Track the second moment (EMA)
   q = mean(square(z.detach()))
   m2 = ema_update(m2, q, beta)

return encode, m2
```

Algorithm 2 JiT training loss.

Input: Frozen `encode` and `m2` from Algorithm 1; target `x`, prediction `x_pred`, time weight `w(t)`, loss weight `lambda`.
Output: Generator training loss.

```
# Fixed rescale, preserves invertibility
def encode_for_loss(x):
   return encode(x) / (2 * sqrt(m2))

z = encode_for_loss(x)
z_pred = encode_for_loss(x_pred)

loss_pixel = l2_loss(x - x_pred)
loss_lilie = l2_loss(z - z_pred)
loss = w(t) * (loss_pixel
           + lambda * loss_lilie)

return loss
```

Augmenting with LILIE. We freeze $\mathcal{E}_\theta$ and additionally score the prediction in LILIE-space. Analogous to the rescaling performed in encoder training, we define

$$\tilde{\mathcal{E}}(x) \triangleq \frac{\mathcal{E}_\theta(x)}{2\sqrt{m_2}}, \tag{1}$$

where m_2 is an EMA estimate of the second moment computed during training (Algorithm 1). Rescaling for $\tilde{\mathcal{E}}$ only depends on corpus statistics, so $\tilde{\mathcal{E}}$ remains a bijection. We finally use $\tilde{\mathcal{E}}$ to construct our training objectives. We consider two objectives. The first is JiT supervision *solely* in LILIE-space, and we additionally combine pixel and LILIE-space losses.

$$\mathcal{L}_{\text{LILIE}} = \mathbb{E}[w(t)\|\tilde{\mathcal{E}}(x) - \tilde{\mathcal{E}}(\hat{x})\|_2^2], \qquad \mathcal{L}_{\text{Pixel+LILIE}} = \mathcal{L}_{\text{Pixel}} + \lambda\mathcal{L}_{\text{LILIE}},$$

where λ is a loss weighting hyperparameter. Algorithm 2 summarizes the Pixel+LILIE JiT training loss.

4 Experiments

We now study the impact of LILIE on pixel-space text-to-image generation quality.

4.1 Experimental Setup

Data. All models are trained on the GPIC corpus of image-caption pairs (Chandrasegaran et al., 2026) at 256×256 resolution. Following the experimental setup from GPIC, we train both the encoder and diffusion models over one epoch on GPIC Full (100M train examples) at a global batch size of 256, yielding 390,600 steps. Every image in the GPIC corpus is paired with a caption from one of four length classes (tag, short, medium, long), which we match with a corresponding prompt for LILIE training (Appendix F).

Encoders. Unless otherwise stated, the encoder is LILIE-S, a 32M parameter improved TARFlow (Lu et al., 2026b) with $K = 4$ blocks, $L = 4$ layers per block, bottleneck dimension $H = 384$, and patch size 16. We use SmolVLM-256M (Marafioti et al., 2025), which we finetune with rank 64 LoRA. We choose SmolVLM-256M as it is a compact openly available VLM, at $8\times$ the size of our encoder. Training LILIE-S takes approximately 40 hours on TPU v5e-32. We detail the full configuration in Table B.2.

Generative models. We train JiT-T2I-XL and -g models (Li & He, 2026; Ma et al., 2026) with 700M and 1.1B parameters, respectively. Both condition on a frozen Qwen3-1.7B text encoder

Table 1: **Generation results on GPIC.** We compare generation quality achieved by JiT-T2I-XL (700M) and JiT-T2I-g (1.1B) models trained with different loss functions. We compare Pixel, LILIE-only, and Pixel+LILIE losses at the 700M generator size. At 1.1B, we compare • Pixel loss; • standalone feature-space losses with DINOv2, SigLIP 2, LPIPS (VGG16), and LILIE backbones; • feature-space losses complimentary with Pixel loss. Lower is better for gFD, higher is better for precision, recall, coverage, and values near 1 are best for density. We bold rows achieving lowest gFD. "GPIC compliant" indicates compliance with the GPIC evaluation protocol. We note that the DINOv2 and SigLIP 2 models used for feature-space losses are both distilled from larger models, while VGG16 and LILIE are not distilled. DINOv2, SigLIP 2, and LPIPS losses result in degenerate generation quality when used without a Pixel loss term. **Key result:** LILIE loss yields competitive quality for end-to-end visual generation, both when used as a standalone loss and as complimentary with Pixel loss. Table B.1 includes selected CFG values and gFD decomposition.

Generator	GPIC compliant	Loss	gFD ↓	Precision ↑	Recall ↑	Density → 1	Coverage ↑	MMD ↓
JiT-T2I-XL (700M)		Pixel	79.94	0.935	0.793	0.982	0.915	0.052
		LILIE	68.88	0.941	0.829	0.996	0.930	0.046
		Pixel+LILIE	**66.00**	**0.944**	**0.831**	**1.001**	**0.932**	**0.042**
JiT-T2I-g (1.1B)		Pixel	68.61	0.938	0.826	0.987	0.925	0.040
	✗	DINOv2	3124.01	0.987	0.000	1.082	0.048	15.225
	✗	SigLIP 2	3161.15	0.952	0.000	1.172	0.050	14.817
	✓	LPIPS (VGG16)	158.45	0.922	0.651	1.053	0.848	0.174
	✓	LILIE	55.78	0.947	0.855	1.028	0.942	0.031
	✗	Pixel+DINOv2	**47.88**	**0.951**	**0.864**	**1.051**	**0.950**	**0.023**
	✗	Pixel+SigLIP 2	52.55	0.956	0.847	1.064	0.949	0.026
	✓	Pixel+LPIPS	55.70	0.948	0.848	1.034	0.938	0.025
	✓	Pixel+LILIE	52.73	0.946	0.856	1.034	0.946	0.028

(Yang et al., 2025). We primarily compare three objectives: Pixel, LILIE, and Pixel+LILIE, while holding the generator configuration constant. Our main results train for one epoch on GPIC Full (390,600 steps), and ablations use 200K steps. Training JiT-T2I-XL and -g takes approximately 27 and 40 hours on TPU v6e-16, respectively. We detail the full configurations in Table B.3.

Perceptual loss baselines. Recent work has shown that perceptual supervision with LPIPS and DINOv2 significantly benefits pixel-space diffusion generation quality (Ma et al., 2026). These methods follow our general paradigm of combining pixel and feature-space losses, so we pretrain JiT models with perceptual supervision to provide reference baselines. We consider LPIPS with VGG16 (Zhang et al., 2018) and perceptual losses in DINOv2 ViT-B/14 (Oquab et al., 2023) and SigLIP 2 ViT-B/16@256px (Tschannen et al., 2025) feature spaces. We pretrain each baseline using a combination of pixel-space loss with a single feature-space loss, and we follow the setup from PixelGen (Ma et al., 2026) while disabling REPA feature alignment. SigLIP 2 provides a language-aligned baseline, and we configure it in an identical manner to P-DINO in PixelGen using cosine dissimilarity. We additionally evaluate each feature space alone, without any pixel-space loss, using L^2 distance for DINOv2 and SigLIP 2. We provide full implementation details in Appendix B.1.

Evaluation. We report Fréchet distance (gFD), precision, recall, density, and coverage (PRDC) (Kynkäänniemi et al., 2019; Naeem et al., 2020), and squared maximum mean discrepancy (MMD) all in the DINOv2 representation space (Oquab et al., 2023; Stein et al., 2023) following the GPIC evaluation protocol. We use generator EMA weights for sampling and select classifier-free guidance (CFG) separately for each model by minimizing gFD on GPIC Validation-50K, then report results on Test-50K. Unless varied below, sampling uses 50 Euler steps.

Hyperparameter selection. We first select the noise level σ used to train the encoder and loss weight λ used to train the generator using JiT-T2I-XL at 200K training steps. We compare five noise levels and three mixing weights, as well as the LILIE-only objective. The best setting in this sweep is $\sigma = 0.25$ and $\lambda = 2$, which we use for our main results and ablations. Appendix C details the procedure and full sweep table.

4.2 Generation Results

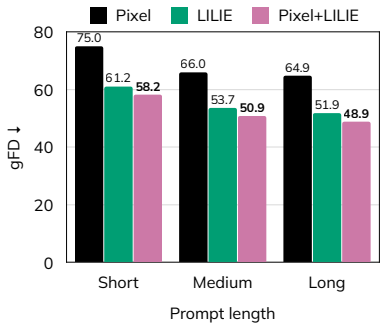

Figure 5: **gFD over prompt length.** We take three sets of 50K prompts from GPIC Test, each containing only short, medium, or long captions. We then sample from the Pixel, LILIE, and Pixel+LILIE JiT-T2I-g models at their respective optimal CFG scales. We report gFD over the reference GPIC Test set. Our optimal Pixel+LILIE configuration outperforms the Pixel baseline at every prompt length.

LILIE improves generation quality. Adding LILIE loss to pixel loss improves gFD, precision, recall, coverage, and MMD for both generators (Table 1). For JiT-T2I-XL, Pixel+LILIE reduces gFD from 79.94 to 66.00, and for JiT-T2I-g, it reduces gFD from 68.61 to 52.73. Precision and recall improve alongside gFD reduction, so our gain in generation quality results from both higher sample fidelity and better coverage of the real data distribution. Figure 1 shows samples from the JiT-T2I-g model trained with Pixel+LILIE loss. Figure 5 provides gFD results when varying prompt length.

LILIE loss can be used on its own. Replacing pixel loss entirely with LILIE loss reduces gFD from 79.94 to 68.88 for JiT-T2I-XL and from 68.61 to 55.78 for JiT-T2I-g. In contrast, we find that perceptual loss baselines fail when used without a pixel loss term (Table 1). Since LILIE is a bijection, matching the target image in LILIE-space exactly recovers the target in pixel space, and LILIE-space alone provides a richer training signal than pixels. LILIE therefore provides a useful training objective even without a pixel loss term, while combining the two yields the best generation quality.

4.3 Sampling Steps

We next vary the number of Euler steps over $10, 15, 20, 25, 50, 75$ (Figure 6). The curves and examples compare JiT-T2I-g models trained to 390,600 steps. Both Pixel and Pixel+LILIE benefit from increasing the budget from 10 to 75 steps. Pixel+LILIE has lower gFD than Pixel at every tested sampling budget. At 25 steps, Pixel+LILIE reaches a gFD of 58.23, already below Pixel with 50 steps at 68.61. The learned loss can therefore improve the tradeoff between generation quality and the number of sampling steps. Beyond 50 steps, the gains are small. Moving to 75 steps reduces gFD from 68.61 to 68.29 for Pixel and from 52.73 to 52.65 for Pixel+LILIE. LILIE sees an increase in gFD from 55.78 to 56.63 from 50 to 75 steps. We retain 50 Euler steps to compare all methods as both Pixel and Pixel+LILIE see minimal gains from moving to 75 steps, and our choice is consistent with JiT-T2I sampling in GPIC (Chandrasegaran et al., 2026). Figure G.1 additionally provides examples of JiT clean image predictions along a 50 step trajectory.

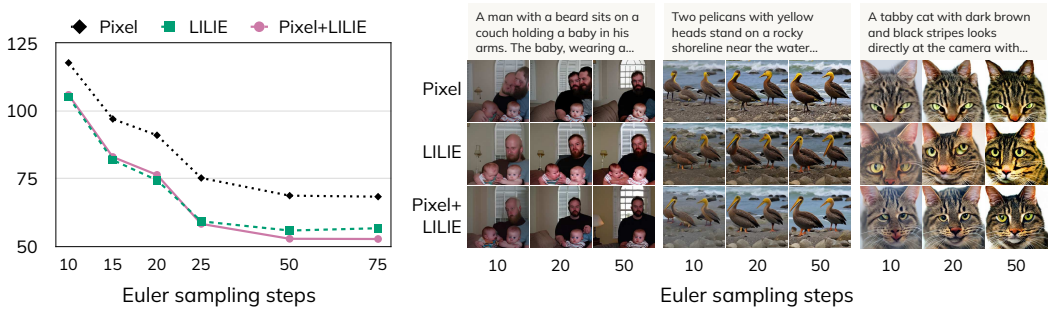

Figure 6: **Effect of Euler sampling steps.** Left: JiT-T2I-g gFD on GPIC Test-50K at 390,600 training steps, with each model's CFG scale chosen at 50 steps and held constant. Right: three selected qualitative examples with Euler steps increasing across columns and Pixel, LILIE, and Pixel+LILIE down rows. Prompts and initial noise are shared across methods and step counts.

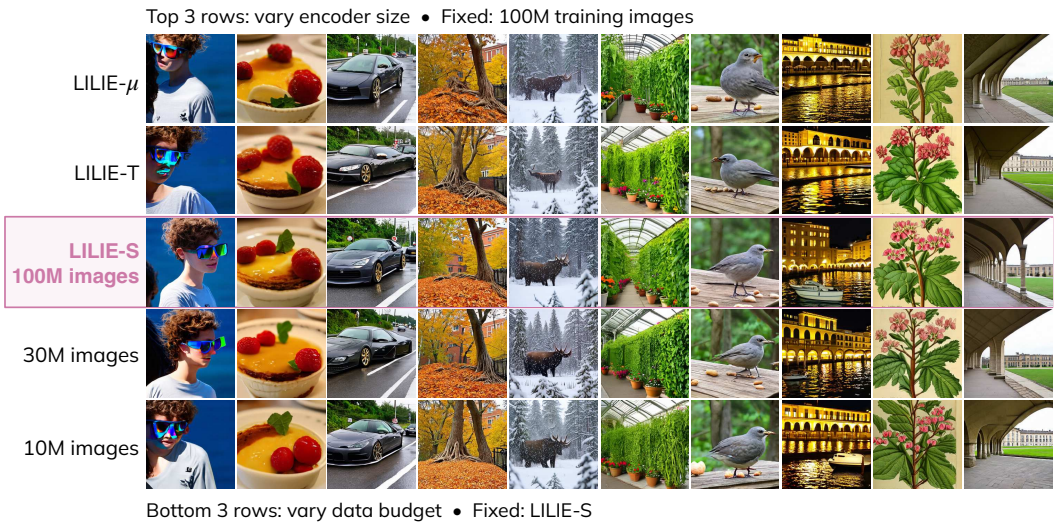

Figure 7: **Qualitative samples over LILIE model size and training data.** Selected qualitative JiT-T2I-g samples from GPIC use the same prompt and initial noise in each column. The highlighted row is LILIE-S/100M, our main configuration. Model size decreases above it, and training data decreases below it. All generators are trained for 200K steps.

Table 2: **Model size ablation.** gFD ↓ at 200K generator steps. All LILIE encoders train on 100M images. The default LILIE-S encoder row is highlighted.

Encoder	Params	JiT-T2I-XL	JiT-T2I-g
Pixel	–	114.59	101.66
LILIE-μ	1.5M	96.11	84.66
LILIE-T	9.1M	92.66	80.87
LILIE-S	32M	**91.79**	**76.59**

Table 3: **Training data ablation.** gFD ↓ at 200K generator steps. All encoders are LILIE-S. The default 100M-image training budget row is highlighted.

# Training Images	JiT-T2I-XL	JiT-T2I-g
Pixel	114.59	101.66
10M	97.79	83.01
30M	93.52	81.18
100M	**91.79**	**76.59**

4.4 Encoder Size and Data Scaling

We vary the model size and training budget for LILIE while keeping the generator setup fixed. All generators in this ablation are trained for 200K steps. The encoder rows use Pixel+LILIE loss. The LILIE-S encoder trained on 100M images is the common setting for both comparisons, highlighted in Table 2, Table 3, and Figure 7. We provide gFD decompositions, PRDC, and MMD results in Appendix D.

Model size. We train LILIE-μ (micro), LILIE-T (tiny), and LILIE-S (small) encoders containing 1.5M, 9.1M, and 32M parameters, respectively. All encoders are trained for 1 epoch on GPIC Full (100M image-caption pairs). Increasing encoder size improves gFD for both generators. For JiT-T2I-g, gFD decreases from 84.66 with LILIE-μ to 80.87 with LILIE-T and 76.59 with LILIE-S. Even LILIE-μ improves over the Pixel baseline of 101.66, so a 1.5M parameter encoder already provides a strong training signal. Table B.2 in Appendix B details the model configurations.

Training data. With the LILIE-S architecture fixed, increasing LILIE training from 10M to 30M and 100M images reduces gFD from 97.79 to 93.52 to 91.79, respectively, for JiT-T2I-XL. For g, the same budgets reduce gFD from 83.01 to 81.18 and 76.59. Hence, increasing the encoder's data and training budget improves generation quality without increasing the generator's own training budget.

5 Discussion and Conclusion

We have shown that language supervision alone is sufficient to learn a feature space that improves downstream generation quality. Pixel-space losses measure differences without regard to image content. LILIE instead learns its representation through language supervision. Information needed to describe an image must survive corruption, while invertibility guarantees that no visual information is lost. Figure 8 illustrates an important property of the proposed language-informed loss. Near real data, at $t = 0.99$, the learned representation amplifies residual standard deviation by approximately $4.9\times$ relative to pixels despite near-equal scale between raw pixel values and LILIE features. Hence LILIE makes these small deviations more prominent in the supervision and provides a stronger training signal closer to the data distribution, whereas pixel differences provide a comparatively weak signal.

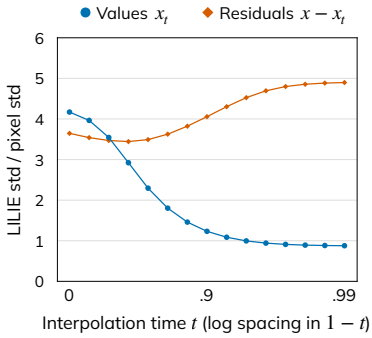

Figure 8: **LILIE provides a stronger training signal closer to the data distribution.** Average standard deviation ratios for LILIE versus pixel values x_t, and residuals from clean images x, using the LILIE-S encoder on GPIC Nano (1M train images). x_t denotes linear interpolation between pure noise ($t = 0$) and real data ($t = 1$).

Comparisons with pretrained perceptual losses. Table 4 summarizes the feature spaces we compare. Our LILIE-S encoder only has 32M parameters and is trained on 100M image-caption pairs from the GPIC dataset. DINOv2 and SigLIP 2 introduce different pretraining data and budgets. DINOv2 uses LVD-142M (142M images) (Oquab et al., 2023), while SigLIP 2 trains on WebLI, a corpus of 10B images (Tschannen et al., 2025). Without access to the same pretraining corpora and compute, we cannot match the foundation models' data distributions. Furthermore, we note that models trained with DINOv2 and SigLIP 2 features violate the GPIC evaluation protocol (Chandrasegaran et al., 2026). Both are large vision foundation models trained on sets of data that may overlap with the GPIC test set. Furthermore, GPIC evaluation uses DINOv2 features itself. We therefore regard DINOv2- and SigLIP 2-based losses as useful comparisons between pretrained representations but not as controlled experiments. LPIPS is the one GPIC-compliant baseline, with its VGG16 backbone trained on 1.28M ImageNet images. LILIE improves on it by three gFD points.

Table 4: **Comparison of feature-space losses.** "GPIC compliant" indicates compliance with the GPIC evaluation protocol. "SSL" refers to self-supervised learning, and "Language" refers to language-supervised learning. The VGG16 backbone is label-supervised, trained for ImageNet classification. SigLIP 2 is trained with both language-supervised and SSL objectives. Our method is only supervised by language.

Loss	Supervised	SSL	Language	Training images	Not distilled	GPIC compliant
DINOv2	-	✓	-	142M	✗	✗
SigLIP 2	-	✓	✓	10B	✗	✗
LPIPS (VGG16)	✓	-	-	1.28M	✓	✓
LILIE	-	-	✓	100M	✓	✓

A simple language-informed loss for visual generation. Using LILIE only requires the frozen encoder and squared-error loss. The encoder is intentionally kept small at 32M parameters so that it amounts to less than 5% of the generator's parameters in all of our experiments. The generative model remains end-to-end and predicts pixels directly. We use no additional loss schedule. LILIE works as a standalone loss for visual generation, without any pixel-loss term. LILIE loss improves JiT-T2I-g gFD from 68.61 to 55.78, while combining the losses reaches 52.73 (Table 1). A simple change to the training loss therefore improves generation quality. We discuss limitations of our work in Appendix E.

AI USE STATEMENT

In this work, we used generative AI tools to aid in implementing methods. We have not used generative AI tools for proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, supporting qualitative and thematic data analysis, or interpreting results. Generating synthetic datasets, helping develop 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 and reformatting a dataset are not applicable to this work. Additionally, we used generative AI tools for creation or editing of software code, sourcing/searching for information, creating or modifying scientific figures, and editing a research paper to improve readability. We have reviewed all AI-assisted work. We verified all LLM-generated code with manual inspection and detailed correctness testing. We verified all information presented by AI directly from the respective sources. We used AI only for aesthetics for scientific figures and ensured that all represented data is accurate. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work improves the capabilities of visual generative models, which have the potential for misuse. In addition, our encoders and generative models are trained on the GPIC corpus and inherit any biases present in the dataset. Our work itself does not involve human studies or constructing datasets. We have not encountered any potentially harmful insights.

REPRODUCIBILITY STATEMENT

We outline our approach in Sections 3 and 4, and we provide full training, sampling, and evaluation details for both LILIE and JiT-T2I models in the Appendix. We have additionally included our training and sampling code in the Supplementary, and our model checkpoints are available in an anonymized HuggingFace repository (link provided in the code submission). The GPIC corpus and `gpic-eval` package that we use for pretraining and evaluation are fully open, as are the SmolVLM-256M and Qwen3-1.7B checkpoints which we use for LILIE and JiT-T2I training, respectively.

REFERENCES

Lynton Ardizzone, Radek Mackowiak, Carsten Rother, and Ullrich Köthe. Training normalizing flows with the information bottleneck for competitive generative classification. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7828–7840. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/593906af0d138e69f49d251d3e7cbed0-Paper.pdf.

Marcello Barbieri. The semantic theory of language. *Biosystems*, 190:104100, 2020. ISSN 0303-2647. doi: https://doi.org/10.1016/j.biosystems.2020.104100. URL https://www.sciencedirect.com/science/article/pii/S0303264720300083.

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, 2026. URL https://arxiv.org/abs/2605.30341.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. PixelFlow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025. URL https://arxiv.org/abs/2504.07963.

Guillem Collell and Marie-Francine Moens. Is an image worth more than a thousand words? On the fine-grain semantic differences between visual and linguistic representations. In Yuji Matsumoto and Rashmi Prasad (eds.), *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2807–2817, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/C16-1264/.

Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11162–11173, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Desai_VirTex_Learning_Visual_Representations_From_Textual_Annotations_CVPR_2021_paper.html.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025. URL https://papers.neurips.cc/paper_files/paper/2025/hash/6d13e085b79d454da5910e4ca82a3d9d-Abstract-Conference.html.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. URL https://jmlr.org/papers/v23/21-0635.html.

Charles Francis Hockett. The origin of speech. *Scientific American*, 203:89–96, 1960. URL https://api.semanticscholar.org/CorpusID:17630776.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 13213–13232. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/hoogeboom23a.html.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14569–14589. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/jabri23a.html.

Jörn-Henrik Jacobsen, Arnold W.M. Smeulders, and Edouard Oyallon. i-RevNet: Deep invertible networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJsjkMb0Z.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *CoRR*, abs/1904.06991, 2019.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36115–36125, June 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. In *International Conference on Machine Learning*, 2026a.

Yiyang Lu, Qiao Sun, Xianbang Wang, Zhicheng Jiang, Hanhong Zhao, and Kaiming He. Bidirectional normalizing flow: From data to noise and back. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2069–2078, June 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lu_Bidirectional_Normalizing_Flow_From_Data_to_Noise_and_Back_CVPR_2026_paper.html.

Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. UniTok: a unified tokenizer for visual generation and understanding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 129274–129297. Curran Associates, Inc., 2025. doi: 10.52202/085713-4306. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/bbe04d329d531e8814f1199098bd8fb6-Paper-Conference.pdf.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. PixelGen: Pixel diffusion beats latent diffusion with perceptual loss. *arXiv preprint arXiv:2602.02493*, 2026. URL https://arxiv.org/abs/2602.02493.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. SmolVLM: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*, 2013. URL https://api.semanticscholar.org/CorpusID:16447573.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, 2020.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494. Curran Associates, Inc., 2022. doi: 10.52202/068431-2643. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/ec795aeadae0b7d230fa35cbaf04c041-Paper-Conference.pdf.

Prem Seetharaman, Gordon Wichern, Bryan Pardo, and Jonathan Le Roux. Autoclip: Adaptive Gradient Clipping For Source Separation Networks. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, September 2020. doi: 10.1109/MLSP49062.2020.9231926. URL https://www.merl.com/publications/TR2020-132.

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think, 2026. URL https://arxiv.org/abs/2603.14366.

Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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.

Michael Tschannen, Manoj Kumar, Andreas Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46830–46855. Curran Associates, Inc., 2023. doi: 10.52202/075280-2029. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/92369a01fbe8046a093746389b2c413e-Paper-Conference.pdf.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://arxiv.org/abs/2502.14786.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. PixNerd: Pixel neural field diffusion. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.23268.

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

Jiawei Yang, Tianhong Li, Lijie Fan, Yonglong Tian, and Yue Wang. Latent denoising makes good tokenizers. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 61642–61668, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/649ad92e7067b3553a0f15acac68806d-Paper-Conference.pdf.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025. URL https://sihyun.me/REPA/.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. PixelDiT: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://github.com/NVlabs/PixelDiT.

Kaiwen Zha, Lijun Yu, Alireza Fathi, David A Ross, Cordelia Schmid, Dina Katabi, and Xiuye Gu. Language-guided image tokenization for generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15713–15722, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Zha_Language-Guided_Image_Tokenization_for_Generation_CVPR_2025_paper.pdf.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Ángel Bautista, Navdeep Jaitly, and Joshua M. Susskind. Normalizing flows are capable generative models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74348–74369. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhai25d.html.

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.

Yue Zhao, Fuzhao Xue, Scott Reed, Linxi Fan, Yuke Zhu, Jan Kautz, Zhiding Yu, Philipp Krähenbühl, and De-An Huang. QLIP: Text-aligned visual tokenization unifies auto-regressive multimodal understanding and generation. *arXiv preprint arXiv:2502.05178*, 2025. URL https://arxiv.org/abs/2502.05178.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3c4141c12660ad3625eb4ae845e0a6f9-Paper-Conference.pdf.

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 *International Conference on Learning Representations*, 2025.

APPENDIX

A Additional Training Details **15**

A.1 Encoder Architecture 15

B Experimental Details **16**

B.1 Implementation of Perceptual Loss Baselines 18

C Noise Level and Loss Weight **19**

D Model Size and Data Ablations **19**

E Limitations **20**

F Captioning Prompts **20**

G Samples **21**

A ADDITIONAL TRAINING DETAILS

`preprocess` in Algorithm 1. Each GPIC image is resized so that its shorter side is 256 pixels using bicubic interpolation with antialiasing, then center cropped to 256×256. Pixel values $p \in [0, 255]$ are mapped to $[-1, 1]$ as $2p/255 - 1$.

SmolVLM input. The noisy features are passed to SmolVLM-256M in place of an image. We do not clamp them to $[-1, 1]$, allowing values outside the usual image range. Since the VLM is finetuned with LoRA during LILIE training, it adapts to these previously out-of-domain input values.

A.1 ENCODER ARCHITECTURE

Figure A.1 expands the encoder in Figure 4. Each image is split into T patches of dimension D. The encoder applies K autoregressive affine blocks, alternating the patch order used by successive blocks, and reassembles the outputs into an image-shaped feature tensor.

Block parameters. For patch t, the coupling parameters (α_t, μ_t) are log-scale and shift vectors. They depend only on patches preceding position t in the block's order. Both vectors are clipped to $[-1, 1]$ before applying the affine map shown in Figure A.1(a). The first patch passes through unchanged.

Transformer internals. Each block projects patch vectors from D dimensions to bottleneck dimension H, adds learned positional embeddings, and applies L causal transformer layers. Each layer uses $H/64$ attention heads and an MLP with expansion ratio four and GELU activation, with residual connections around attention and the MLP. Attention uses layer normalization. The final zero-initialized linear head produces $2D$ values per position. We drop the last position, prepend zeros, and clip the result. This shift makes each patch's parameters depend only on earlier patches. The model-size ablation varies K, H, and L, and the configurations are detailed in Table B.2.

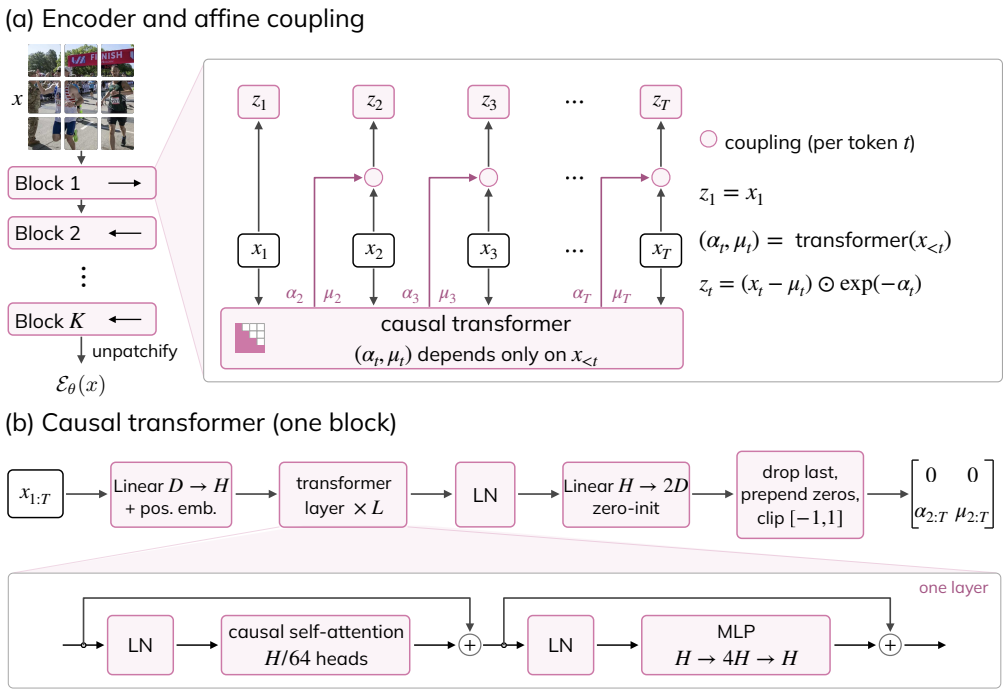

Figure A.1: **Encoder implementation details.** (a) The block stack and autoregressive affine map. The tuple $\gamma_t = (\alpha_t, \mu_t)$ contains the log-scale and shift of patch t. (b) The causal transformer within one block, with bottleneck dimension H and L layers. LN denotes layer normalization. The lower row expands one transformer layer.

B Experimental Details

Table B.2 and Table B.3 summarize configurations for LILIE and JiT training. Table B.1 expands on Table 1, including gFD decomposition and selected CFG values. Figure B.1 shows gFD on GPIC Validation-50K over CFG for JiT-T2I-XL and JiT-T2I-g trained with Pixel, LILIE, and Pixel+LILIE.

Table B.1: **Full generation results on GPIC Test-50K.** Models train for 390,600 steps and use 50 Euler steps. CFG is selected on GPIC Validation-50K at 0.05 granularity. For JiT-T2I-g, LPIPS, DINOv2, and SigLIP 2 denote perceptual losses in the corresponding feature space, used alone and combined with the pixel loss. Despite the large compute cost associated with each pretraining run (Section 4), we perform two pretraining trials for both the encoder and generator in the JiT-T2I-XL with Pixel+LILIE setting to provide error estimates. For this configuration, we report mean $\pm$ standard deviation for each metric over two encoder and generator pretraining trials with different seeds (`100` and `102`). gFD is decomposed into mean (gFD_μ) and covariance (gFD_Σ) components. Precision, recall, density, and coverage measure sample fidelity and diversity. MMD is reported in raw squared units.

Generator	Loss	CFG	gFD ↓	gFD_μ ↓	gFD_Σ ↓	Precision ↑	Recall ↑	Density → 1	Coverage ↑	MMD ↓
JiT-T2I-XL (700M)	Pixel	6.30	79.94	14.07	65.88	0.935	0.793	0.982	0.915	0.052
	LILIE	5.10	68.88	12.59	56.28	0.941	0.829	0.996	0.930	0.046
	Pixel+LILIE	5.10	**66.00** ± 1.16	**11.67** ± 0.42	**54.34** ± 0.74	**0.944** ± 0.0001	**0.831** ± 0.002	**1.001** ± 0.002	**0.932** ± 0.002	**0.042** ± 0.001
JiT-T2I-g (1.1B)	Pixel	6.00	68.61	10.87	57.74	0.938	0.826	0.987	0.925	0.040
	DINOv2	7.20	3124.01	1556.14	1567.87	0.987	0.000	1.082	0.048	15.225
	SigLIP 2	13.50	3161.15	1535.85	1625.30	0.952	0.000	1.172	0.050	14.817
	LPIPS (VGG16)	6.15	158.45	45.97	112.48	0.922	0.651	1.053	0.848	0.174
	LILIE	5.00	55.78	8.48	47.30	0.947	0.855	1.028	0.942	0.031
	Pixel+DINOv2	4.60	**47.88**	**6.09**	**41.79**	**0.951**	**0.864**	**1.051**	**0.950**	**0.023**
	Pixel+SigLIP 2	4.60	52.55	7.03	45.52	0.956	0.847	1.064	0.949	0.026
	Pixel+LPIPS	6.35	55.70	6.53	49.17	0.948	0.848	1.034	0.938	0.025
	Pixel+LILIE	4.85	52.73	7.49	45.24	0.946	0.856	1.034	0.946	0.028

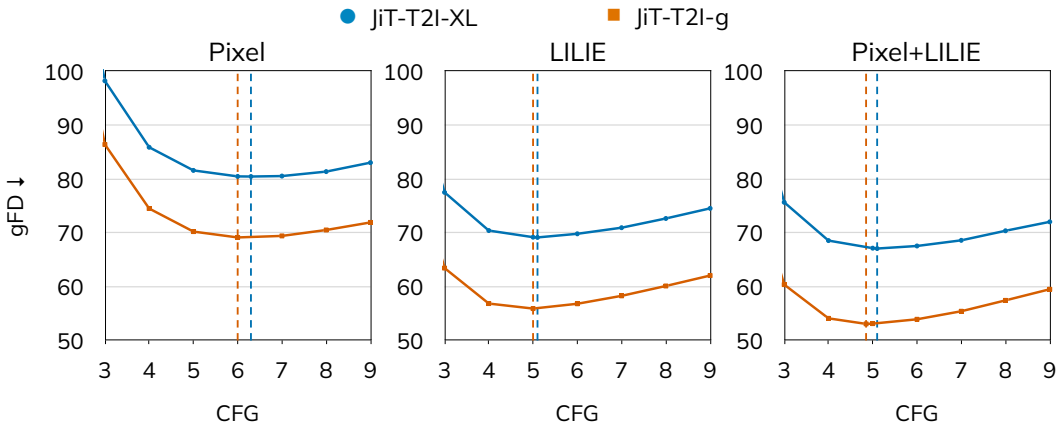

Figure B.1: **gFD over CFG on GPIC Validation-50K.** We show gFD on the GPIC Validation-50K set against CFG for JiT-T2I-XL and JiT-T2I-g models for Pixel, LILIE, and Pixel+LILIE. Selected CFG scales in Table B.1 are indicated with dashed vertical lines.

Table B.2: **LILIE configurations.**

	LILIE-μ	**LILIE-T**	**LILIE-S**
architecture			
flow backbone	Improved TARFlow (Zhai et al., 2025; Lu et al., 2026b)		
blocks K	2	4	4
layers per block L	2	4	4
bottleneck dim H	128	192	384
heads	2	3	6
parameters	1.5M	9.1M	32M
image size	256		
patch size	16		
coupling clip range	1.0		
VLM	SmolVLM-256M (Marafioti et al., 2025)		
LoRA (Hu et al., 2022) rank	64		
LoRA α	32		
LoRA targets	all linear layers except the output head		
max caption tokens	256		
training			
training steps	390,600 (1 epoch on GPIC Full)		
warmup steps	20,000		
optimizer	AdamW (Loshchilov & Hutter, 2019), $\beta_1, \beta_2 = 0.9, 0.95$		
batch size	256		
flow learning rate	1.2e-3	8e-4	4e-4
VLM learning rate	4e-4		
learning rate schedule	linear warmup, cosine decay to 1e-6		
weight decay	1e-4		
gradient clipping	AutoClip (Seetharaman et al., 2020), 10th percentile		
noise level σ	0.25		
feature RMS target	1/2		
ema decay for m_2	0.99		

Table B.3: **Generative model configurations.**

	JiT-T2I-XL	JiT-T2I-g
architecture		
depth	24	16
hidden dim	1024	1536
heads	16	24
bottleneck	128	256
parameters	700M	1.1B
image size	256	
patch size	16	
text encoder	Qwen3-1.7B (Yang et al., 2025), frozen	
max caption tokens	300	
training		
training steps	200K (ablations), 390,600 (1 epoch on GPIC Full)	
warmup steps	4,000	
optimizer	AdamW (Loshchilov & Hutter, 2019), $\beta_1, \beta_2 = 0.9, 0.95$	
batch size	256	
learning rate	1e-4	
learning rate schedule	constant	
weight decay	0	
gradient clipping	1.0	
ema decay	0.9999	
time sampler	$\text{logit}(t) \sim \mathcal{N}(\mu, \sigma^2)$, $\mu = -0.8$, $\sigma = 0.8$	
noise scale	1.0	
clip of $(1-t)$ in division	0.05	
caption drop (for CFG)	0.1	
sampling		
ODE solver	Euler	
ODE steps	50	
time steps	linear in [0.0, 1.0]	
CFG scale (Ho & Salimans, 2022)	selected on GPIC Validation-50K	
weights	EMA	

B.1 Implementation of Perceptual Loss Baselines

We follow PixelGen (Ma et al., 2026) in implementing our LPIPS, DINOv2, and SigLIP 2 perceptual loss baselines. Each baseline uses only one perceptual loss, and we disable REPA feature alignment. We train JiT-T2I-g models from scratch with the configuration from Table B.3.

LPIPS. We use the standard pretrained VGG16 LPIPS metric (Zhang et al., 2018).

DINOv2 and SigLIP 2. PixelGen uses a P-DINO perceptual loss using the DINOv2 ViT-B/14 backbone, and we follow the same methodology to construct P-SigLIP with a SigLIP 2 ViT-B/16@256px backbone. For a predicted $\hat{x}$ and target x, we compare these images in DINOv2 and SigLIP 2 feature spaces. Let f_p denote the transformation from an input image in pixel space to its pth patch token. For SigLIP 2, these are the features prior to the final pooling head in the vision encoder. For both the DINOv2 and SigLIP 2 backbones, we have $P = 256$ tokens total and $D = 768$ dimensionality of each token. We then compare $f_p(\hat{x})$ and $f_p(x)$ with one of the following distance functions.

$$d_{\cos}(\hat{x}, x) = \frac{1}{P} \sum_{p=1}^{P} \left(1 - \frac{f_p(\hat{x})^\top f_p(x)}{\|f_p(\hat{x})\|_2 \, \|f_p(x)\|_2} \right),$$

$$d_2(\hat{x}, x) = \frac{1}{PD} \sum_{p=1}^{P} \|f_p(\hat{x}) - f_p(x)\|_2^2 .$$

Following PixelGen, we use cosine dissimilarity when combining either feature loss with the pixel loss. For feature-space-only training, we use d_2.

Aggregate form and time gating. For pixel + perceptual loss baselines, the loss function takes the form

$$\mathcal{L} = \mathcal{L}_{\text{Pixel}} + \mathbf{1}_{t \geq \tau} \lambda \mathcal{L}_{\text{Perceptual}},$$

where $\mathbf{1}_{t \geq \tau}$ is a simple time-based gate, turning off perceptual supervision in the high-noise regime. We use PixelGen's threshold of $\tau = 0.3$. For LPIPS, $\lambda = 0.1$, and for DINOv2 and SigLIP 2, $\lambda = 0.01$. When we train solely with the perceptual loss and without any pixel loss term, we set $\lambda = 1$ and omit the $\mathcal{L}_{\text{Pixel}}$ term and time gate entirely.

C Noise Level and Loss Weight

We select the corruption level used to train LILIE and the weight of the learned loss using GPIC Validation-50K for evaluation. We train LILIE-S encoders with $\sigma \in \{0, 0.25, 0.5, 0.75, 1\}$. For each encoder, we train JiT-T2I-XL generators with mixing weights $\lambda \in \{0.5, 1, 2\}$ and with the LILIE objective. We compare generators at 200K training steps. Each model uses 50 Euler steps, and its CFG scale is selected from the grid $\{5.50, 6.25, 7.00, 7.75, 8.50\}$ using gFD on 50K validation samples. The selected scales appear in parentheses in Table C.1.

Among the mixed objectives, the setting $\sigma = 0.25$ and $\lambda = 2$ gives the lowest validation gFD. We use this as the default setting for our 1-epoch GPIC Full training runs. We notice that encoders trained without noise result in degenerate generation quality, so corruption in the training procedure is required for learning a representation that is useful to supervise generation. Increasing noise beyond 0.25 does not improve the best mixed result.

Table C.1: **Noise level and loss weight.** gFD on GPIC Validation-50K with JiT-T2I-XL at 200K training steps. The selected CFG scale is in parentheses. The Pixel baseline scores 114.90 at CFG 7.75. Bold marks the lowest gFD in the sweep.

σ	$\lambda = 0.5$	$\lambda = 1$	$\lambda = 2$	LILIE-only
0	1832.52 (8.50)	1854.12 (8.50)	2165.24 (8.50)	1502.33 (8.50)
0.25	97.12 (6.25)	92.60 (7.00)	**92.07** (6.25)	96.58 (6.25)
0.5	124.30 (7.75)	125.55 (7.75)	124.30 (7.00)	96.72 (6.25)
0.75	118.81 (7.75)	110.90 (7.75)	101.66 (7.00)	100.58 (7.00)
1.0	103.32 (7.00)	121.37 (7.75)	116.73 (7.00)	106.44 (7.75)

D Model Size and Data Ablations

We expand on Tables 2 and 3 to include all metrics. Table D.1 expands Table 2, and Table D.2 expands Table 3. All generators are evaluated at 200K training steps with 50 Euler steps. We select CFG scale via the same method described in Section C. All quantitative results use GPIC Test-50K prompts. Lower is better for gFD, its components, and MMD. Higher is better for precision, recall, and coverage. Values near 1 are ideal for density.

Table D.1: **Full model-size ablation results.** The LILIE encoders are trained on 100M images. TARFlow configuration is hidden dimension/blocks/layers (H/K/L) per block. Our primary setting is highlighted.

Generator	Encoder	H/K/L	Params	CFG	gFD ↓	gFD_μ ↓	gFD_Σ ↓	Precision ↑	Recall ↑	Density → 1	Coverage ↑	MMD ↓
JiT-T2I-XL	Pixel	–	0	7.75	114.59	22.07	92.52	0.924	0.696	0.940	0.870	0.082
	LILIE-μ	128/2/2	1.5M	7.00	96.11	17.98	78.14	0.931	0.760	0.957	0.896	0.066
	LILIE-T	192/4/4	9.1M	6.25	92.66	17.58	75.07	0.931	0.763	0.972	0.902	0.065
	LILIE-S	384/4/4	32M	6.25	**91.79**	17.39	74.40	0.928	0.762	0.973	0.905	0.064
JiT-T2I-g	Pixel	–	0	7.00	101.66	18.52	83.14	0.922	0.737	0.934	0.880	0.069
	LILIE-μ	128/2/2	1.5M	6.25	84.66	14.11	70.55	0.934	0.788	0.972	0.915	0.052
	LILIE-T	192/4/4	9.1M	6.25	80.87	13.43	67.44	0.937	0.792	0.982	0.910	0.050
	LILIE-S	384/4/4	32M	6.25	**76.59**	12.89	63.70	0.941	0.805	0.989	0.917	0.047

Table D.2: **Full training-data ablation results.** The data budget counts images used to train the LILIE-S flow (32M parameters); generator training is held fixed. We use $\sigma = 0.25$ and $\lambda = 2$ throughout both ablations. The 100M rows are the same LILIE-S configurations as in Table D.1 and are highlighted as our primary configuration.

Generator	# training images	CFG	gFD ↓	gFD_μ ↓	gFD_Σ ↓	Precision ↑	Recall ↑	Density → 1	Coverage ↑	MMD ↓
JiT-T2I-XL	10M	7.00	97.79	18.58	79.22	0.927	0.751	0.950	0.895	0.068
	30M	6.25	93.52	17.67	75.85	0.930	0.766	0.969	0.906	0.065
	100M	6.25	**91.79**	17.39	74.40	0.928	0.762	0.973	0.905	0.064
JiT-T2I-g	10M	7.00	83.01	13.58	69.44	0.934	0.788	0.983	0.914	0.051
	30M	7.00	81.18	13.34	67.85	0.934	0.789	0.983	0.915	0.050
	100M	6.25	**76.59**	12.89	63.70	0.941	0.805	0.989	0.917	0.047

E LIMITATIONS

We perform all of our experiments on a single corpus, resolution, and generator family, namely GPIC at 256×256 with JiT-T2I. Additionally, training the encoder requires captions and a VLM, and we study only the SmolVLM-256M backbone. A larger VLM may help construct a better feature space, but it comes at an increase in encoder training cost. Our largest version of LILIE is a 32M parameter encoder, and we do not experiment on scales closer to DINOv2 and SigLIP 2 base model sizes (86M parameters).

F CAPTIONING PROMPTS

GPIC consists of captions in four length categories (`tag`, `short`, `medium`, `long`), and LILIE training pairs each caption type with its own prompt for SmolVLM-256M. Figures F.1–F.4 list the four pairs. These are condensed versions of prompts used for GPIC corpus captioning (Chandrasegaran et al., 2026).

```
[tag]
system:
  Your goal is to write a compressed keyword-style caption (unordered tags)
  for the given image. The output must be a short, unordered list of tags
  describing the main visible content.
user:
  Write a keyword-style caption (tag-style) for the image shown.
```

Figure F.1: Prompt given to SmolVLM-256M during LILIE training for GPIC `tag`-length captions.

```
[short]
system:
  Your goal is to write a short caption (1-2 sentences) that describes the
  given image in simple language that a middle school student can
  understand. The caption must be enough to identify the main subject and
  the basic setting/action.
user:
  Write a short caption (1-2 sentences) for the image shown.
```

Figure F.2: Prompt given to SmolVLM-256M during LILIE training for GPIC `short`-length captions.

```
[medium]
system:
  Your goal is to write a medium-length caption (2-3 sentences) that
  describes the given image in simple language that a middle school student
  can understand. The caption must be enough to identify the main subject
  and the basic setting/action.
user:
  Please write a caption (2-3 sentences) for the image shown.
```

Figure F.3: Prompt given to SmolVLM-256M during LILIE training for GPIC `medium`-length captions.

```
[long]
system:
  Your goal is to write a long, highly detailed caption (5-7 sentences)
  that describes the given image in simple language that a middle school
  student can understand. The caption should include enough detail for them
  to understand what appears in the foreground and background and picture
  how the scene is arranged.
user:
  Please write a caption (5-7 sentences) for the image shown.
```

Figure F.4: Prompt given to SmolVLM-256M during LILIE training for GPIC `long`-length captions.

G SAMPLES

We provide selected sampling trajectories for JiT-T2I-g models trained with Pixel, LILIE, and Pixel+LILIE in Figure G.1. We provide random samples from the same models in Figures G.2 to G.6. All samples are taken from each model's optimal CFG scale, listed in Table B.1.

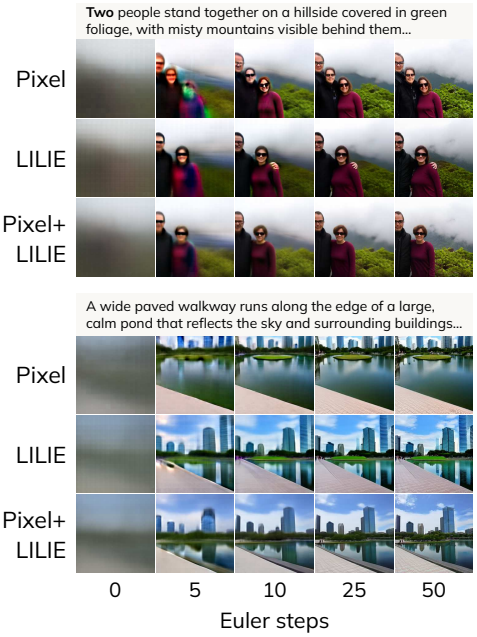

Figure G.1: **Selected sampling trajectories over 50 steps.** We fix a 50 step sampling budget and show clean image predictions at intermediate Euler steps 0, 5, 10, and 25, followed by the final image at step 50. Each example uses the same prompt and initial noise across methods. The top sample illustrates a counting error in the Pixel baseline, which is fixed in the models trained with LILIE.

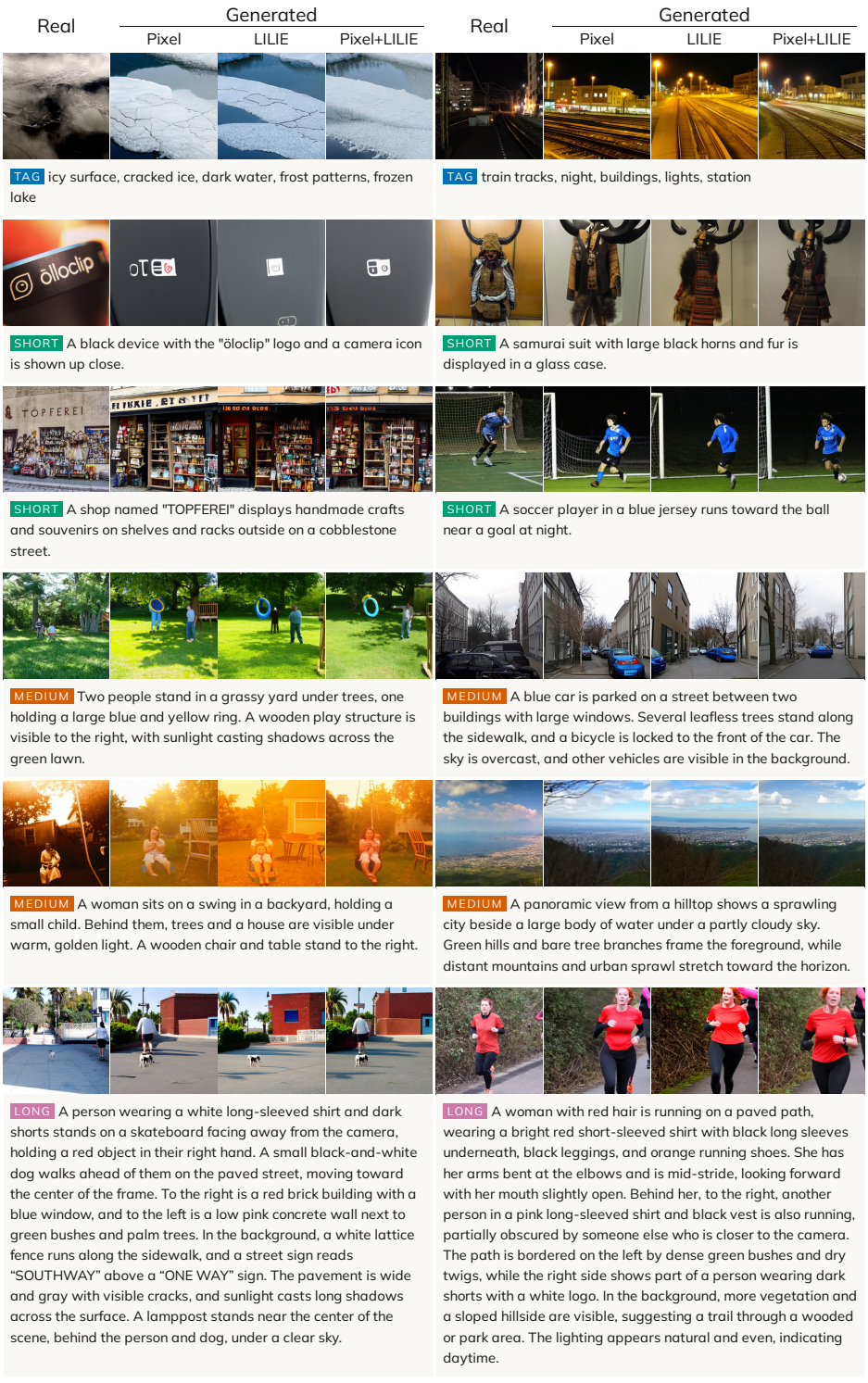

Figure G.2: Random samples from the held-out GPIC Test-50K. Each example shows the real image from the GPIC corpus on the left, followed by the three generated samples from JiT-T2I-g models. The first generated sample is from the Pixel baseline, and the following two are from LILIE and Pixel+LILIE.

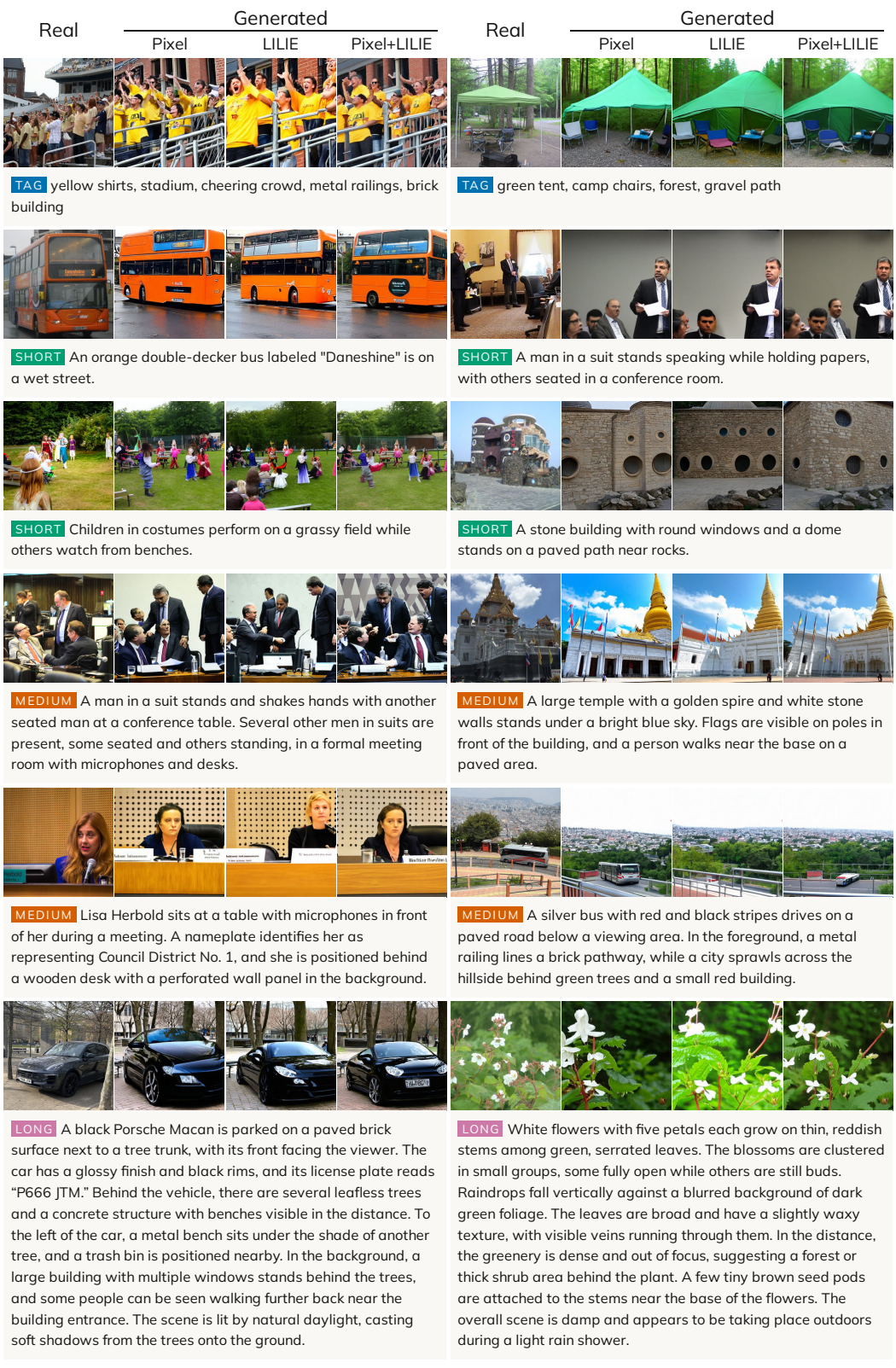

Figure G.3: Random samples from the held-out GPIC Test-50K. Each example shows the real image from the GPIC corpus on the left, followed by the three generated samples from JiT-T2I-g models. The first generated sample is from the Pixel baseline, and the following two are from LILIE and Pixel+LILIE.

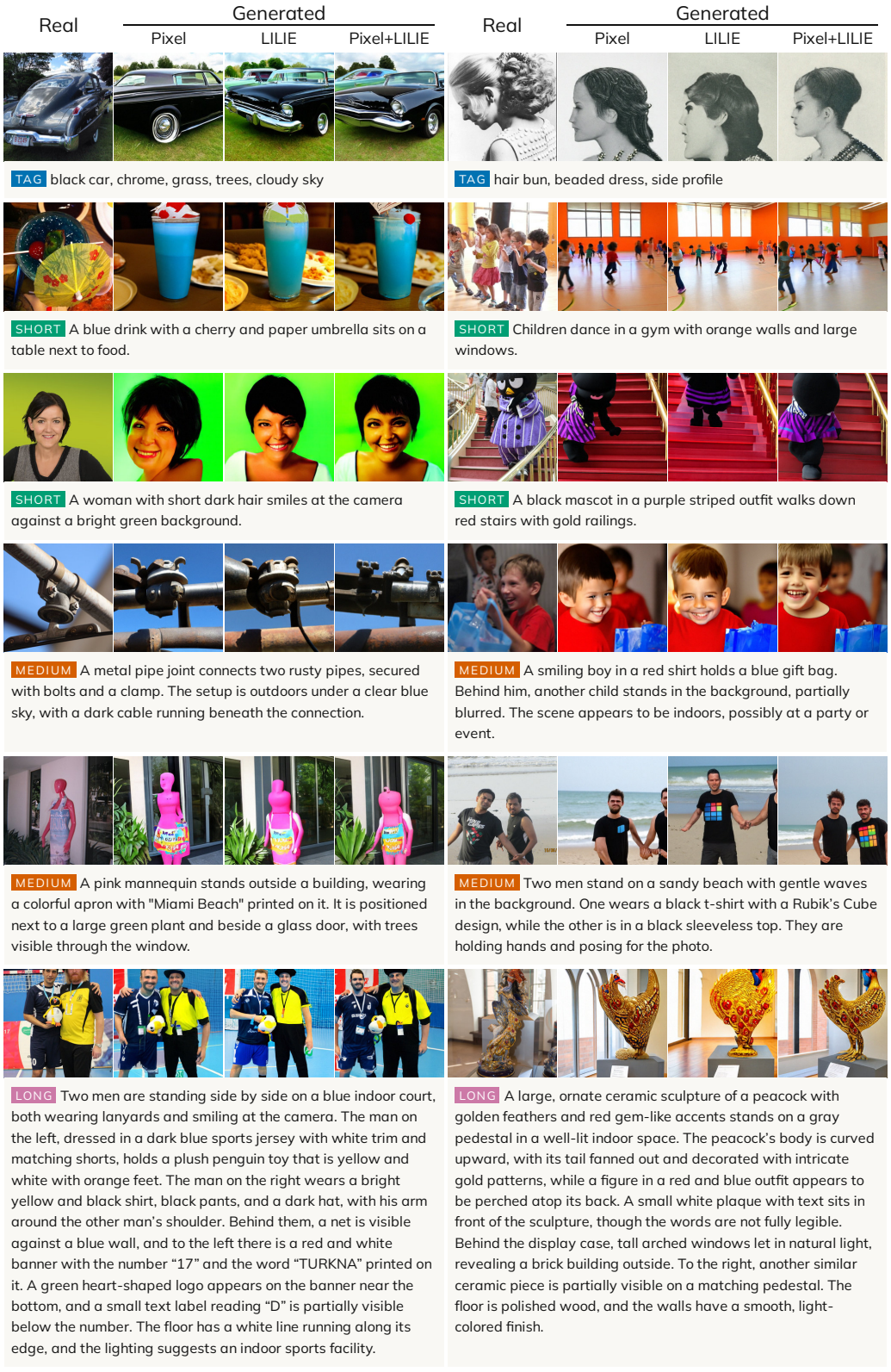

Figure G.4: Random samples from the held-out GPIC Test-50K. Each example shows the real image from the GPIC corpus on the left, followed by the three generated samples from JiT-T2I-g models. The first generated sample is from the Pixel baseline, and the following two are from LILIE and Pixel+LILIE.



TAG three women, wooden bench, colorful buildings, paved ground, winter trees

TAG horse, rider, jump, sand, trees



SHORT Gray stone sculptures sit on green grass in front of a large white building with a clock tower.

SHORT A woman and a child stand on the porch steps of a small cabin in a forest.



SHORT A man and a woman stand on a city street at night, smiling near a storefront window.

SHORT A blonde woman with sunglasses on her head stands in a field of blue hydrangeas.



MEDIUM A person stands in the dark, creating glowing red circular patterns with a sparkler. Another person is visible to the right, standing nearby. The scene appears to be outdoors at night.

MEDIUM A group of people walk and stand in a large, bright indoor space with wooden ceilings. Some individuals carry bags or wear lanyards, and tables with items are visible on the right side. The area appears to be a convention center or exhibition hall.



MEDIUM Two people in suits shake hands in a conference room. A woman with short brown hair and a man wearing glasses stand facing each other, surrounded by seated attendees and desks.

MEDIUM A person in a colorful orange and yellow costume plays table tennis on a blue table. Another person stands nearby at a table with a computer, while balloons and event signage are visible in the background.



LONG A young woman with shoulder-length brown hair and light brown highlights is sitting and looking slightly toward the camera with a subtle smile. She wears a blue short-sleeved button-up shirt that has a textured fabric and a small white lace detail at the neckline. The shirt has faint lettering on the left side of the chest, though it's not fully legible. Behind her is a solid reddish-brown textured wall that fills the background. Her arms are resting in front of her, and she appears to be seated in a chair, though only the edge of the chair is visible on the right side. The lighting is even and soft, highlighting her facial features and the details of her clothing.

LONG A woman in a white patterned top and dark pants stands on a green and red tennis court, holding a paddle with her right hand raised as if preparing to hit a yellow ball that is suspended in mid-air above her. To her left, the arm and shoulder of another person wearing a red shirt are visible, suggesting they are also playing. A white tennis net with black lettering along its edge stretches across the court, separating the players from the opposite side. Behind them, a tall chain-link fence runs along the perimeter, with large gray tarps covering the area beyond it, and trees with green leaves stand behind the fence. The court surface has painted lines and appears to be made of hardcourt material, and sunlight casts clear shadows of the players onto the ground. The woman's posture shows she is actively engaged in the game, with her body slightly turned toward the ball.

Figure G.5: Random samples from the held-out GPIC Test-50K. Each example shows the real image from the GPIC corpus on the left, followed by the three generated samples from JiT-T2I-g models. The first generated sample is from the Pixel baseline, and the following two are from LILIE and Pixel+LILIE.

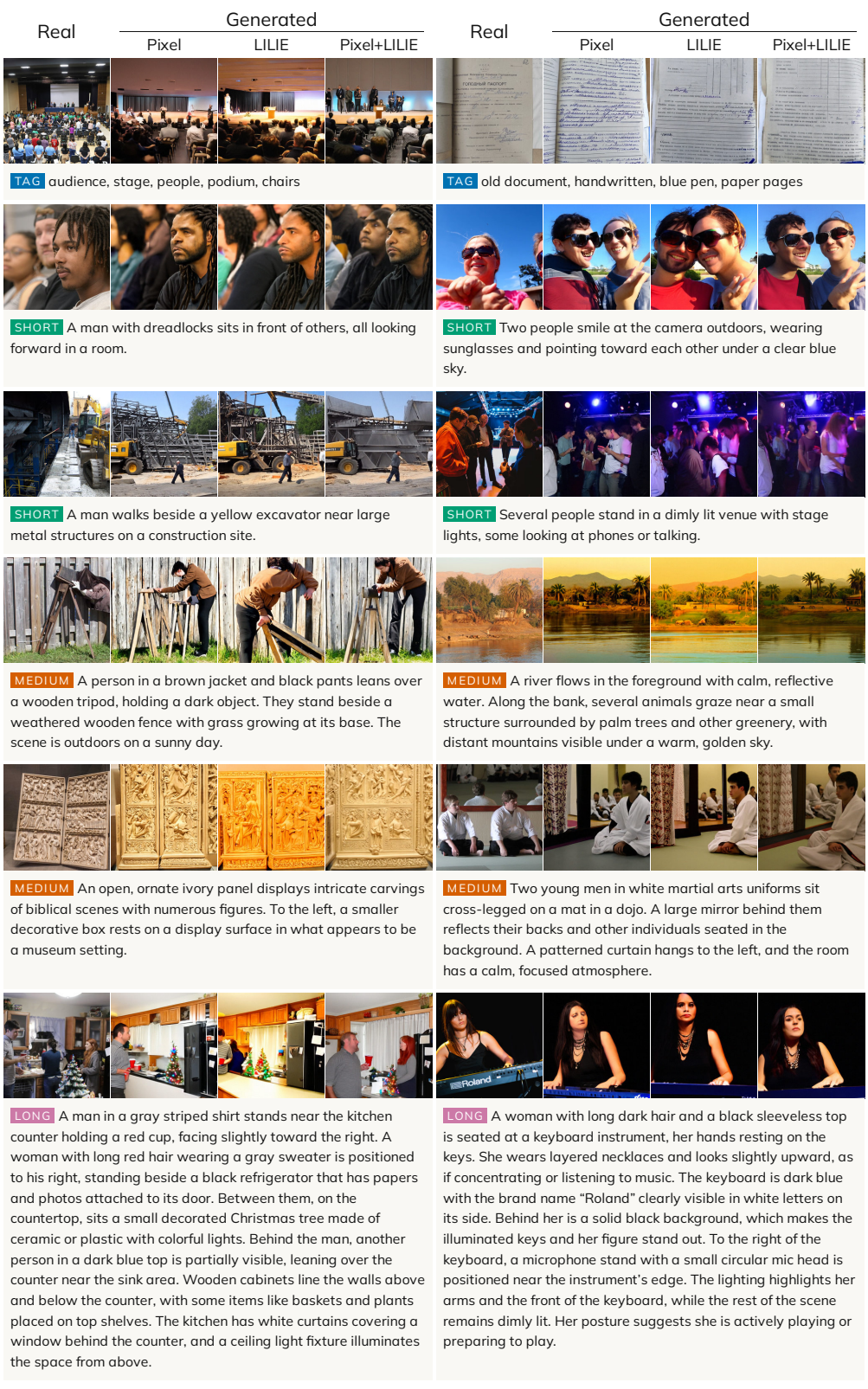

Figure G.6: Random samples from the held-out GPIC Test-50K. Each example shows the real image from the GPIC corpus on the left, followed by the three generated samples from JiT-T2I-g models. The first generated sample is from the Pixel baseline, and the following two are from LILIE and Pixel+LILIE.