# LEVEL-OF-TOKEN DIFFUSION

**Anonymous authors**
Paper under double-blind review




Figure 1: **Level-of-Token (LoT) Diffusion** allocates compute using a spatial token layout that specifies the spatially varying level of detail for efficient image and video generation. Derived from diverse spatial priors, token layouts guide detail allocation with speedups tied to the token budget.

## ABSTRACT

Image and video diffusion models allocate equal computation to every region, even when the intended scene calls for varying levels of detail. The spatial distribution of detail can often be anticipated before generation, indicating where computation can be reduced. We introduce *Level-of-Token (LoT) Diffusion*, a framework that turns this knowledge into an explicit multiresolution token layout (*Level-of-Token layout*) for adaptive and efficient generation. Tokens represent rectangular patches of varying sizes and shapes, allocating finer tokens where detail is needed and coarser tokens elsewhere. We adapt pretrained diffusion transformers to LoT layouts through a patch-wise asymmetric flow parametrization and embeddings for multiresolution tokens, preserving full-resolution flow prediction at every denoising step while processing only a reduced token sequence. LoT Diffusion enables layout-adaptive generation while preserving pretrained generative priors. We demonstrate LoT with layouts derived from semantic masks, bounding boxes, texture variance, and depth-of-field cues, as well as agentic plans. Across image and video generation, LoT offers favorable quality–efficiency tradeoffs, with significant speedups determined by the layout's token budget.



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 and video generative models produce remarkably high-quality content, yet generating high-resolution images and long videos remains computationally expensive. Much of this cost comes from diffusion transformers (Peebles & Xie, 2023) processing long sequences of spatial tokens through attention and linear layers at every denoising step. Reducing this cost begins with recognizing that not all regions require the same level of detail: intricate textures, faces, and text require fine detail, while clear skies, blurred backgrounds, and flat surfaces can be synthesized more coarsely. Pretrained models nevertheless allocate equal computation to all regions. Allocating resources according to the detail needed is well established in level-of-detail rendering (Clark, 1976), variable-rate shading (Vaidyanathan et al., 2014), and image compression (Shapiro, 1993). Whereas rendering and compression operate on existing scene or image content, generation can begin with a specified distribution of detail before the final pixels are available. Notably, such knowledge can come from the structure, description, or plan of the intended scene, which generative models increasingly support through richer prompting and explicit layout specification (OpenAI, 2026; Reve AI, 2026; Ideogram, 2026a). These observations motivate using prior knowledge of where detail matters to determine how to allocate compute for generation.

Ideally, a generative model would use this prior knowledge both to adapt where fine detail is concentrated and to reduce computation. Spatial conditioning methods (Zhang et al., 2023; Mou et al., 2024; Tan et al., 2025; Zhang et al., 2025a; Yang et al., 2023) guide generated content using spatial inputs, but retain full-resolution processing and often introduce additional computational cost. Efficiency methods exploit redundancy through either reusing intermediate computations (Liu et al., 2025a; Zou et al., 2024) or token reduction (Bolya & Hoffman, 2023; Lu et al., 2025; Xiao et al., 2026; Jeong et al., 2026; Kim et al., 2026; Wu et al., 2026; Chao et al., 2026), but do not utilize this adaptivity as a control mechanism. Our goal is to use prior knowledge to control where fine detail is concentrated and allocate computation accordingly, enabling both adaptive and efficient generation.

In this paper, we introduce *Level-of-Token (LoT) Diffusion*, in which an image or video is synthesized from a content-adaptive, multiresolution token layout (*Level-of-Token Layout*). Each token represents a rectangular spatial patch with variable extent, specified by its height and width, allowing important regions to be finely tokenized while less important ones are tokenized coarsely. Realizing this on pretrained diffusion transformers requires adapting models trained on uniform grids to heterogeneous token layouts. To this end, we develop a formulation that preserves full-resolution flow prediction at every denoising step while the transformer processes only a reduced token sequence. We achieve this through a patch-wise asymmetric flow parameterization (Chen et al., 2026), together with adaptations to token embeddings and positional encoding that account for various patch extents. Fine-tuning pretrained models within our simple framework enables efficient generation that adapts to the specified layout while leveraging their generative priors.

We demonstrate LoT across diverse layout sources, including semantic masks, bounding boxes, texture variance, and depth-of-field cues, as well as agentic plans. Generation speedup depends on the token budget specified by the layout, reaching up to $2.04\times$ for images and $3.53\times$ for videos on average relative to the full-resolution pretrained models. Across the evaluated budgets, LoT offers favorable tradeoffs between generation quality and efficiency compared with existing token-reduction methods. Qualitative results further show how generated content adapts to the specified token layout. More results are in the Supplementary Website at `https://iclr-submission7014.github.io/supplementary/`.

To summarize, our main contributions are:

- We introduce *Level-of-Token Diffusion*, a framework for efficient image and video generation with token layouts that adaptively vary patch size and shape across an image or video.
- We adapt pretrained DiTs through patch-wise asymmetric flow and embeddings for multiresolution tokens, enabling full-resolution flow prediction at every step from a reduced token sequence.
- We demonstrate favorable quality and efficiency tradeoffs across image and video generation, using layouts derived from scene cues and agentic plans.

## 2 Related Work

**Methods for efficient generation.** A large body of work improves the efficiency of diffusion models along several axes. Along the spatial axis, token merging and pruning techniques identify redundant tokens and merge or discard them based on predefined heuristics or learned criteria (Bolya & Hoffman, 2023; Kim et al., 2024; Lu et al., 2025; Haurum et al., 2024; Wu et al., 2025; Lee et al., 2025; Fang et al., 2025), while sparse attention mechanisms exploit sparse patterns in the attention maps and use block-wise computation to reduce the quadratic cost of self-attention (Li et al., 2025; Xia et al., 2025; Zhan et al., 2025; Katharopoulos et al., 2020; Wang et al., 2020; Choromanski et al., 2020; Beltagy et al., 2020). Along the timestep axis, feature caching methods reuse intermediate features across denoising steps to avoid redundant computation (Liu et al., 2025a; Zou et al., 2024; Liu et al., 2025b; Ma et al., 2024), whereas distillation and sampler-based approaches reduce the number of denoising steps themselves, either by training a student to approximate a pretrained teacher (Yin et al., 2024b;a; 2025; Luo et al., 2023; Wang et al., 2024; Salimans & Ho, 2022) or by deriving faster solvers and consistency-based samplers (Song et al., 2020; 2023). Other progressive-upsampling methods grow the spatial resolution over the course of generation to save computation in early stages (Jeong et al., 2026; Xiao et al., 2026). However, these methods treat spatial regions equally, operating along denoising-trajectory or attention-level axes independent of how details are spatially distributed, and thus leave the prior knowledge of *where* content requires computation unexploited.

Content-adaptive tokenization is an orthogonal and complementary axis that exploits exactly this prior, accelerating generation by allocating compute based on the content. However, existing approaches address only part of the problem: some are spatially adaptive but with no or limited user control (Chao et al., 2026; Haridas et al., 2026; Zhao et al., 2025b; Zamfir et al., 2026; Zhao et al., 2025a), others adapt the token budget without redistributing it across space (Yan et al., 2025; Bachmann et al., 2025; Ye et al., 2025; Shen et al., 2025; Duggal et al., 2024; Haji-Ali et al., 2026), and still others achieve spatial adaptivity but require training from scratch and are incompatible with pretrained diffusion models (Zhang et al., 2025c; Jia et al., 2025; Aasan et al., 2026). In contrast, our method is spatially adaptive and compatible with various pretrained models, redistributing tokens non-uniformly according to an intended layout while additionally exposing user control over the allocation. Most importantly, our method greatly accelerates image and video generation while remaining compatible with the other efficiency techniques above.

**Layout control for diffusion models.** A large body of work augments diffusion models with explicit control over object placement and scene composition. Prior conditioning-based methods attach trainable modules onto a frozen backbone, injecting spatial signals such as depth, segmentation masks, or bounding boxes, so that users can dictate where content appears (Zhang et al., 2023; Mou et al., 2024). More recently, another line of work makes layout a direct input to the model rather than a separate branch, encoding it into tokens that enter the denoising sequence (Tan et al., 2025; Xiao et al., 2025; Zhang et al., 2025a) or expressing it as structured text in the prompt itself (Yang et al., 2023). Most strikingly, the strongest deployed systems have made layout their primary interface: Ideogram 4.0 is trained exclusively on structured JSON captions carrying a description and a bounding box per element (Ideogram, 2026b), and Reve 2 derives a hierarchical layout from the input text prompt before rendering from it (Reve AI, 2026). However, in all these methods the layout merely decides *what* appears *where*, never how much compute is allocated to it. A region the layout identifies as empty sky is still generated on the same dense token grid as detailed typography. In contrast, our method uses this prior knowledge of where detail matters to allocate computation spatially, enabling content-adaptive generation with fewer tokens. Semantic masks and depth-of-field cues identify regions that can be generated more coarsely, allowing LoT to reduce computation there while preserving finer tokenization where detail is needed.

## 3 Preliminaries

We define level-of-token layouts and review flow-based image and video generation. We denote the full-resolution latent grid by $\boldsymbol{x} \in \mathbb{R}^{H_0 \times W_0 \times D'}$, where $D'$ is the channel dimension. The concepts naturally extend to videos by adding a temporal axis.

### 3.1 LEVEL-OF-TOKEN LAYOUT

Existing image and video generative models represent visual content as sequences of latent tokens. A pretrained DiT uses a fixed size $(p_h, p_w)$ to rearrange $\boldsymbol{x}$ into a *uniform token layout*:

$$\boldsymbol{x}_U = \mathrm{Patchify}(\boldsymbol{x}) \in \mathbb{R}^{H \times W \times D}, \qquad H = H_0/p_h, \quad W = W_0/p_w, \quad D = p_h p_w D' \tag{1}$$

where Patchify flattens each non-overlapping $p_h \times p_w$ latent patch into a $D$-dimensional token.

Motivated by level-of-detail geometry, variable-rate shading, and adaptive image and video representations (Clark, 1976; Vaidyanathan et al., 2014; Shapiro, 1993; Ye et al., 2025), which allocate more budget to regions that are visually important or contain high-frequency textures, we generalize uniform token layouts to *level-of-token (LoT) layouts*. Formally, a LoT layout allows the patch's shape to vary spatially based on a partition $\mathcal{P} = \{R_1, \ldots, R_L\}$ of the uniform token grid $[0, H) \times [0, W)$ into $L$ disjoint axis-aligned tokens:

$$\begin{aligned} R_i &= [u_i,\, u_i + e_{h,i}) \times [v_i,\, v_i + e_{w,i}), \\ 0 \le u_i < u_i + e_{h,i} \le H, &\qquad 0 \le v_i < v_i + e_{w,i} \le W, \end{aligned} \tag{2}$$

where $R_i$'s are pair-wise disjoint and cover the $H \times W$ token grid. We denote the $i$-th token's extent as $\boldsymbol{e}_i = (e_{h,i}, e_{w,i})$ and define $D_{\boldsymbol{e}_i} := e_{h,i} e_{w,i} D$. Furthermore, let $\boldsymbol{x}^{(i)} \in \mathbb{R}^{D_{\boldsymbol{e}_i}}$ denote the concatenation of all uniform tokens within region $R_i$ and let $\bar{\boldsymbol{x}}_{R_i} \in \mathbb{R}^D$ denote the single token in the LoT layout representing the spatial region of this patch. We will discuss the mapping between $\boldsymbol{x}^{(i)}$ and $\bar{\boldsymbol{x}}_{R_i}$ in Sec. 4.1.

The partition induces the level-of-token sequence $\bar{\boldsymbol{x}}_{\mathcal{P}} = (\bar{\boldsymbol{x}}_{R_1}, \ldots, \bar{\boldsymbol{x}}_{R_L}) \in \mathbb{R}^{L \times D}$ (see Fig. 2). We refer to its $i$-th token's top-left coordinate as $\boldsymbol{p}_i = (u_i, v_i)$. The same notation extends to videos by adding a temporal axis where a latent grid of size $T_0 \times H_0 \times W_0$ is patchified into a $T \times H \times W$ token grid, with $T = T_0/p_t$ and $D = p_t p_h p_w D'$. Throughout the paper we refer to token compression relative to the pretrained DiT as $HW/L$ for images and $THW/L$ for videos.

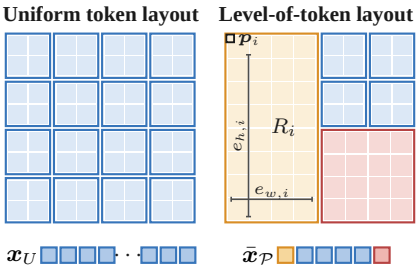

Figure 2: **Token layout notation.**

### 3.2 FLOW-BASED IMAGE AND VIDEO GENERATION

**Flow matching.** Flow matching defines a probability path between the data distribution and a standard Gaussian noise distribution. This path couples a data sample $\boldsymbol{x}_0 \in \mathbb{R}^{H_0 \times W_0 \times D'}$ with a noise sample $\boldsymbol{x}_1 := \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ using the time schedules $\alpha_t = 1 - t$ and $\sigma_t = t$ for $t \in [0, 1]$. The state and velocity remain on this full-resolution latent grid. We define the probability path and its velocity for any $t \in [0, 1]$ as

$$\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\varepsilon}, \quad \boldsymbol{u} = \boldsymbol{\varepsilon} - \boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \boldsymbol{x}_0}{\sigma_t}. \tag{3}$$

The flow model $v_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$ is trained to predict the flow velocity $\boldsymbol{u}$ given $\boldsymbol{x}_t$, time step $t$, and optional text conditioning $\boldsymbol{c}$, by optimizing the following loss:

$$\mathcal{L}_{\text{flow}}(\theta) \;=\; \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{c}} \Big[ \big\| v_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) - \boldsymbol{u} \big\|_2^2 \Big]. \tag{4}$$

After training, generation solves the ODE $\frac{d\boldsymbol{x}_t}{dt} = v_\theta$ with $\boldsymbol{x}_1 = \boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ from $t = 1$ to 0.

**Asymmetric flow matching.** Recently, asymmetric flow matching (AsymFlow) (Chen et al., 2026) proposes to fine-tune pretrained image latent flow matching models by adopting an asymmetric parameterization where the model learns to predict the flow velocity from low-rank noise to full-rank data $\boldsymbol{u}_{\text{A}} = \boldsymbol{P}\boldsymbol{\varepsilon} - \boldsymbol{x}$ where $\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}^T \in \mathbb{R}^{D \times D}$, $\boldsymbol{A} \in \mathbb{R}^{D \times d}$, $D \ge d$ is a fitted orthonormal projector. The full-rank velocity can be recovered by

$$\boldsymbol{u} = \boldsymbol{P}\boldsymbol{u}_{\text{A}} + (\boldsymbol{I} - \boldsymbol{P}) \frac{\boldsymbol{x}_t + u_{\text{A}}}{\sigma_t}. \tag{5}$$

AsymFlow generates images directly in pixel space at a computational cost similar to that of the original latent flow models. Inspired by this, Level-of-Token generates a full-resolution latent grid from a level-of-token layout, enabling efficient, content-aware generation.

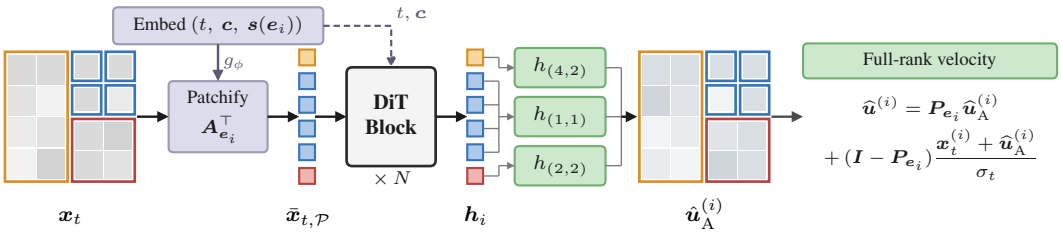


Figure 3: **Level-of-Token Diffusion pipeline.** LoT Diffusion maps a full-resolution token grid $\boldsymbol{x}_t$ to a sequence of reduced tokens specified by the LoT layout. The DiT blocks process these tokens conditioned on token shapes $\boldsymbol{s}$, timestep $t$, and text $\boldsymbol{c}$. Extent-dependent heads $h_{\boldsymbol{e}_i}$ restore the asymmetric velocity $\hat{\boldsymbol{u}}$, which is converted into the full-rank velocity for flow-matching training.

**Diffusion Transformers (DiT) and RoPE.** Under a uniform token layout in Eq. 1, Diffusion Transformers (DiT) (Peebles & Xie, 2023) embed the $HW$ tokens of $\boldsymbol{x}_t$ into the hidden dimension $d_{\text{model}}$ and process them with transformer blocks. The output head predicts full-resolution velocities, which are rearranged into a full-resolution velocity field of size $H_0 \times W_0 \times D'$. Because self-attention is permutation equivariant, models such as FLUX.2 and Wan encode each token's spatial position $\boldsymbol{p}_i = (i_1, \ldots, i_d)$ on the uniform token grid using multi-axis RoPE (Su et al., 2024). For query and key vectors $\boldsymbol{q}_i, \boldsymbol{k}_i \in \mathbb{R}^{d_{\text{h}}}$, where $d_{\text{h}}$ is the attention-head dimension, RoPE splits the channels across the $d$ spatial axes and applies

$$\widetilde{\boldsymbol{q}}_i = \mathbf{R}_{\text{axial}}(\boldsymbol{p}_i)\boldsymbol{q}_i, \qquad \widetilde{\boldsymbol{k}}_i = \mathbf{R}_{\text{axial}}(\boldsymbol{p}_i)\boldsymbol{k}_i, \qquad \mathbf{R}_{\text{axial}}(\boldsymbol{p}_i) = \bigoplus_{a=1}^{d} \mathbf{R}^{(a)}(i_a). \tag{6}$$

where $\mathbf{R}^{(a)}(i_a)$ denotes the standard RoPE rotation (Su et al., 2024) applied to the channel block assigned to spatial axis $a$. Usually, RoPE assigns a spatial position to each token and does not encode the size of its spatial support. A DiT operating on a level-of-token layout must therefore place all tokens in a consistent coordinate system and further encode their extents separately.

## 4 Method

Given a LoT layout defined by $\mathcal{P} = \{R_1, \ldots, R_L\}$, our goal is to reduce the number of tokens processed by the DiT while remaining compatible with pretrained models. Specifically, we use patch-wise asymmetric flow matching (Sec. 4.1) and introduce Level-of-Token DiT (Sec. 4.2) that minimally modifies the DiT architecture to handle LoT layouts. We present our method for image generation, with a natural extension to video through per-frame LoT layouts. Fig. 3 shows our overall pipeline.

### 4.1 Flow Matching with Level-of-Tokens

A LoT layout reduces the DiT sequence length from $HW$ to $L$, but adapting a pretrained latent diffusion model to this reduced sequence presents two challenges. First, the pretrained VAE decoder requires a full-resolution latent grid of size $H_0 \times W_0 \times D'$. Second, the pretrained DiT predicts flow on the same sequence it receives, so directly applying it to the reduced sequence would only produce $L$ token-level flow predictions, rather than the full-resolution flow needed for denoising.

**Patch-wise asymmetric flow.** Inspired by AsymFlow (Chen et al., 2026), we use compressed LoT tokens to predict a dense asymmetric velocity field. For each extent $\boldsymbol{e} = (e_h, e_w)$, we fit a semi-orthonormal basis $\boldsymbol{A}_{\boldsymbol{e}} \in \mathbb{R}^{D_{\boldsymbol{e}} \times D}$ using orthogonal Procrustes (Schönemann, 1966) between aligned dense patches and extent-matched multi-scale VAE tokens (see Sec. A.2 for fitting details). Given the noisy latent grid $\boldsymbol{x}_t$ and a LoT layout $\mathcal{P}$, we project each region's patch onto its corresponding basis and assemble the resulting token sequence:

$$\bar{\boldsymbol{x}}_{t,R_i} = \boldsymbol{A}_{\boldsymbol{e}_i}^\top \boldsymbol{x}_t^{(i)} \in \mathbb{R}^D, \qquad \bar{\boldsymbol{x}}_{t,\mathcal{P}} = \left(\bar{\boldsymbol{x}}_{t,R_1}, \ldots, \bar{\boldsymbol{x}}_{t,R_L}\right) \in \mathbb{R}^{L \times D}. \tag{7}$$

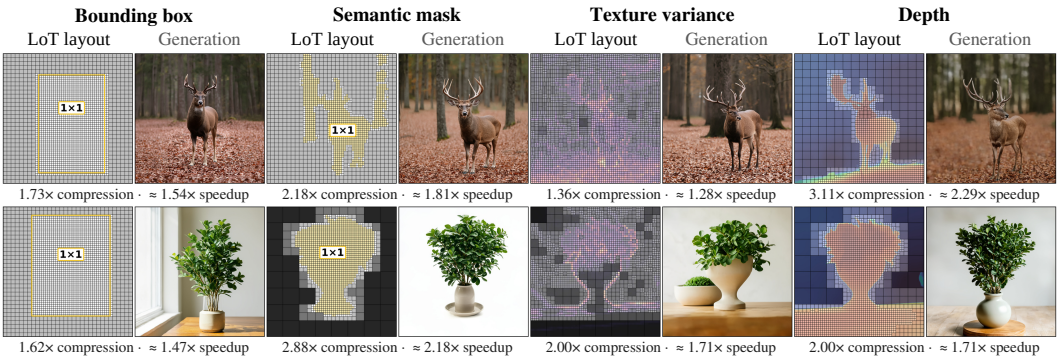


Figure 4: **Image generation with different layouts.** We generate images using arbitrary LoT layouts derived from signals that indicate spatial importance, including bounding boxes, semantic masks, texture variance, or depth maps, allocating finer tokens to regions that require detail.

Thus, the DiT input dimension is the same as in the pretrained model, while its sequence length is determined by $\mathcal{P}$. For each region $R_i$ with extent $\boldsymbol{e}_i$, we further define $\boldsymbol{P}_{\boldsymbol{e}_i} = \boldsymbol{A}_{\boldsymbol{e}_i}\boldsymbol{A}_{\boldsymbol{e}_i}^{\top}$ to be the extent's projector. Then the dense asymmetric velocity target for $R_i$ becomes

$$\boldsymbol{u}_{\mathrm{A}}^{(i)} = \boldsymbol{P}_{\boldsymbol{e}_i}\boldsymbol{\varepsilon}^{(i)} - \boldsymbol{x}_0^{(i)}. \tag{8}$$

Given a prediction $\widehat{\boldsymbol{u}}_{\mathrm{A}}^{(i)}$, we recover a full-rank velocity for the patch as

$$\widehat{\boldsymbol{u}}^{(i)} = \boldsymbol{P}_{\boldsymbol{e}_i}\widehat{\boldsymbol{u}}_{\mathrm{A}}^{(i)} + (\boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{e}_i})\frac{\boldsymbol{x}_t^{(i)} + \widehat{\boldsymbol{u}}_{\mathrm{A}}^{(i)}}{\sigma_t}. \tag{9}$$

In practice, we clamp the denominator to $\max(\sigma_t, 10^{-6})$ for numerical stability. For $t > 0$, Eq. 9 recovers the dense flow $\boldsymbol{\varepsilon} - \boldsymbol{x}_0$ when the asymmetric target is predicted exactly. We convert the recovered velocity into a clean-data estimate and train using the loss in Eq. 17.

**Extent-dependent output heads.** A token of extent $\boldsymbol{e}$ must predict an asymmetric velocity for all $e_h e_w$ tokens in its patch. We therefore use one linear output head per supported extent $\boldsymbol{e}$,

$$h_{\boldsymbol{e}} : \mathbb{R}^{d_{\mathrm{model}}} \to \mathbb{R}^{D_{\boldsymbol{e}}}, \qquad \widehat{\boldsymbol{u}}_{\mathrm{A}}^{(i)} = h_{\boldsymbol{e}_i}(\boldsymbol{h}_i) = \boldsymbol{W}_{\boldsymbol{e}}\boldsymbol{h}_i, \qquad \boldsymbol{W}_{\boldsymbol{e}} \in \mathbb{R}^{D_{\boldsymbol{e}} \times d_{\mathrm{model}}}, \tag{10}$$

where $d_{\mathrm{model}}$ is the DiT's hidden dimension. We apply Eq. 9 patch-wise to obtain a full-resolution velocity field, i.e., $\boldsymbol{u} \in \mathbb{R}^{H_0 \times W_0 \times D'}$. Furthermore, we initialize each output head as $\boldsymbol{W}_{\boldsymbol{e}} = \boldsymbol{A}_{\boldsymbol{e}}\boldsymbol{W}_{\mathrm{out}}$, where $\boldsymbol{W}_{\mathrm{out}} \in \mathbb{R}^{D \times d_{\mathrm{model}}}$ is the pretrained output head.

**Extent-dependent data scaling.** When fitting $\boldsymbol{A}_{\boldsymbol{e}}$, we also estimate a scalar $s_{\boldsymbol{e}}$ by matching projected and reference token root-mean-square norms. AsymFlow (Chen et al., 2026) compensates for this scale mismatch through input/output rescaling and timestep calibration. However, applying that calibration per extent would introduce heterogeneous per-token timesteps, departing from the pretrained DiT's shared-timestep conditioning. Therefore, we instead calibrate at the data level, namely we divide each clean dense patch by $s_{\boldsymbol{e}}$ before adding noise and undo the scaling before VAE decoding. Fitting and scaling details are provided in Sec. A.2 of the supplementary material.

### 4.2 Level-of-Token DiT

The projected tokens in Eq. 7 allows us to reuse the DiT's pretrained weights. We further introduce extent-dependent input heads and initialize each by composing the pretrained input projection with $\boldsymbol{A}_{\boldsymbol{e}}^{\top}$. To inform the DiT of the Level-of-Token layout, we introduce patch-shape conditioning and finest-grid-aligned RoPE to encode each LoT token's spatial extent and location, respectively.

**Patch-shape conditioning.** To represent a token with extent $\boldsymbol{e}_i = (e_{h,i}, e_{w,i})$, we encode its per-axis size, area, and aspect ratio as

$$\boldsymbol{s}(\boldsymbol{e}_i) = \big(\log_2 e_{h,i},\, \log_2 e_{w,i},\, \log_2(e_{h,i}e_{w,i}),\, \log_2(e_{h,i}/e_{w,i})\big). \tag{11}$$

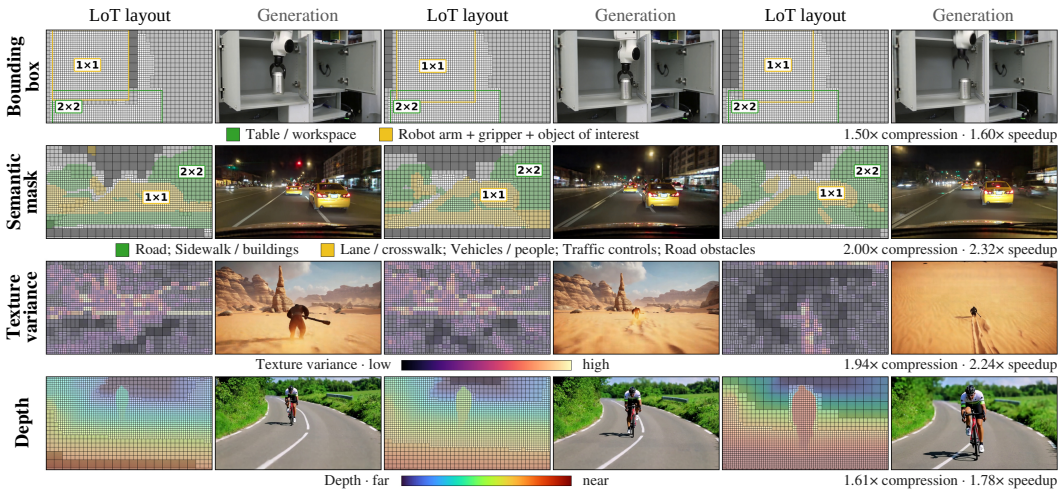


Figure 5: **Video generation with different layouts.** Our model generates video under arbitrary LoT layouts derived from bounding box, semantic mask, texture variance, or depth input, allocating tokens where an application needs detail and reducing them elsewhere, such as sky and road surface in autonomous driving and flat untextured surfaces in real-time gaming.

A zero-initialized MLP maps these features to the DiT hidden dimension and is added to the input token embedding:

$$\boldsymbol{h}_i^0 = \boldsymbol{W}_{\text{in},\boldsymbol{e}_i} \boldsymbol{x}_t^{(i)} + g_\phi(\boldsymbol{s}(\boldsymbol{e}_i)) \,. \tag{12}$$

**Finest-grid-aligned RoPE.** We retain the pretrained DiT's axial RoPE and index each token at its patch center on the uniform token grid

$$\boldsymbol{c}_i = \left( u_i + \frac{e_{h,i} - 1}{2},\, v_i + \frac{e_{w,i} - 1}{2} \right), \tag{13}$$

where $(u_i, v_i)$ is the top-left coordinate of the $i$-th token. We use $\boldsymbol{c}_i$ in Eq. 6 for both queries and keys, while the shape embedding conveys patch extent. This parameter-free RoPE change supports standard dense attention across all resolutions. For unit patches, $\boldsymbol{c}_i = (u_i, v_i)$, recovering the pretrained positional encoding. Unlike multi-scale RoPE variants such as Wu et al. (2026); Jeong et al. (2026); Chao et al. (2026) that require query-dependent position maps, our simple approach is query-independent and applies unchanged to every supported extent.

## 5 Results

*Experimental Setup.* For image generation, we fine-tune FLUX.2-klein-base-4B (Black Forest Labs, 2025) on Aesthetic-Train-V2 (Zhang et al., 2025b) at $1024 \times 1024$, following (Chao et al., 2026), with a random 95%/5% split for training and held-out evaluation. We also fine-tune the 9B variant using the separate training setup described in Sec. A.4. Image results use 4B unless otherwise noted. For video generation, we fine-tune Wan2.1-T2V-14B (Team Wan et al., 2025) on a selected 100k-example subset of Vchitect-T2V-Dataverse (Fan et al., 2025), using 81-frame clips at 720P. We reserve an additional 200 text-video pairs for held-out evaluation.

LoT-Flux supports extents $(e_h, e_w)$ with $e_h, e_w \in \{1, 2, 4, 8\}$, while LoT-Wan uses $(1, e_h, e_w)$ with $e_h, e_w \in \{1, 2, 4\}$. Models used for layout-controlled generation are trained on mixed-source layouts derived from semantic masks, object bounding boxes, reference-image texture variance, and depth maps. We adapt the pretrained transformer projections with LoRA (Hu et al., 2021) and fully fine-tune the extent-dependent input and output heads and patch-size embeddings. See the supplementary material for detailed training recipes: Sec. A.4 for images and Sec. A.10 for videos.

| Method | Compression | Speed | IR ↑ | HPSv2.1 ↑ | HPSv3 ↑ | FID ↓ | pFID ↓ | TOPIQ ↑ | MUSIQ ↑ |
|---|---|---|---|---|---|---|---|---|---|
| Full Res. FLUX2 4B | 1.000× | 1.001× | 0.9518 | 0.2808 | 10.4299 | 16.748 | 20.884 | 0.5852 | 69.783 |
| ToMe-SD | 1.893× | 1.538× | 0.1757 | 0.2182 | 5.3569 | 25.757 | 30.502 | 0.4888 | 58.991 |
| DDiT | | 1.340× | **0.8850** | 0.2613 | 9.0052 | 19.671 | 23.739 | 0.5593 | 66.233 |
| Foveated Diffusion | | 1.040× | 0.8280 | 0.2615 | 9.0092 | 14.944 | 18.595 | 0.5822 | 67.494 |
| Ours (LoT-Flux2 4B) | | **1.560**× | 0.8833 | **0.2728** | **10.1211** | **13.361** | **16.393** | **0.6080** | **69.691** |
| ToMe-SD | 2.406× | 1.793× | -0.2547 | 0.1946 | 2.9150 | 34.321 | 41.187 | 0.4488 | 53.558 |
| DDiT | | 1.521× | 0.7865 | 0.2512 | 8.0549 | 24.150 | 32.311 | 0.5444 | 64.612 |
| Foveated Diffusion | | 1.100× | 0.7737 | 0.2543 | 8.2674 | 15.702 | 21.456 | 0.5433 | 65.109 |
| Ours (LoT-Flux2 4B) | | **1.818**× | **0.8513** | **0.2672** | **9.7533** | **13.308** | **17.046** | **0.5978** | **69.420** |
| ToMe-SD | 2.935× | 2.003× | -0.6029 | 0.1781 | 0.9253 | 43.528 | 50.116 | 0.4203 | 49.119 |
| DDiT | | 1.321× | 0.7156 | 0.2447 | 7.4075 | 28.156 | 38.974 | 0.5341 | 63.435 |
| Foveated Diffusion | | 1.180× | 0.7582 | 0.2507 | 7.8433 | 16.124 | 23.645 | 0.5097 | 63.105 |
| Ours (LoT-Flux2 4B) | | **2.040**× | **0.8326** | **0.2639** | **9.5522** | **13.345** | **17.822** | **0.5873** | **68.760** |

Table 1: **Image generation baseline comparisons.** Our method achieves superior performance on most generation metrics across different token budgets, while remaining faster than the baselines.

| Method | Compression | Speed ↑ | Aesthetic ↑ | Imaging ↑ | Dynamic ↑ | Background ↑ | Subject ↑ | Motion ↑ |
|---|---|---|---|---|---|---|---|---|
| Full Res. Wan2.1 14B | 1× | 1.000× | 0.5518 | 0.6465 | 0.8900 | 0.9332 | 0.9254 | 0.9899 |
| ToMe-SD | 2× | 2.223× | 0.3497 | 0.4925 | 0.9000 | 0.9017 | 0.8235 | 0.9594 |
| Foveated Diffusion | | 2.153× | 0.5258 | 0.5598 | **0.9650** | **0.9252** | **0.8800** | **0.9844** |
| Ours (LoT-Wan2.1 14B) | | **2.262**× | **0.5324** | **0.6120** | **0.9650** | 0.9242 | 0.8775 | 0.9836 |
| ToMe-SD | 2.5× | 2.802× | 0.2930 | 0.4477 | 0.9300 | 0.9122 | 0.8071 | 0.9555 |
| Foveated Diffusion | | 2.746× | 0.5153 | 0.5263 | 0.9550 | **0.9236** | **0.8776** | **0.9839** |
| Ours (LoT-Wan2.1 14B) | | **2.909**× | **0.5204** | **0.5851** | **0.9650** | 0.9232 | 0.8729 | 0.9832 |
| ToMe-SD | 3× | 3.305× | 0.2704 | 0.4208 | **0.9800** | 0.9189 | 0.8031 | 0.9526 |
| Foveated Diffusion | | 3.276× | 0.5080 | 0.4952 | 0.9550 | **0.9234** | **0.8761** | **0.9839** |
| Ours (LoT-Wan2.1 14B) | | **3.530**× | **0.5104** | **0.5650** | 0.9500 | 0.9232 | 0.8687 | 0.9829 |

Table 2: **Video generation quantitative comparisons.** Our model obtains the best Aesthetic and Imaging scores compared with baselines and achieves the most speedup under the same budgets.

### 5.1 Layout-Adaptive Image and Video Generation

Figures 4 and 5 show image and video generation with token layouts derived from bounding boxes, semantic masks, texture variance, and depth maps. Across these sources, LoT generates content that adapts to the token allocation, retaining intricate details in finer-token regions while favoring smoother or defocused content in coarser-token regions. Fig. 6 further illustrates this behavior with user-specified layouts using LoT-Flux2-9B. For additional layout adaptive results see supplementary website and Sec. A.9.

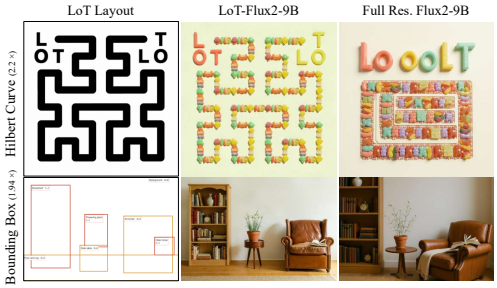


Figure 6: **Layout-adaptive image generation.**

### 5.2 Runtime and Quality Analysis of Image and Video Generation

We compare LoT with ToMe-SD (Bolya & Hoffman, 2023), DDiT (Kim et al., 2026), and Foveated Diffusion (Chao et al., 2026) for images, and with ToMe-SD and Foveated Diffusion for videos, using LoT-Flux2-4B and LoT-Wan2.1-14B fine-tuned exclusively on texture-variance-derived layouts and evaluating each method at matched token budgets. Speedups are measured relative to the corresponding full-resolution model and cover generation given a supplied layout, including diffusion sampling and VAE decoding and excluding layout construction and model loading. Comparison setups, metrics, and baseline details are in Sections A.5 and A.11 of the supplementary.

For images, we assess predicted human preference using ImageReward (Xu et al., 2023), HPSv2.1 (Wu et al., 2023), and HPSv3 (Ma et al., 2025); content fidelity using FID and pFID; and image quality using TOPIQ (Chen et al., 2024) and MUSIQ (Ke et al., 2021). In Tab. 1, LoT-Flux2-4B achieves 1.32–2.04× speedup and leads the baselines on most of the metrics across all token compression rates. For videos, Tab. 2 reports six VBench scores on 200 held-out prompts. LoT-Wan2.1-14B achieves 1.58–3.53× speedup and the highest aesthetic and imaging quality among the compressed methods at every budget, while remaining competitive on the other evaluated metrics.

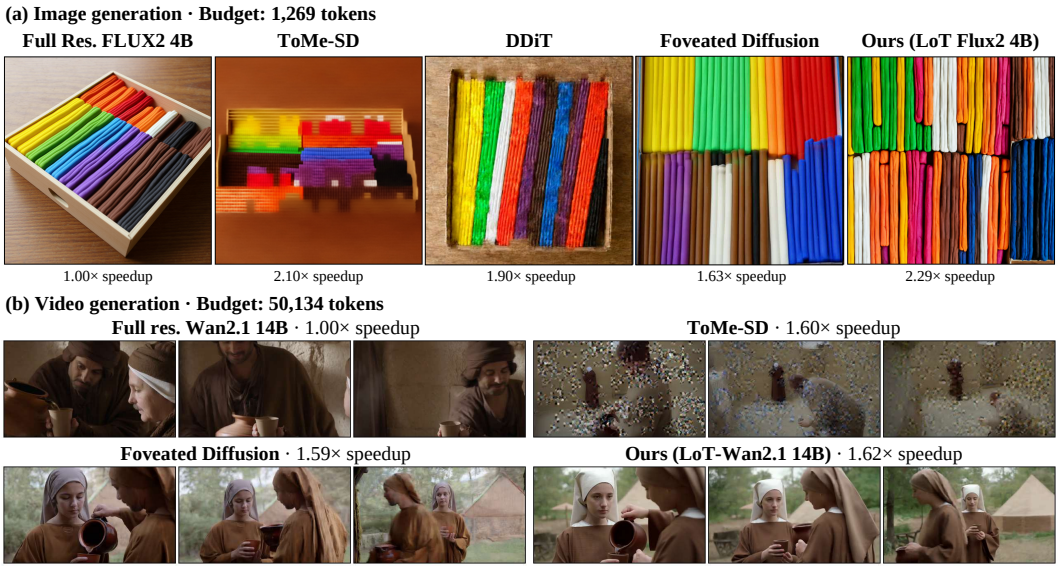


Figure 7: **Qualitative baseline comparisons** for (a) image and (b) video generation with the same token budget. ToMe-SD and DDiT lead to noisy low-res. regions. Foveated Diffusion results in distortions near mixed-res. token boundaries. We preserve visual detail while accelerating generation.

Qualitative image and video comparisons appear in Figures 7(a) and 7(b), respectively. See the supplementary website for additional side-by-side comparisons.

### 5.3 Ablation Study

Tab. 3 compares three variants of our framework design choices at a matched token budget: removing patch-size conditioning (*No size embedding*), replacing patch-wise asymmetric flow recovery with a vanilla flow parameterization (*Direct HR prediction*), and restricting token extents to $1 \times 1$ and $2 \times 2$ similar to previous works (Chao et al., 2026) (*Binary grid*). The full model performs best on all metrics except ImageReward, with particularly large gains over direct HR prediction and binary grids. Implementation details and additional qualitative and quantitative results are provided in Sec. A.13 of the supplementary material.

| Variant | HPSv3 ↑ | FID ↓ | MUSIQ ↑ |
|---|---|---|---|
| Direct HR prediction | 7.0595 | 26.487 | 63.346 |
| Binary grid | 8.1955 | 20.603 | 66.443 |
| No size embedding | <u>10.0284</u> | <u>14.584</u> | <u>69.433</u> |
| Ours | **10.3477** | **13.800** | **70.410** |

Table 3: **Ablation study.**

## 6 Conclusion

We introduced Level-of-Token Diffusion, a framework that uses level-of-detail information to guide efficient multiresolution image and video generation. By adapting pretrained diffusion models to variable token layouts, LoT connects control over detail with computational savings. Across diverse layout sources, our results demonstrate generation that adapts to the specified layout, with substantial speedups and favorable quality–efficiency tradeoffs over token-reduction baselines. We note that scenes requiring fine structures or textures throughout offer limited opportunities for token reduction, and aggressive compression can degrade their fidelity. LoT is therefore most advantageous when detail requirements vary spatially, allowing computation to be reduced selectively. See Sec. A.14 for a more detailed discussion.

Looking ahead, LoT suggests a closer connection between planning and efficient generation. Our demonstrations with agentic plans for images and videos provide an initial step, turning planned spatial importance into an explicit computational allocation. Level-of-Token layouts could serve as an editable intermediate representation, allowing multimodal systems and users to refine a visual plan and its computational budget. Extending this interaction to video could support faster iterations between planning and synthesis in robotics and interactive simulation, where low generation latency is essential for responsive interaction. More broadly, LoT offers a step toward generative systems that use their understanding of the intended content to decide where computation is needed.

### AI USE STATEMENT

We used generative AI tools to assist with implementing and debugging methods. We also used these tools for manuscript editing and making visualization code. The authors take responsibility for the final text, claims, code, and experimental results.

### ETHICS STATEMENT

LoT builds on pretrained image and video generation models and inherits their underlying biases and potential societal risks, including stereotypical representations and the generation of misleading or harmful content. Our changes to token allocation do not explicitly address these risks, and fine-tuning may alter the models' behavior. Reducing the computational cost of generation can broaden access to creative tools and research, but can also lower the cost of producing deceptive media at scale. Potential misuse includes impersonation and non-consensual imagery, with associated privacy and intellectual-property concerns. Responsible deployment should therefore include appropriate content safeguards, disclosure of synthetic content, respect for consent and applicable usage restrictions, and application-specific bias and safety evaluation. Our quality and efficiency evaluations do not establish the safety or fairness of downstream deployments.

### REPRODUCIBILITY STATEMENT

Sec. 4 describes the method. Training settings are provided in Sec. A.4, A.5, A.10, and A.11 specify the comparison setups, baselines, inference parameters, and metrics, while Sec. A.13 details the ablation variants and additional results. We will release code and model weights upon paper acceptance.

## REFERENCES

Marius Aasan, Martine Hjelkrem Tan, Nico Catalano, Changkyu Choi, and Adín Ramírez Rivera. Differentiable hierarchical visual tokenization. *Advances in Neural Information Processing Systems*, 38:87605–87634, 2026.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length. In *Forty-second International Conference on Machine Learning*, 2025.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *2023 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 4599–4603. IEEE, 2023.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL `https://arxiv.org/abs/2511.16719`.

Brian Chao, Lior Yariv, Howard Xiao, and Gordon Wetzstein. Foveated diffusion: Efficient spatially adaptive image and video generation. *arXiv preprint arXiv:2603.23491*, 2026.

Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Topiq: A top-down approach from semantics to distortions for image quality assessment. *IEEE Transactions on Image Processing*, 33:2404–2418, 2024.

Hansheng Chen, Jan Ackermann, Minseo Kim, Gordon Wetzstein, and Leonidas Guibas. Asymmetric flow models. *arXiv preprint arXiv:2605.12964*, 2026.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

James H. Clark. Hierarchical geometric models for visible-surface algorithms. In *Proceedings of the 3rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '76, pp. 267, New York, NY, USA, 1976. Association for Computing Machinery. ISBN 9781450373531. doi: 10.1145/563274.563323. URL https://doi.org/10.1145/563274.563323.

Shivam Duggal, Phillip Isola, Antonio Torralba, and William T Freeman. Adaptive length image tokenization via recurrent allocation. *arXiv preprint arXiv:2411.02393*, 2024.

Weichen Fan, Chenyang Si, Junhao Song, Zhenyu Yang, Yinan He, Long Zhuo, Ziqi Huang, Ziyue Dong, Jingwen He, Dongwei Pan, et al. Vchitect-2.0: Parallel transformer for scaling up video diffusion models. *arXiv preprint arXiv:2501.08453*, 2025.

Haipeng Fang, Sheng Tang, Juan Cao, Enshuo Zhang, Fan Tang, and Tong-Yee Lee. Attend to not attended: Structure-then-detail token merging for post-training dit acceleration. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18083–18092. IEEE, 2025.

Moayed Haji-Ali, Willi Menapace, Ivan Skorokhodov, Dogyun Park, Anil Kag, Michael Vasilkovsky, Sergey Tulyakov, Vicente Ordonez, and Aliaksandr Siarohin. One model, many budgets: Elastic latent interfaces for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4558–4568, 2026.

Akash Haridas, Utkarsh Saxena, Parsa Ashrafi Fashi, Mehdi Rezagholizadeh, Vikram Appia, and Emad Barsoum. Dc-dit: Adaptive compute and elastic inference for visual generation via dynamic chunking. *arXiv preprint arXiv:2603.06351*, 2026.

Joakim Bruslund Haurum, Sergio Escalera, Graham W Taylor, and Thomas B Moeslund. Agglomerative token clustering. In *European Conference on Computer Vision*, pp. 200–218. Springer, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Ideogram. Ideogram 4 Prompting Guide. https://github.com/ideogram-oss/ideogram4/blob/main/docs/prompting.md, 2026a. Accessed: 2026-09-12.

Ideogram. Ideogram 4.0. https://ideogram.ai/models/4.0, 2026b. Released June 3, 2026. Accessed: 2026-09-15.

Wongi Jeong, Kyungryeol Lee, Hoigi Seo, and Se Young Chun. Training-free mixed-resolution latent upsampling for spatially accelerated diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18673–18682, 2026.

Weinan Jia, Mengqi Huang, Nan Chen, Lei Zhang, and Zhendong Mao. D 2 it: Dynamic diffusion transformer for accurate image generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12860–12870. IEEE, 2025.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24174–24184. IEEE, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5128–5137. IEEE, 2021.

Dahye Kim, Deepti Ghadiyaram, and Raghudeep Gadde. Ddit: Dynamic patch scheduling for efficient diffusion transformers. *arXiv preprint arXiv:2602.16968*, 2026.

Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1372–1381. IEEE, 2024.

Orest Kupyn, Hirokatsu Kataoka, and Christian Rupprecht. S3od: towards generalizable salient object detection with synthetic data. In *International Conference on Learning Representations*, volume 2026, pp. 145157–145179, 2026.

Min-Jeong Lee, Hee-Dong Kim, and Seong-Whan Lee. Local representative token guided merging for text-to-image generation. In *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 966–971. IEEE, 2025.

Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, Maneesh Agrawala, Ion Stoica, Kurt Keutzer, and Song Han. Radial attention: $o(n \log n)$ sparse attention with energy decay for long video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7353–7363. IEEE, 2025a.

Jiacheng Liu, Peiliang Cai, Qinming Zhou, Yuqi Lin, Deyang Kong, Benhao Huang, Yupei Pan, Haowen Xu, Chang Zou, Junshu Tang, et al. Freqca: Accelerating diffusion models via frequency-aware caching. *arXiv preprint arXiv:2510.08669*, 2025b.

Wenbo Lu, Shaoyi Zheng, Yuxuan Xia, and Shengjie Wang. Toma: Token merge with attention for diffusion models. *arXiv preprint arXiv:2509.10918*, 2025.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15762–15772. IEEE, 2024.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15086–15095. IEEE, 2025.

MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities, 2026. URL https://www.minimax.io/blog/minimax-h3. Official model announcement, July 31, 2026.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i5.28226. URL https://doi.org/10.1609/aaai.v38i5.28226.

OpenAI. Introducing ChatGPT Images 2.0. https://openai.com/index/introducing-chatgpt-images-2-0/, 2026. Released April 21, 2026. Accessed: 2026-09-15.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Reve AI. Introducing Reve 2.0. https://blog.reve.com/posts/announcing-reve-2.0/, 2026. Released June 3, 2026. Accessed: 2026-06-18.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.

Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. doi: 10.1007/BF02289451.

Jerome M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, 1993.

Junhong Shen, Kushal Tirumala, Michihiro Yasunaga, Ishan Misra, Luke Zettlemoyer, Lili Yu, and Chunting Zhou. CAT: Content-adaptive image tokenization. *arXiv preprint arXiv:2501.03120*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14940–14950. IEEE, 2025.

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.

K. Vaidyanathan, M. Salvi, R. Toth, T. Foley, T. Akenine-Möller, J. Nilsson, J. Munkberg, J. Hasselgren, M. Sugihara, P. Clarberg, T. Janczak, and A. Lefohn. Coarse pixel shading. In *Proceedings of High Performance Graphics*, HPG '14, pp. 9–18, Goslar, DEU, 2014. Eurographics Association.

Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Transactions on Graphics*, 37(4), 2018. doi: 10.1145/3197517.3201329.

Fu-Yun Wang, Zhaoyang Huang, Alexander W. Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, Xiaogang Wang, and Hongsheng Li. Phased consistency models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Haoyu Wu, Jingyi Xu, Hieu Le, and Dimitris Samaras. Importance-based token merging for efficient image and video generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4983–4995. IEEE, 2025.

Haoyu Wu, Jingyi Xu, Qiaomu Miao, Dimitris Samaras, and Hieu Le. Phase-aligned rope for mixed-resolution diffusion transformer. In *European Conference on Computer Vision*, pp. 272–293. Springer, 2026.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Yifei Xia, Suhan Ling, Fangcheng Fu, Yujie Wang, Huixia Li, Xuefeng Xiao, and Bin Cui. Training-free and adaptive sparse attention for efficient long video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Howard Xiao, Brian Chao, Lior Yariv, and Gordon Wetzstein. Spectral progressive diffusion for efficient image and video generation. *arXiv preprint arXiv:2605.18736*, 2026. URL https://arxiv.org/abs/2605.18736.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13294–13304. IEEE, 2025.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 15903–15935, 2023.

Wilson Yan, Volodymyr Mnih, Aleksandra Faust, Matei Zaharia, Pieter Abbeel, and Hao Liu. Elastictok: Adaptive tokenization for image and video. In *International Conference on Learning Representations*, volume 2025, pp. 38036–38056, 2025.

Lei Yang, Dmitry Zhdan, Emmett Kilgariff, Eric B. Lum, Yubo Zhang, Matthew Johnson, and Henrik Rydgård. Visually lossless content and motion adaptive shading in games. *Proc. ACM Comput. Graph. Interact. Tech.*, 2(1), June 2019. doi: 10.1145/3320287. URL https://doi.org/10.1145/3320287.

Yang Yang, Siming Zheng, Jinwei Chen, Boxi Wu, Xiaofei He, Deng Cai, Bo Li, and Peng-Tao Jiang. Any-to-bokeh: One-step video bokeh via multi-plane image guided diffusion. *arXiv preprint arXiv:2505.21593v1*, 2025. URL https://arxiv.org/abs/2505.21593v1.

Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Reco: Region-controlled text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14246–14255, June 2023.

Haotian Ye, Qiyuan He, Jiaqi Han, Puheng Li, Jiaojiao Fan, Zekun Hao, Fitsum Reda, Yogesh Balaji, Huayu Chen, Sheng Liu, Angela Yao, James Zou, Stefano Ermon, Haoxiang Wang, and Ming-Yu Liu. InfoTok: Adaptive discrete video tokenizer via information-theoretic compression. *arXiv preprint arXiv:2512.16975*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024b.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974. IEEE, 2025.

Eduard Zamfir, Christian Reisswig, Zongwei Wu, Yongqin Xian, and Radu Timofte. Elastic token compression for pixel-space diffusion transformers, 2026. URL `https://arxiv.org/abs/2608.29281`.

Chenlu Zhan, Wen Li, Chuyu Shen, Jun Zhang, Suhui Wu, and Hao Zhang. Bidirectional sparse attention for faster video diffusion training. *arXiv preprint arXiv:2509.01085*, 2025.

Hui Zhang, Dexiang Hong, Yitong Wang, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-Gang Jiang. Creatilayout: Siamese multimodal diffusion transformer for creative layout-to-image generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18487–18497. IEEE, 2025a.

Jinjin Zhang, Qiuyu Huang, Junjie Liu, Xiefan Guo, and Di Huang. Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3836–3847, 2023.

Zhengqiang Zhang, Rongyuan Wu, Lingchen Sun, and Lei Zhang. GPSToken: Gaussian parameterized spatially-adaptive tokenization for image representation and generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025c.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Hao Luo, Yibing Song, Gao Huang, Fan Wang, and Yang You. DyDiT++: Diffusion transformers with timestep and spatial dynamics for efficient visual generation. *arXiv preprint arXiv:2504.06803*, 2025a.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer. In *International Conference on Learning Representations*, volume 2025, pp. 65520–65552, 2025b.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *arXiv preprint arXiv:2410.05317*, 2024.

# A Appendix

Unless otherwise specified, image-generation results in the main paper and this appendix use LoT-FLUX.2-4B and the corresponding 4B baselines. Figure 6 uses LoT-FLUX.2 9B. Additional LoT-FLUX.2 9B results are provided on the supplementary website.

## Contents

A.1 Supplementary Evaluation Material . . . . . 16

A.2 Extent-dependent Subspace Fitting and Data Scaling . . . . . 16

A.3 Deriving Token Layouts from Different Sources . . . . . 17

A.4 Image Generation Training Details . . . . . 18

A.5 Image Generation Baseline Comparison Details . . . . . 19

A.6 Additional Image Generation Baseline Comparison Qualitative Results . . . . . 19

A.7 Additional Quantitative Comparison: Token Budget and Image Quality . . . . . 22
A.8 Additional Layout-Adaptive Image Generation Results . . . . . 28
A.9 Layout Controllability Compared with Dense Generation . . . . . 33
A.10 Video Generation Training Details . . . . . 36
A.11 Video Generation Baseline Comparison Details . . . . . 36
A.12 Additional Video Results . . . . . 36
A.13 Additional Ablation Study Results . . . . . 37
A.14 Limitations . . . . . 39

### A.1 Supplementary Evaluation Material

We encourage readers to view the accompanying website alongside the paper and PDF appendix to assess layout-dependent generation and temporal behavior beyond static figures. It presents recorded manual and agent-assisted image-layout workflows (website Section 2), typography and curve-based layouts with full-resolution FLUX.2 comparisons (Section 4.4), temporal-layout workflows and animated 3D-scene demonstrations (Section 1), and synchronized video baseline comparisons (Section 5). It also provides further layout-adaptive image and video results (Sections 4 and 3, respectively) and image baseline comparisons (Section 6), with token overlays for inspecting the relationship between allocation and generated content. The recordings show how layouts are constructed and used, while full video playback enables inspection of motion and temporal artifacts that isolated frames cannot reveal. Website: `https://iclr-submission7014.github.io/supplementary/`.

### A.2 Extent-dependent Subspace Fitting and Data Scaling

For each extent $\boldsymbol{e} = (e_h, e_w)$, we collect aligned pairs $(\boldsymbol{x}, \bar{\boldsymbol{z}}) \sim \mathcal{D}_{\boldsymbol{e}}$. For both image and video models, we fit the Procrustes bases and associated scales on a randomly sampled subset of the respective training split. Here, $\boldsymbol{x} \in \mathbb{R}^{D_{\boldsymbol{e}}}$ is a vectorized dense patch, and $\bar{\boldsymbol{z}} \in \mathbb{R}^{D}$ is the corresponding token from an extent-matched multi-scale VAE encoding after the pretrained model's fixed patchification. Thus, $D$ denotes the patchified token dimension, and extents are measured on the uniform token lattice, as in Sec. 3.1. Following AsymFlow (Chen et al., 2026), we fit a semi-orthonormal lift $\boldsymbol{A}_{\boldsymbol{e}} \in \mathbb{R}^{D_{\boldsymbol{e}} \times D}$ with orthogonal Procrustes:

$$\boldsymbol{A}_{\boldsymbol{e}}^{\star} = \underset{\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{I}_D}{\arg\min}\ \mathbb{E}_{(\boldsymbol{x}, \bar{\boldsymbol{z}}) \sim \mathcal{D}_{\boldsymbol{e}}} \left[ \|\boldsymbol{x} - \boldsymbol{A}\bar{\boldsymbol{z}}\|_2^2 \right]. \tag{14}$$

The fitted lift defines the extent-specific projector $\boldsymbol{P}_{\boldsymbol{e}} = \boldsymbol{A}_{\boldsymbol{e}} \boldsymbol{A}_{\boldsymbol{e}}^\top$.

Projection alone does not preserve the pretrained token magnitude. We therefore fit a scalar data scale for each extent by matching projected-token norms:

$$s_{\boldsymbol{e}} = \left( \frac{\mathbb{E}_{(\boldsymbol{x}, \bar{\boldsymbol{z}}) \sim \mathcal{D}_{\boldsymbol{e}}} \left[ \|\boldsymbol{A}_{\boldsymbol{e}}^\top \boldsymbol{x}\|_2^2 \right]}{\mathbb{E}_{(\boldsymbol{x}, \bar{\boldsymbol{z}}) \sim \mathcal{D}_{\boldsymbol{e}}} \left[ \|\bar{\boldsymbol{z}}\|_2^2 \right]} \right)^{1/2}. \tag{15}$$

Consequently, $\boldsymbol{A}_{\boldsymbol{e}}^\top \boldsymbol{x} / s_{\boldsymbol{e}}$ and $\bar{\boldsymbol{z}}$ have the same root-mean-square norm. Given a partition $\mathcal{P}$, we broadcast the corresponding scale over every dense site in each patch and apply the patch-wise asymmetric flow of Sec. 4.1 in the scaled data space:

$$s_{\mathcal{P}}(\boldsymbol{q}) = s_{\boldsymbol{e}_i} \quad \text{for } \boldsymbol{q} \in R_i, \qquad \boldsymbol{y}_0(\boldsymbol{q}) = \frac{\boldsymbol{x}_0(\boldsymbol{q})}{s_{\mathcal{P}}(\boldsymbol{q})}, \qquad \boldsymbol{y}_t = \alpha_t \boldsymbol{y}_0 + \sigma_t \boldsymbol{\varepsilon}. \tag{16}$$

We use Eq. 8 and 9 with $(\boldsymbol{x}_0, \boldsymbol{x}_t)$ replaced by $(\boldsymbol{y}_0, \boldsymbol{y}_t)$. The model predicts asymmetric velocity, from which we recover the dense velocity $\widehat{\boldsymbol{u}}$ and form the clean-data estimate $\widehat{\boldsymbol{y}}_0 = \boldsymbol{y}_t - \sigma_t \widehat{\boldsymbol{u}}$. We train this velocity parameterization using a clean-data loss with clamped noise-level weighting:

$$\mathcal{L} = \mathbb{E}\left[ \frac{\|\widehat{\boldsymbol{y}}_0 - \boldsymbol{y}_0\|_2^2}{\max(\sigma_t, 0.05)^2} \right]. \tag{17}$$

At inference time, we integrate the flow in $\boldsymbol{y}$-space and map the result back to the pretrained latent space as $\widehat{\boldsymbol{x}}_0(\boldsymbol{q}) = s_{\mathcal{P}}(\boldsymbol{q})\widehat{\boldsymbol{y}}_0(\boldsymbol{q})$. The multi-scale VAE tokens are used only to fit $\boldsymbol{A_e}$ and $s_{\boldsymbol{e}}$; both remain fixed during training and inference.

**Comparison with timestep calibration.** For a single global scale $s$, AsymFlow (Chen et al., 2026) retains the flow in the original data coordinates and compensates for the scale mismatch through input/output rescaling and a calibrated internal timestep. A patch-wise extension to our extent-dependent scales would use

$$k_{\boldsymbol{e}}(t) = \frac{1}{s_{\boldsymbol{e}}(1-t)+t}, \qquad \tau_{\boldsymbol{e}}(t) = t\,k_{\boldsymbol{e}}(t). \tag{18}$$

Token $i$ would then use input factor $k_{\boldsymbol{e}_i}(t)$ and internal timestep $\tau_{\boldsymbol{e}_i}(t)$. When scales differ, these internal timesteps generally differ within a sample, departing from the pretrained DiT's shared-timestep conditioning. We do not use this extent-dependent timestep calibration. Instead, we scale the clean data before adding standard Gaussian noise as in Eq. 16. All tokens then use the same flow timestep $t$, and the orthonormal projections preserve standard Gaussian projected noise, avoiding this additional conditioning mismatch.

### A.3 Deriving Token Layouts from Different Sources

We describe how we derive level-of-token layouts from semantic masks and bounding boxes, texture variance (VRS), depth maps, and custom detail maps. The same spatial allocation rules apply to images and video frames.

**Semantic masks and bounding boxes.** During training, we sample up to three regions defined by semantic masks or bounding boxes. Masks can also come from S3OD (Kupyn et al., 2026), using the same token-allocation procedure. Each selected region $M_k$ is assigned a token level $\ell_k \in \{0,1,2,3\}$, corresponding to a spatial token extent $b_k = 2^{3-\ell_k}$ in each axis. Thus, levels $0,1,2,3$ correspond to $8\times8$, $4\times4$, $2\times2$, and $1\times1$ tokens, respectively. We also assign the background a level $\ell_{\mathrm{bg}}$ and sample $\ell_k > \ell_{\mathrm{bg}}$, so selected regions receive finer tokens. The resulting spatial assignment is

$$\ell(\boldsymbol{q}) = \max(\{\ell_{\mathrm{bg}}\} \cup \{\ell_k : \boldsymbol{q} \in M_k\}), \qquad b(\boldsymbol{q}) = 2^{3-\ell(\boldsymbol{q})}. \tag{19}$$

This assigns a token level throughout each region and the background, with the finer level taking precedence where selected regions overlap. We use the resulting extent map to construct the level-of-token layout.

**Texture variance (Variable-Rate Shading).** We adapt the content-adaptive VRS rule of Yang et al. (2019) to select token extents using directional luminance variation. Candidate regions are coarsened along directions whose estimated error falls below the tolerance

$$\tau(R) = s\big(\mathbb{E}_R[I] + a\big), \tag{20}$$

where $I$ is reference luminance. The sensitivity $s$ and ambient term $a$ control the overall detail of the token layout: larger values permit coarser tokens, while smaller values preserve finer detail. For threshold-based VRS training, we independently sample $s \in [0.01, 0.1]$ and $a \in [0.01, 1.0]$ uniformly. For quantitative evaluation, reference luminance comes from held-out reference images for image comparisons and prompt-generated reference videos for video comparisons, as detailed in Sec. A.5 and A.11.

**Depth maps and depth of field.** Following the circle-of-confusion model used in computational bokeh rendering (Wadhwa et al., 2018; Yang et al., 2025), we convert a positive depth map $d(\boldsymbol{q})$ and focal depth $d_f$ into a blur-radius field

$$r(\boldsymbol{q}) = K\left|d(\boldsymbol{q})^{-1} - d_f^{-1}\right|. \tag{21}$$

The focal depth is the median depth of a selected subject region. We calibrate $K$ for the reference depth map and selected focal plane as

$$K = \frac{r_{\mathrm{target}}}{\mathrm{percentile}_{95}\Big(\left|d(\boldsymbol{q})^{-1} - d_f^{-1}\right|\Big)}, \tag{22}$$

where the percentile is computed over the reference pixels and $r_{\text{target}}$ sets the desired blur strength. The first equation gives a small blur radius $r(\boldsymbol{q})$ when a pixel's depth is close to the focal depth $d_f$, and a larger radius as its inverse depth differs from the focal plane. The second equation scales these radii so their 95th percentile equals $r_{\text{target}}$. We then use this radius map to choose token sizes: small radii require fine tokens, while larger radii permit coarser tokens. A region is merged only when its smallest blur radius supports the larger token size. Thus, even a small in-focus part prevents a region from being coarsened, preserving detail around the subject and its boundaries. Increasing $r_{\text{target}}$ increases $K$ and the blur radii, allowing coarser layouts.

**Custom detail maps.** Users can specify where to allocate computation by painting a spatial detail map, without requiring a reference image or an automatically extracted mask. An agent can also construct this map programmatically from a text description. In our image-generation interface, each pixel is assigned the score of its nearest brush color: black, orange, yellow, and white correspond to detail scores $0$, $0.3$, $0.6$, and $1$, respectively. The canvas initially uses yellow to provide medium detail throughout the image. We area-average the scores onto the finest token lattice, optionally apply Gaussian smoothing, and multiply by a user-controlled gain before clipping to $[0, 1]$. This gives a detail field $D(\boldsymbol{q})$, where larger values request finer tokens.

We convert $D$ into a valid, lattice-aligned quadtree tiling. Starting from $8 \times 8$ blocks, a block $R$ of extent $b \in \{8, 4, 2\}$ is split into four children whenever

$$\max_{\boldsymbol{q} \in R} D(\boldsymbol{q}) \geq t_b, \qquad t_8 \leq t_4 \leq t_2. \tag{23}$$

Recursion stops when the test fails or a $1 \times 1$ token is reached. The maximum is taken after projection onto the token lattice, so a high-detail cell prevents its containing block from remaining coarse. The thresholds, smoothing, and gain let the user adjust the resulting token count without requiring an exact token budget or VRS-based refinement. The model receives the resulting token layout and text prompt, not the painted RGB map as an additional appearance condition. Section 1 of the supplementary website demonstrates both manual painting and agent-created custom detail maps.

### A.4 Image Generation Training Details

We fine-tune FLUX.2-klein-base-4B (Black Forest Labs, 2025) with its standard latent normalization, keeping the pretrained text encoder and VAE frozen.

**Training data and source mixture.** We train on Aesthetic-Train-V2 (Zhang et al., 2025b), using aspect-preserving resizing and $1024 \times 1024$ center crops, with a random 95%/5% split for training and held-out evaluation. For each training image, we derive its tokenization layout from segmentation masks, bounding boxes from SAM3 (Carion et al., 2025), image-luminance variation (texture variance), and depth maps (Lin et al., 2025).

**Training setup.** We fully fine-tune the per-extent input/output heads, patch-size MLP, and final adaptive normalization. We apply LoRA (Hu et al., 2021) with rank $r = 256$ and dropout 0.05 to the image and context query/key/value and output projections in the dual-stream blocks; image and context feed-forward `linear_in` and `linear_out` layers; the fused `to_qkv_mlp_proj` inputs and single-stream attention `to_out` projections; and the timestep embedder. All other pretrained parameters remain frozen.

We train for 50,000 steps using AdamW with a total batch size of 32, learning rate $10^{-4}$, $(\beta_1, \beta_2) = (0.9, 0.999)$, and $\epsilon = 10^{-8}$. Weight decay of $0.01$ is applied only to the LoRA adapters. We train the asymmetric-velocity predictor with the clean-data loss in Eq. 17, using the scaled space of Eq. 16. The weight $1/\max(\sigma_t, 0.05)^2$ is bounded near zero noise. We use power-function EMA (Karras et al., 2024) with $\gamma = 7$ for evaluation.

**9B model.** We also fine-tune FLUX.2-klein-base-9B (Black Forest Labs, 2025). We use three million images from the LAION-3M dataset used by Asymmetric Flow Models (Chen et al., 2026) for the semantic-mask, bounding-box, saliency, and VRS branches, and bokeh-rendered images from Aesthetic-Train-V2 (Zhang et al., 2025b) for the depth-of-field branch. The five branches are sampled with equal probability. We train on 32 NVIDIA H100 GPUs across four nodes with a

total batch size of 256, and use the 24,500-step EMA checkpoint for the qualitative results on the supplementary website.

### A.5 Image Generation Baseline Comparison Details

**Inference parameters.** For all baselines, we generate with 30 sampling steps, classifier-free guidance scale 4, and orthogonal guidance strength 1. We evaluate on all 5,261 prompts in the held-out split. For each prompt, we derive the VRS layout from its corresponding held-out reference image. This is a layout-conditioned comparison: the layout provides reference-derived spatial detail allocation beyond the text prompt, but the reference RGB image is not an input to the generator.

**Training Details.** For a fair comparison, we extend DDiT (Kim et al., 2026) and Foveated Diffusion (Chao et al., 2026) to support token levels with spatial extents up to $8\times8$. For these comparisons, we train ours, DDiT, and Foveated Diffusion for 20,000 steps each exclusively on VRS-derived grids using the same settings: rank-256 LoRA with dropout 0.05 on attention, feed-forward, and timestep projection layers, AdamW with learning rate $10^{-4}$ and weight decay $0.01$, and a global batch size of 16. We evaluate the 20,000-step power-function EMA checkpoint with $\gamma = 7$.

We also fine-tune a full-resolution FLUX.2-klein-base-4B control on the same training split as our VRS-only quantitative model, using the same LoRA target layers, rank 256, dropout 0.05, AdamW learning rate $10^{-4}$, weight decay $0.01$, global batch size 16, and 20,000 training steps.

**Metric details.** We evaluate overall generation quality, content fidelity, image quality, and efficiency. For overall generation quality, ImageReward (IR) (Xu et al., 2023), HPSv2.1 (Wu et al., 2023), and HPSv3 (Ma et al., 2025) are learned predictors of human preference for prompt–image pairs, capturing prompt alignment and visual appeal. Higher scores are better.

For content fidelity, Fréchet Inception Distance (FID) (Heusel et al., 2017) compares the feature distributions of generated images and held-out reference images. Patch FID (pFID) applies the same comparison to one native-resolution $299\times299$ center crop per image, emphasizing local appearance.

For image quality, TOPIQ (Chen et al., 2024) combines semantic information with local distortion cues, while MUSIQ (Ke et al., 2021) assesses perceptual quality using multi-scale image representations. Both are used without a reference image or text prompt; higher scores are better.

For efficiency, runtime measures the CUDA-synchronized generation call, including prompt encoding, diffusion sampling, and VAE decoding, on an H100 per image; it excludes grid preprocessing, image saving, and model loading. We retain the original pretrained full-resolution FLUX.2 4B timing reference of $5.7783$ seconds per image for the reported speedups. The fine-tuned dense control takes $5.7726$ seconds per image (steady-state mean), corresponding to $1.001\times$ speed relative to that reference.

### A.6 Additional Image Generation Baseline Comparison Qualitative Results

We show additional image generation baseline comparison qualitative results in Fig. 8 and 9. More qualitative image baseline comparisons are available in Section 3 (Image Generation Qualitative Baseline Comparisons) of the supplementary website.

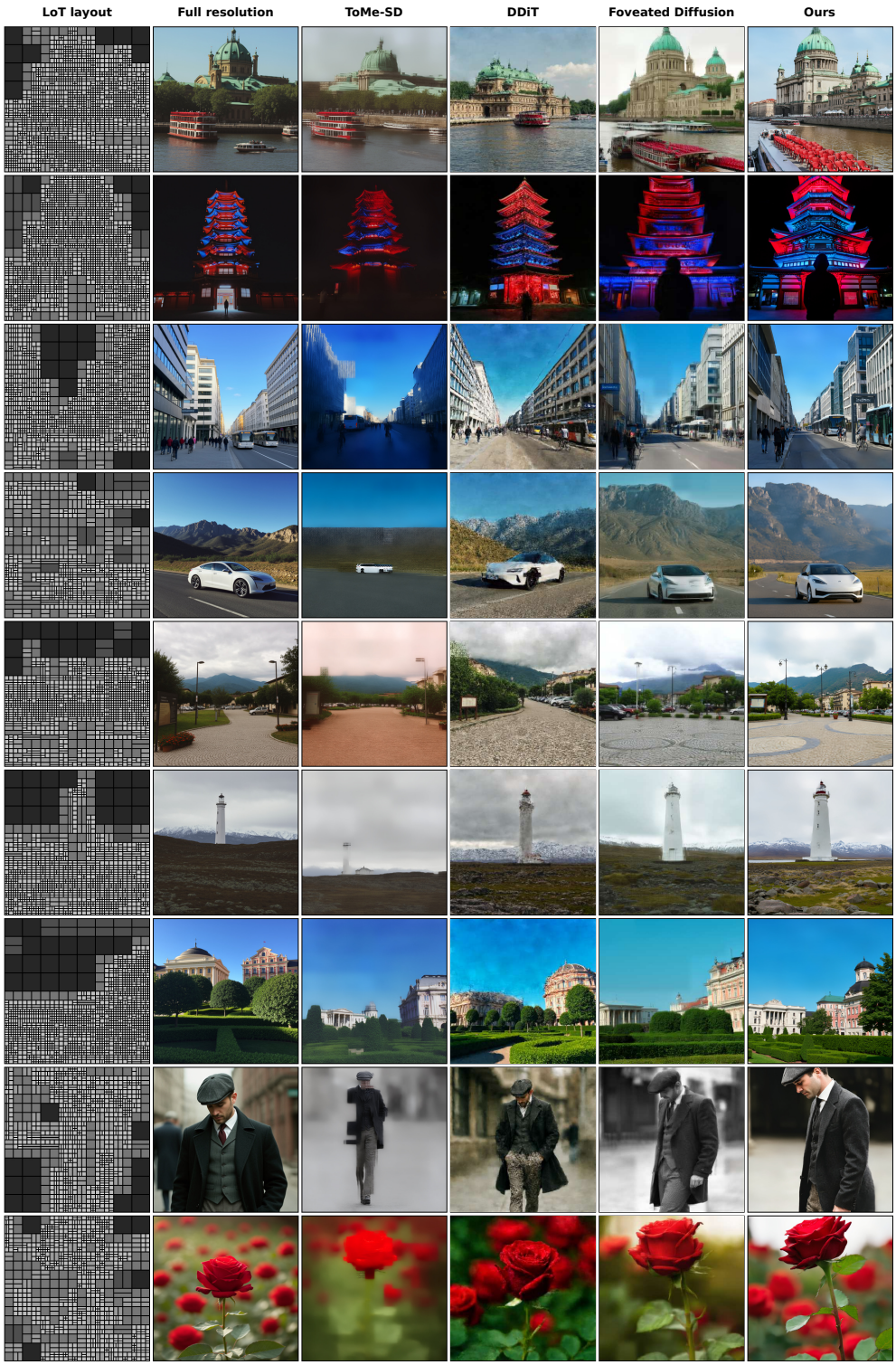


Figure 8: **Additional Qualitative Image Generation Baseline Comparison Results.**

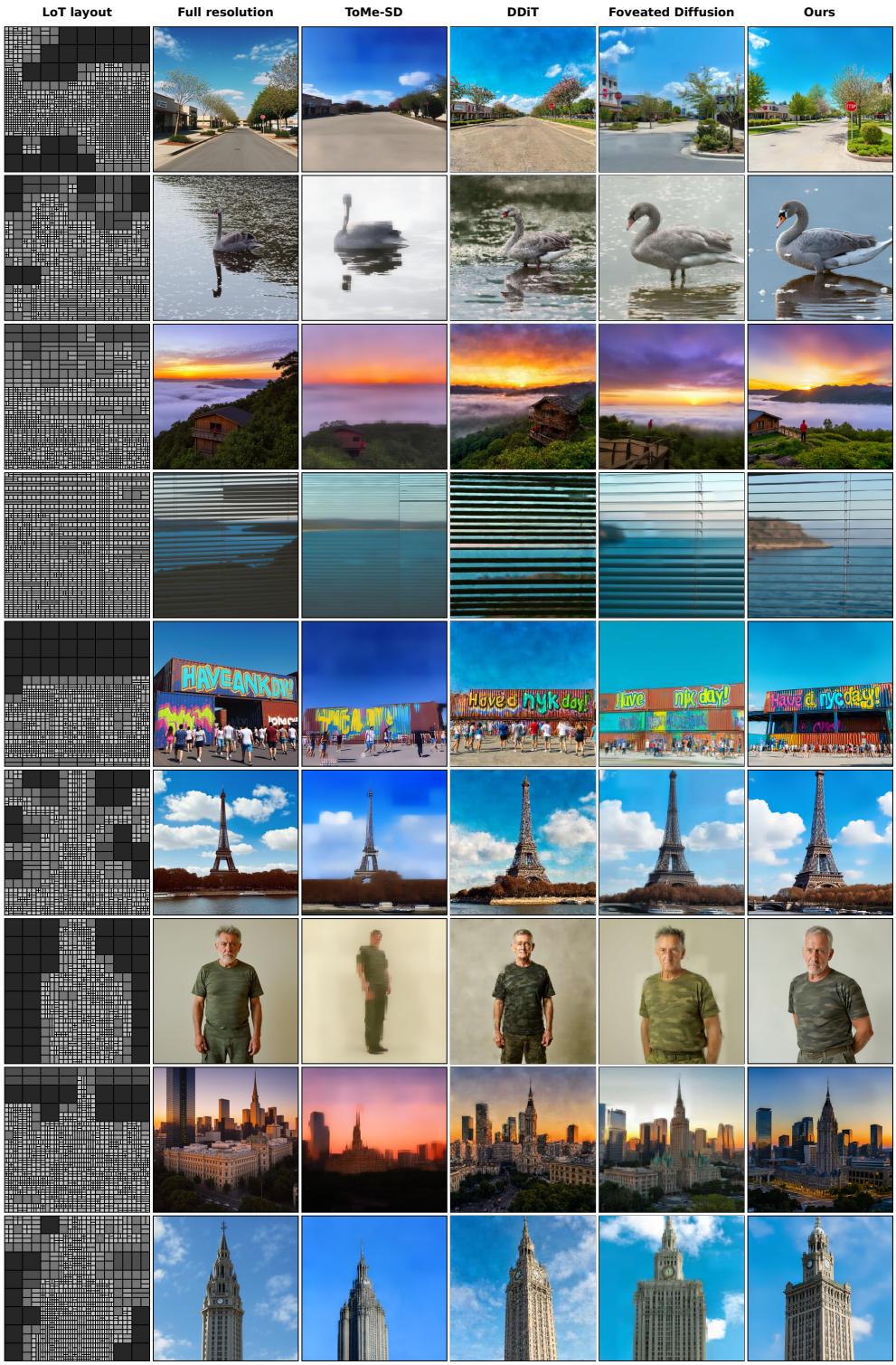


Figure 9: **Additional Qualitative Image Generation Baseline Comparison Results.**

### A.7 Additional Quantitative Comparison: Token Budget and Image Quality

We provide an additional comparison between LoT-FLUX.2-4B and pretrained FLUX.2 4B at matched per-image token budgets, using all 5,261 held-out test prompts. We denote the pretrained baseline as *FLUX.2 (token matched)*: for each prompt, it uses a uniform square token grid whose token count is as close as possible to that of our content-adaptive grid. It generates at the corresponding lower resolution and is bicubically upsampled to $1024 \times 1024$ for evaluation. LoT-FLUX.2-4B instead generates directly at $1024 \times 1024$ using spatially varying token extents.

Fig. 14(a) compares FID and pFID against the held-out reference images, together with the image-quality metrics MUSIQ and TOPIQ, averaged over the test images at each threshold. Overall, LoT-FLUX.2-4B achieves lower FID and pFID and higher image-quality scores than FLUX.2 (token matched). This advantage is consistent with our adaptive allocation of finer tokens to detailed regions while retaining full-resolution output. In contrast, the pretrained baseline reduces resolution uniformly, and subsequent upsampling cannot recover fine details lost during low-resolution generation. Fig. 14(b) provides qualitative comparisons, with each token-matched output displayed at its generation scale relative to our $1024 \times 1024$ output.

Figures 10–13 show how generation changes as the target token budget $B$ varies for the same prompt. We guide token allocation using semantic masks in Figures 10 and 11, and texture variance in Figures 12 and 13. Lower budgets use larger tokens over more of the image, retaining recognizable subjects and scene structure while simplifying fine textures. Higher budgets support finer spatial detail, such as foliage, fur, and architectural features. These results illustrate a controllable trade-off between computational budget and visual detail within the same model.

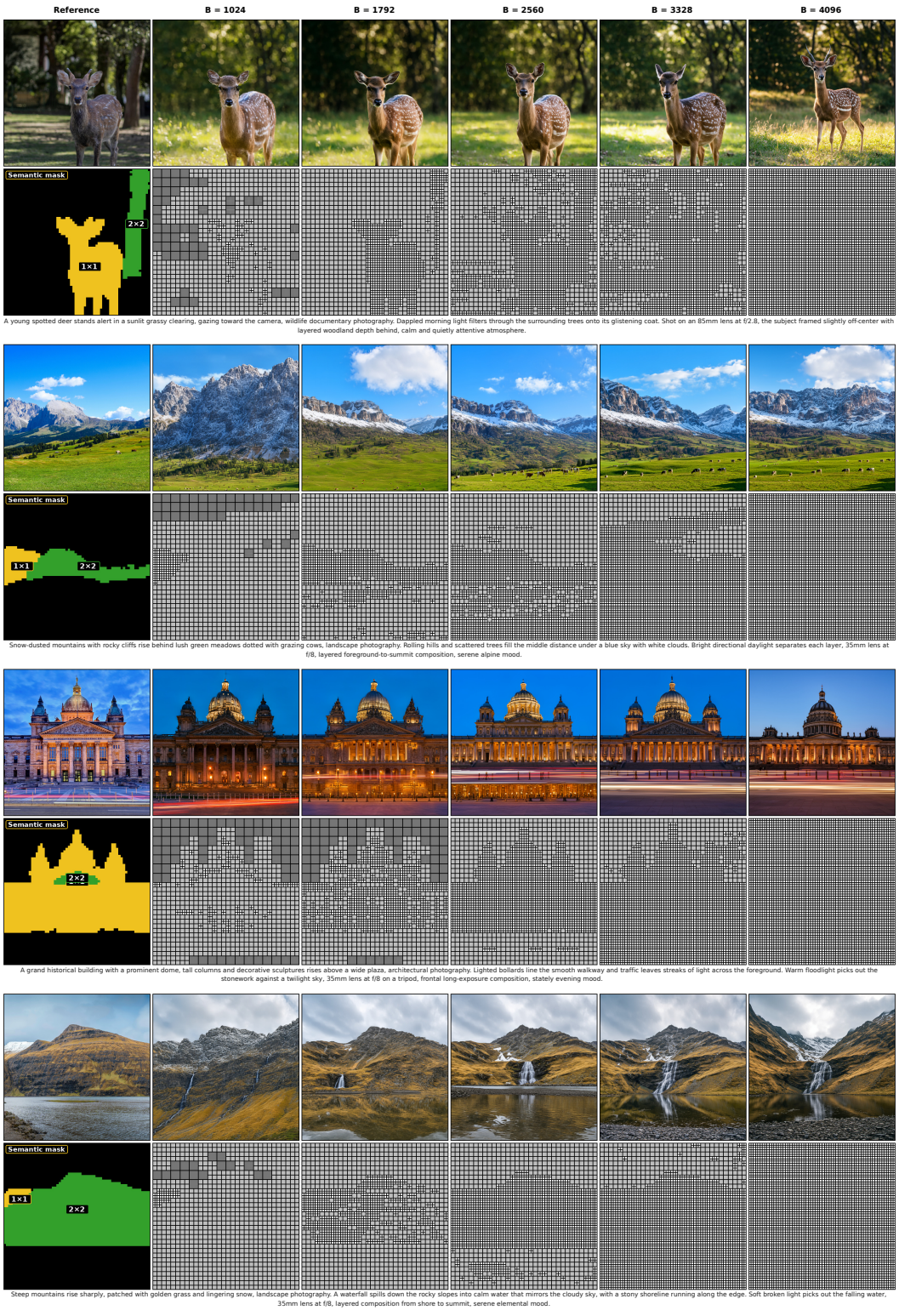


Figure 10: **Mask-guided Image Generation with Varying Token Budget.**

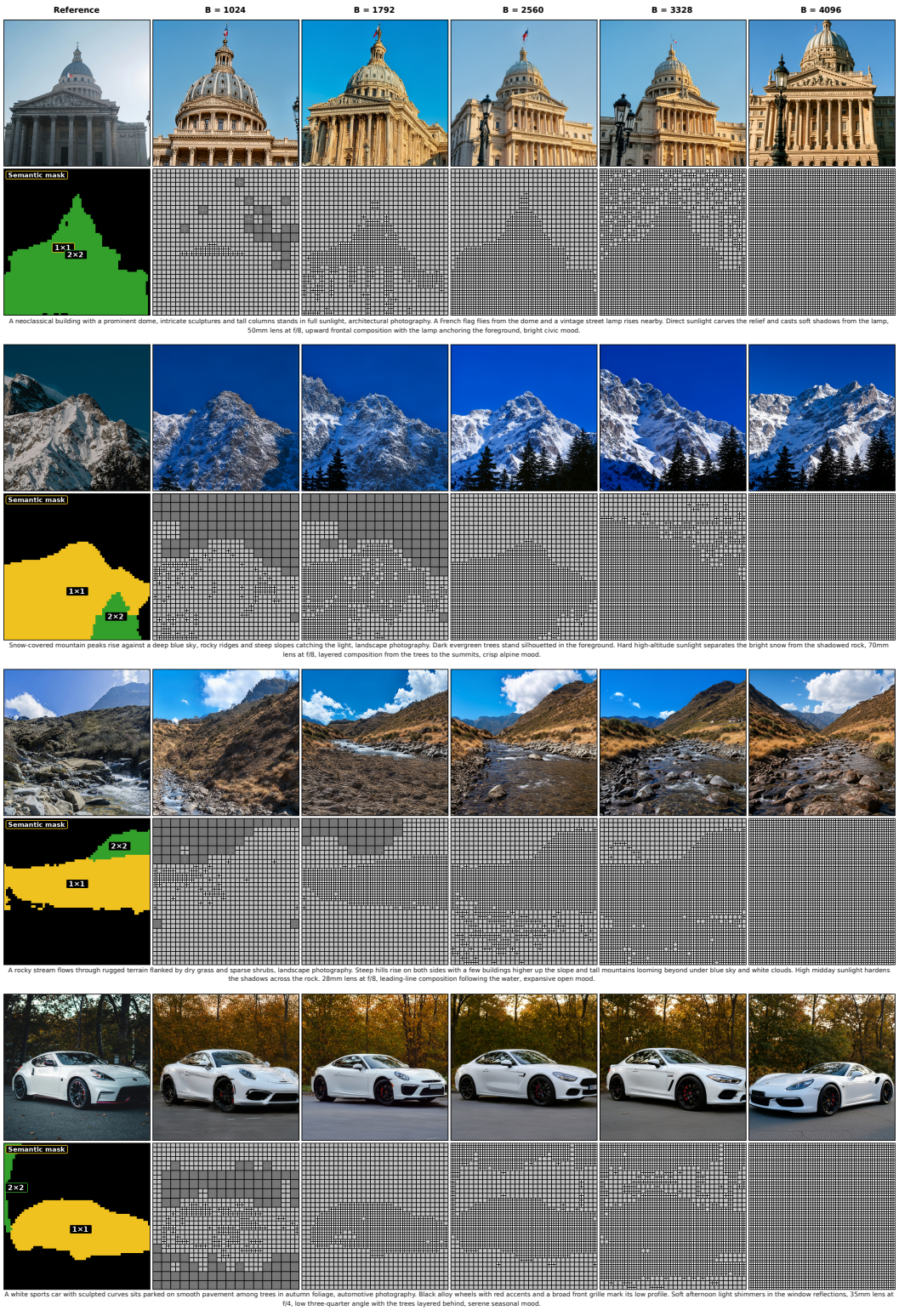


Figure 11: **Mask-guided Image Generation with Varying Token Budget.**

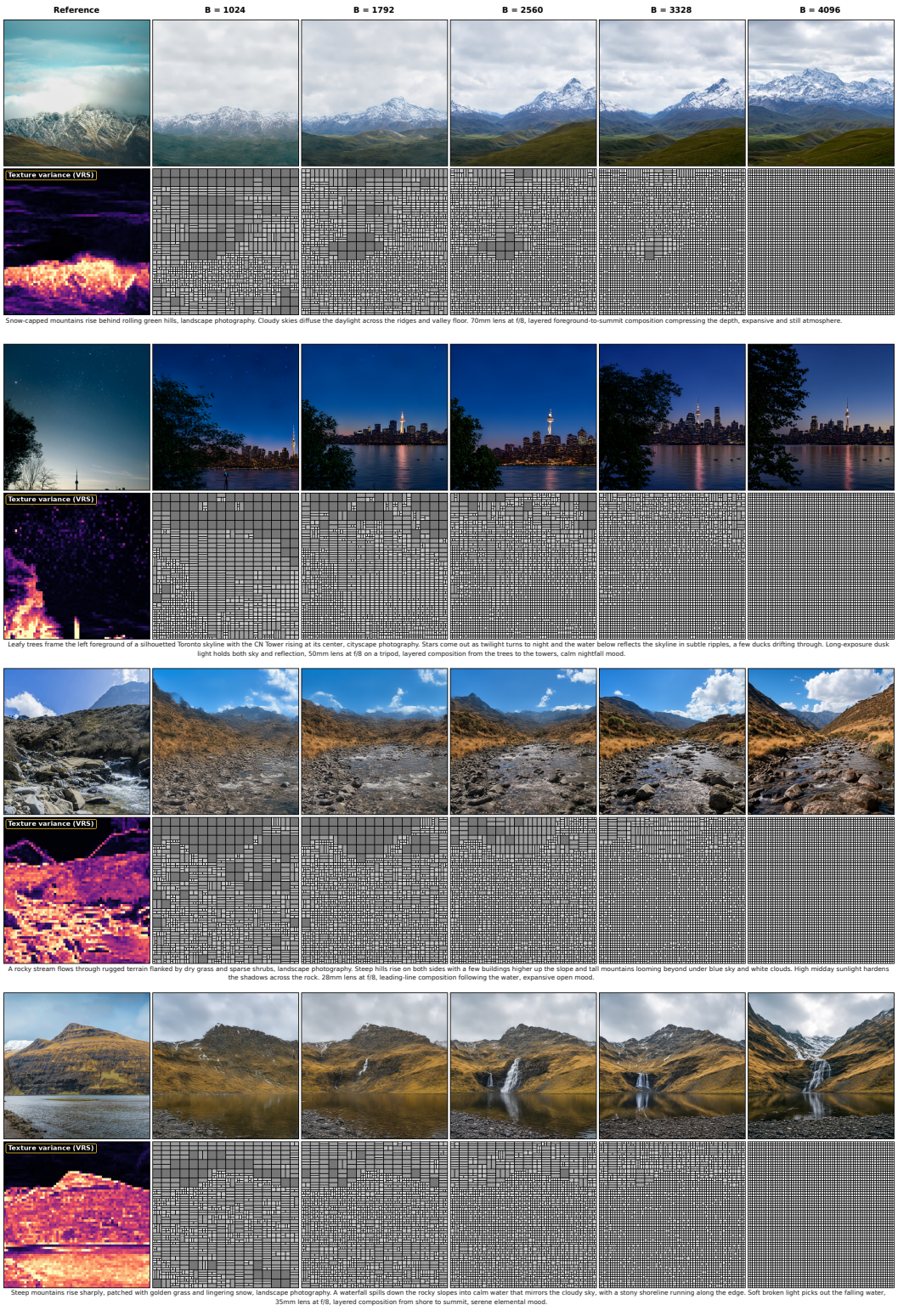


Figure 12: **VRS-guided Image Generation with Varying Token Budget – Part 1.**

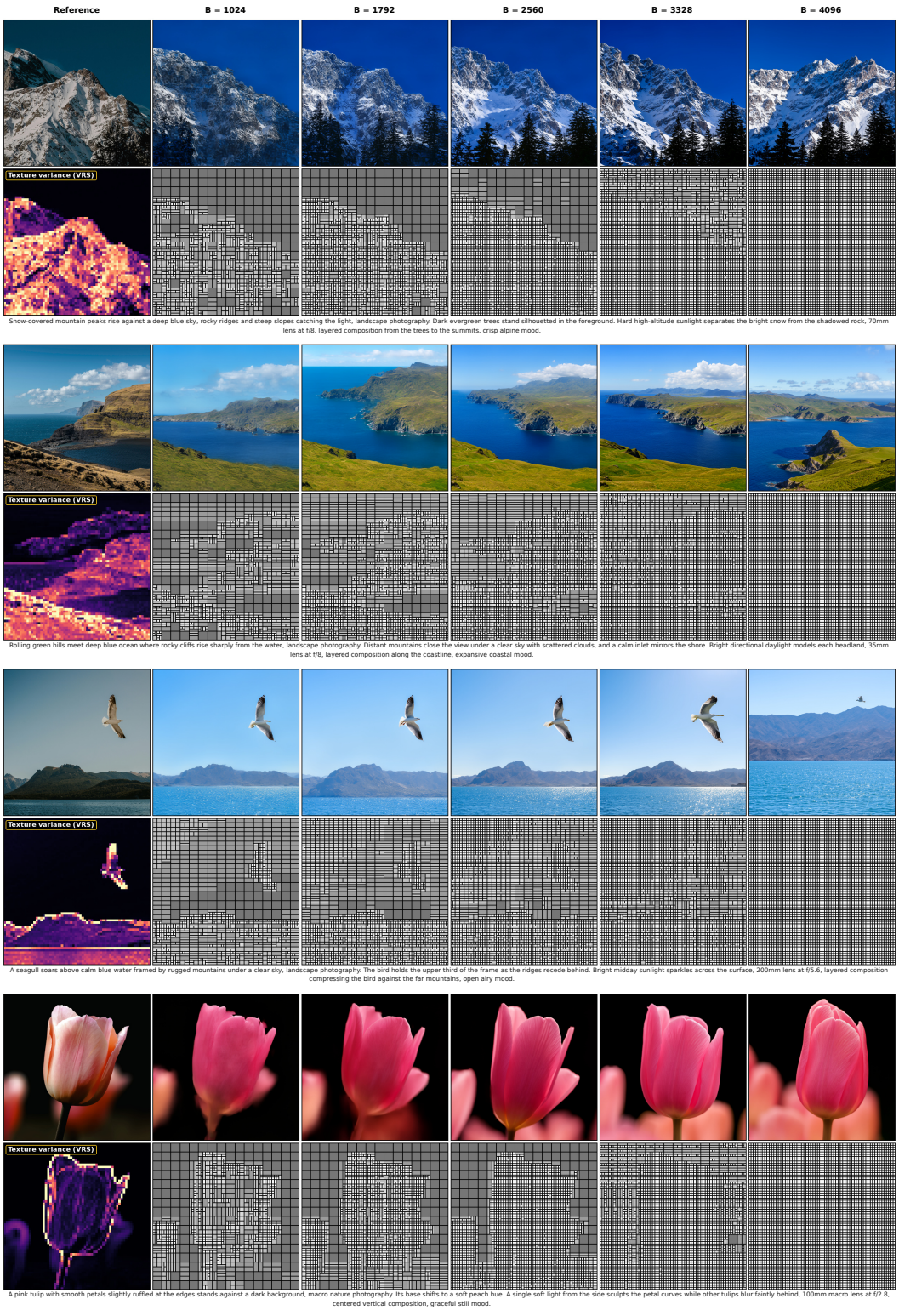


Figure 13: **VRS-guided Image Generation with Varying Token Budget – Part 2**

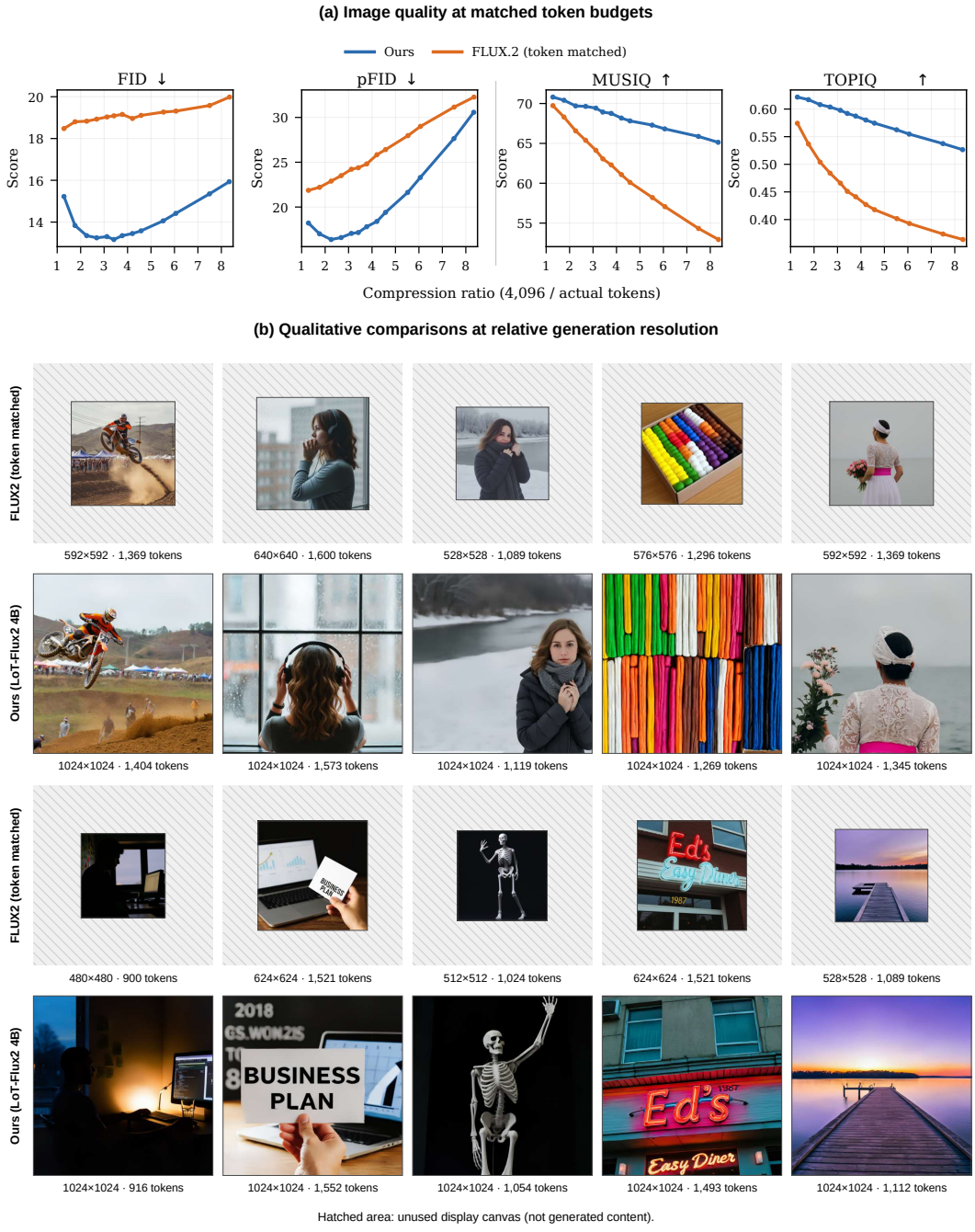


Figure 14: **Image Quality at Matched Token Budgets.** (a) Quantitative comparison. (b) Ten qualitative pairs, with token-matched outputs shown at their relative generation resolution. Hatched areas are display padding, not generated content.

### A.8 Additional Layout-Adaptive Image Generation Results

Figures 15–18 show additional image generation examples generated by LoT-FLUX.2-4B using LoT layouts derived from different sources. Bounding boxes and semantic masks concentrate fine tokens in selected regions, where the model places salient subjects and detailed structures. Texture-variance layouts support spatially varying detail, while depth-derived layouts favor sharp content near the focal region and smoother, defocused surroundings. More layout-adaptive image generation results from LoT-FLUX.2-4B and 9B are available in Section 2 of the supplementary website.

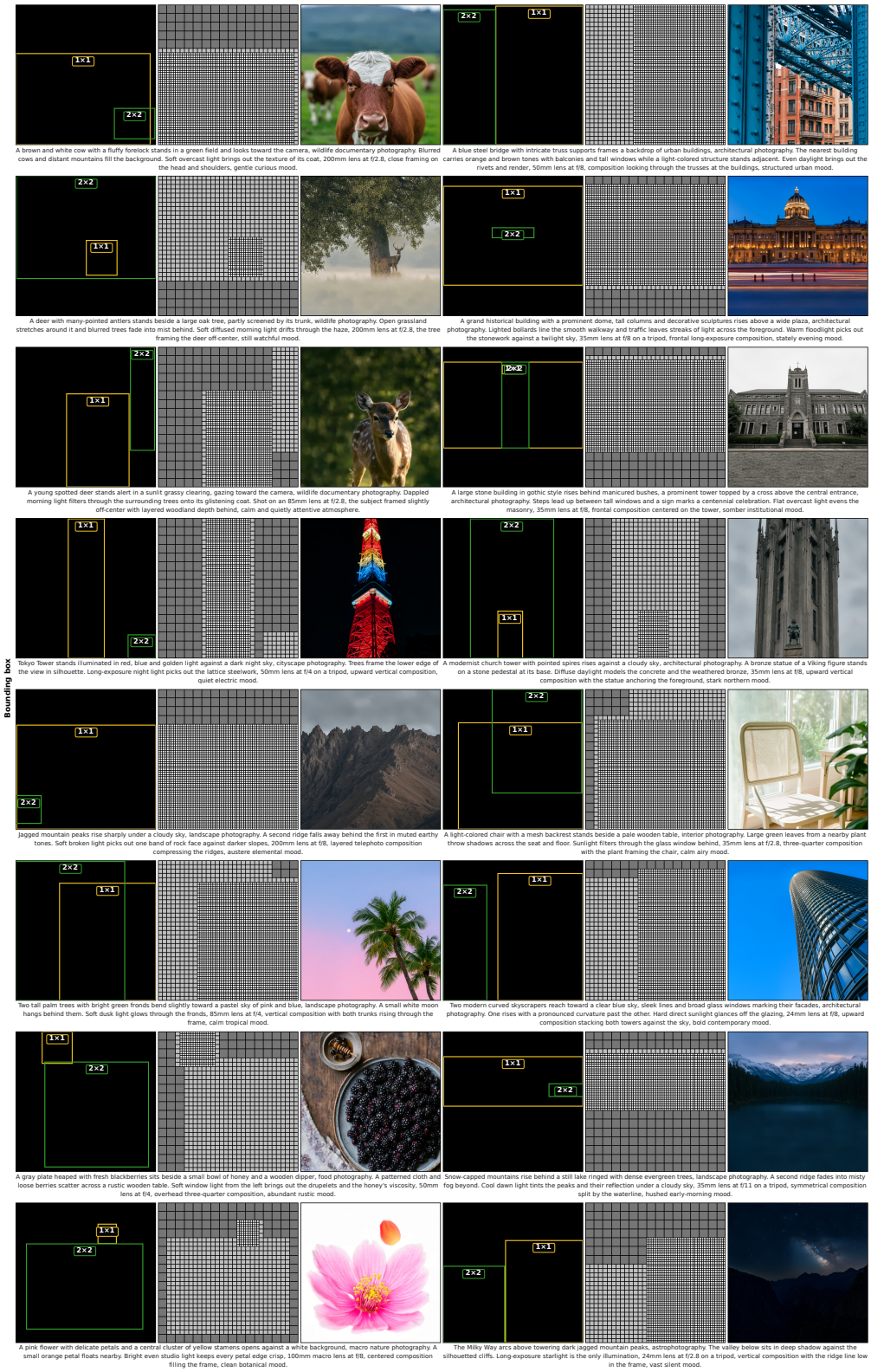


Figure 15: **Bounding-box–controlled Generation Results**

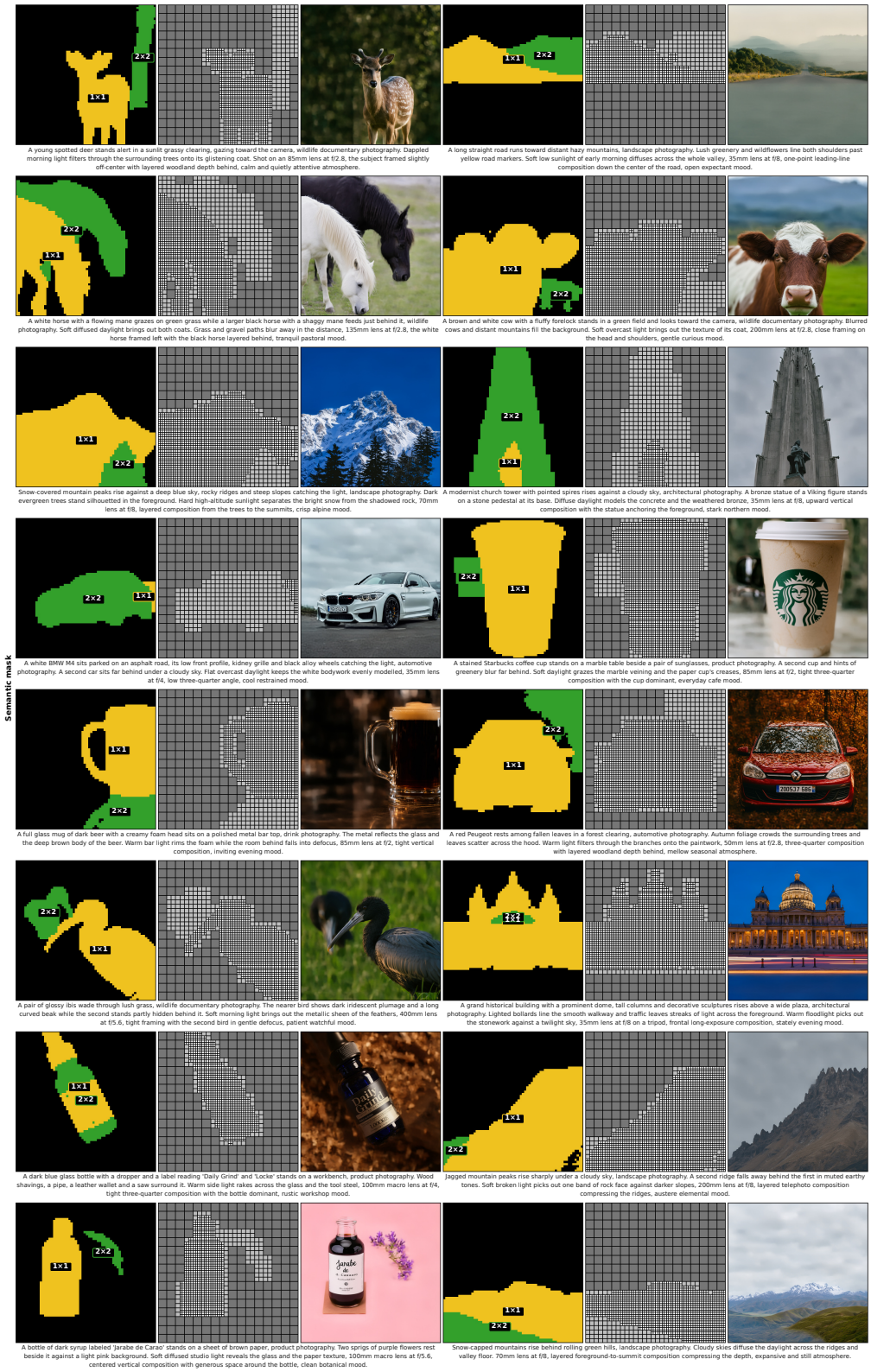


Figure 16: **Semantic-mask–controlled Generation Results**

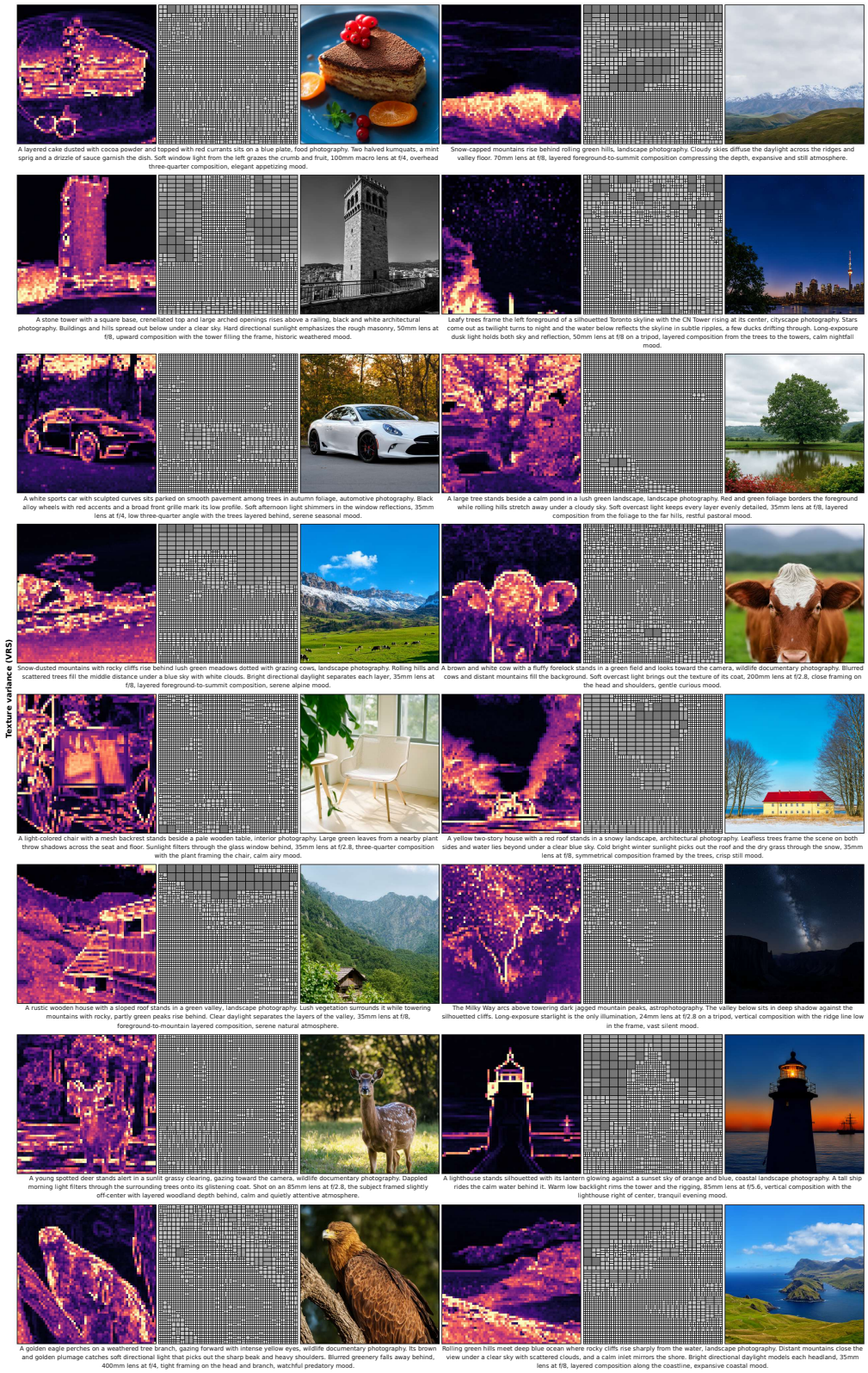


Figure 17: **Texture-variance–controlled Generation Results**

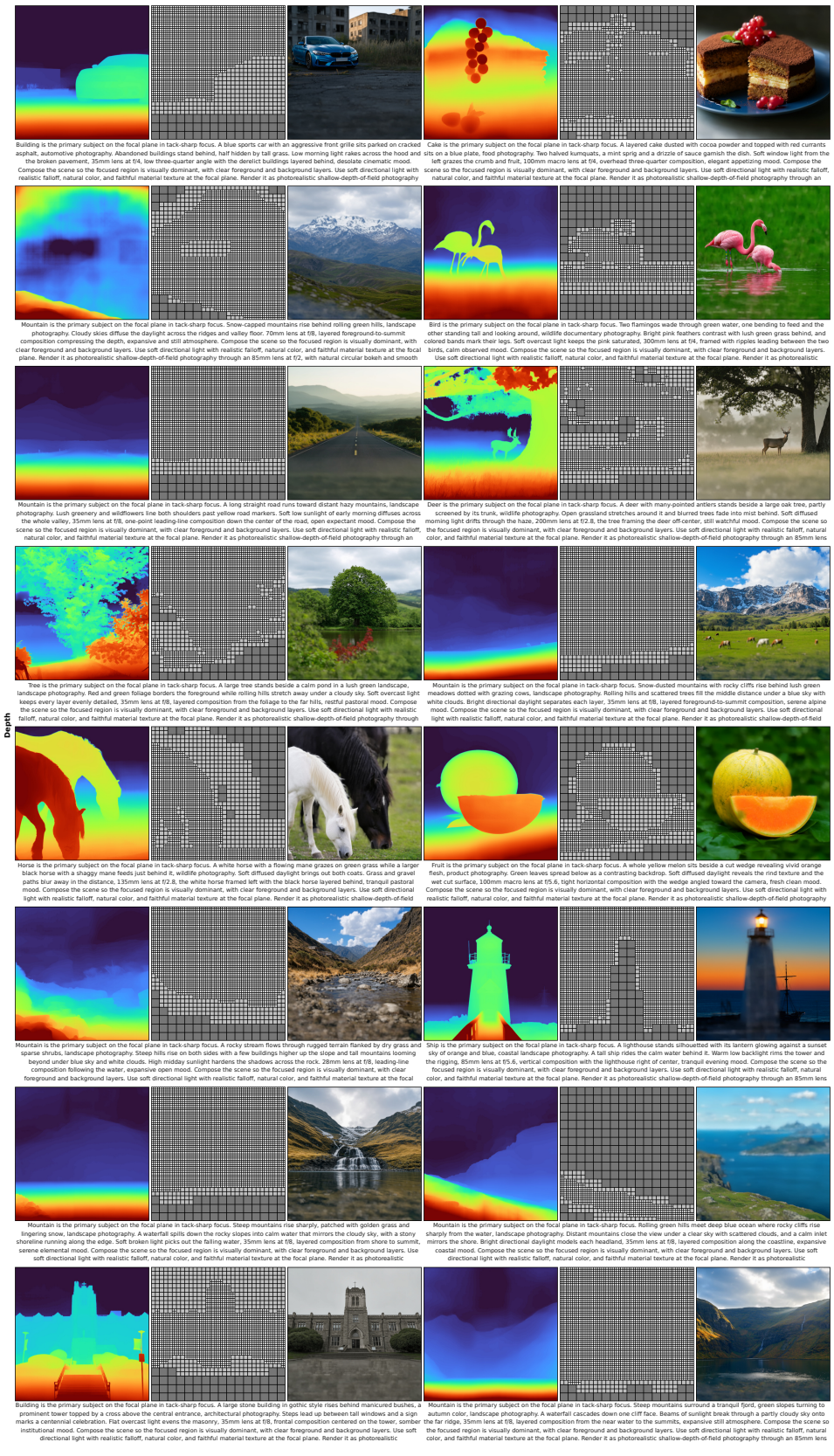


Figure 18: **Depth-of-field–controlled Generation Results**

### A.9 Layout Controllability Compared with Dense Generation

We evaluate whether changing the token layout changes where the generated subject appears, comparing the mixed-source LoT-FLUX.2-4B and 9B models with their corresponding pretrained full-resolution FLUX.2 models.

**Experimental setup.** We select 2,096 prompts from COCO-10k that describe one main subject and have an unambiguous corresponding COCO bounding box. For each prompt, we construct two layouts: one using the reference box and one randomly translating it while preserving its size. We allocate $1 \times 1$ tokens inside the box and $2 \times 2$ tokens outside; the paired layouts have identical token counts. Both model sizes use the same prompts, layouts, and seeds. LoT generates both images with the same prompt and initial noise, changing only the layout. Dense FLUX.2 uses the same prompt and initial noise but receives no layout; its single output is evaluated against both target regions. This comparison measures the additional spatial adaptability afforded by the token layout, rather than comparing two layout-conditioned models.

**Metrics and results.** We use SAM3 (Carion et al., 2025) with the subject name, without providing the target box, and retain the highest-confidence subject mask above a confidence threshold of $0.5$. *Subject detection* is the percentage of images with a detected subject. *Target-region coverage* is the fraction of the detected subject mask lying inside the target box, $|M \cap R|/|M|$, with zero assigned when no subject is detected. Tab. 4 shows higher coverage for LoT under both layouts, with overall gains of 15.79 and 21.89 percentage points for 4B and 9B, respectively. Subject detection is slightly lower for LoT at 4B and comparable at 9B. Thus, subjects more often occupy the fine-token region when generation can adapt to the supplied layout.

Figures 19 and 20 provide selected qualitative comparisons. With the prompt, initial noise, and token count fixed, relocating the fine-token region changes the spatial placement of the subject in LoT's outputs; the dense result is shown alongside for reference.

| | | Target-region coverage | | |
|---|---|---|---|---|
| Method | Subject detection | Original | Relocated | Overall |
| FLUX.2 4B (full res.) | **98.57** | 53.52 | 37.24 | 45.38 |
| LoT-FLUX.2-4B | 98.12 | **68.84** | **53.50** | **61.17** |
| FLUX.2 9B (full res.) | 98.95 | 52.98 | 36.90 | 44.94 |
| LoT-FLUX.2 9B | **99.07** | **74.74** | **58.92** | **66.83** |

Table 4: **Layout Controllability Compared with Dense Generation.** All values are percentages (higher is better).

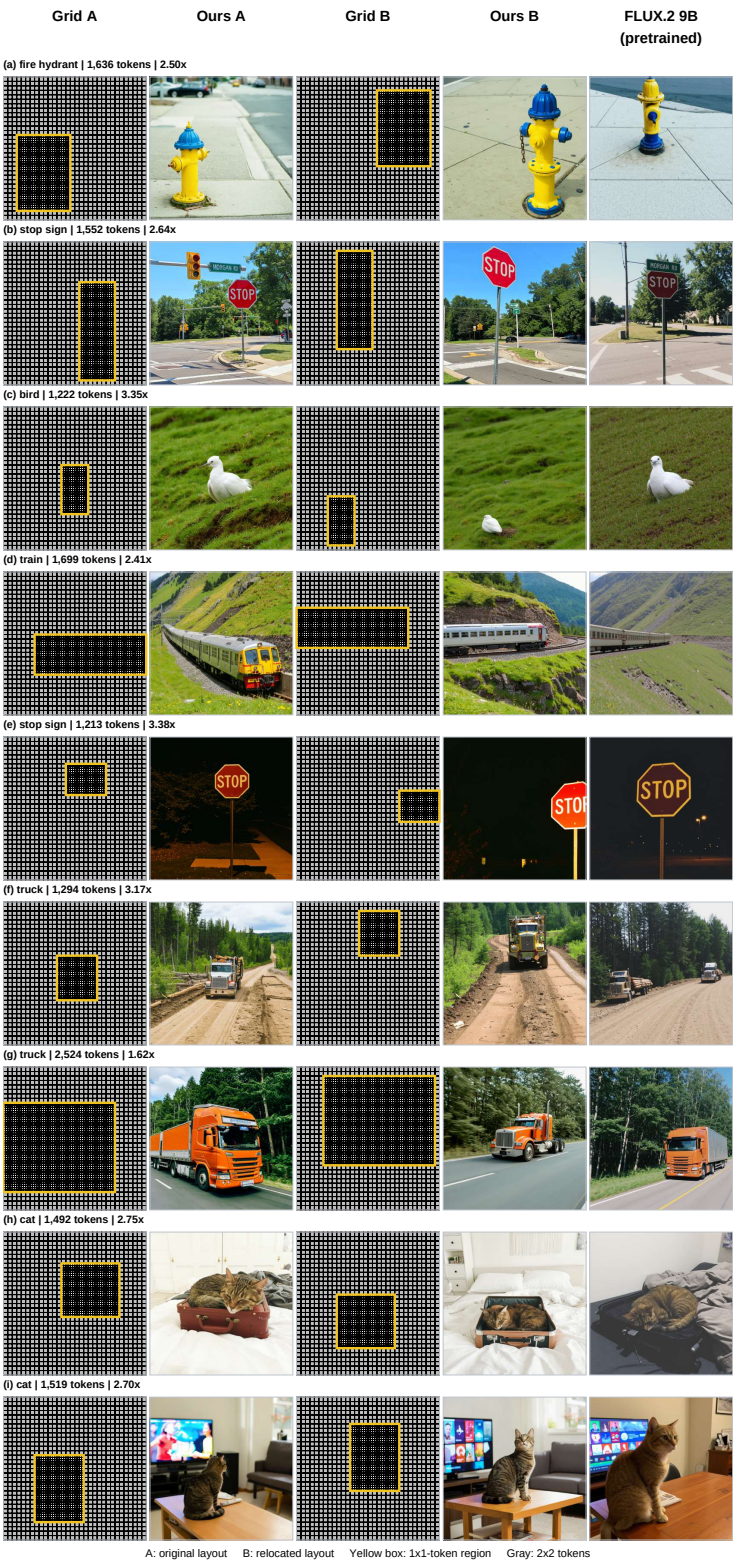


Figure 19: **Layout Controllability: LoT-FLUX.2 9B and Dense FLUX.2 9B.** Selected comparisons (1/2): original grid (A), ours (A), relocated grid (B), ours (B), and pretrained FLUX.2 9B. Each row shares the prompt and initial noise; A and B also share the token budget. Yellow boxes mark the $1 \times 1$-token regions in the layouts. Dense FLUX.2 receives no layout.

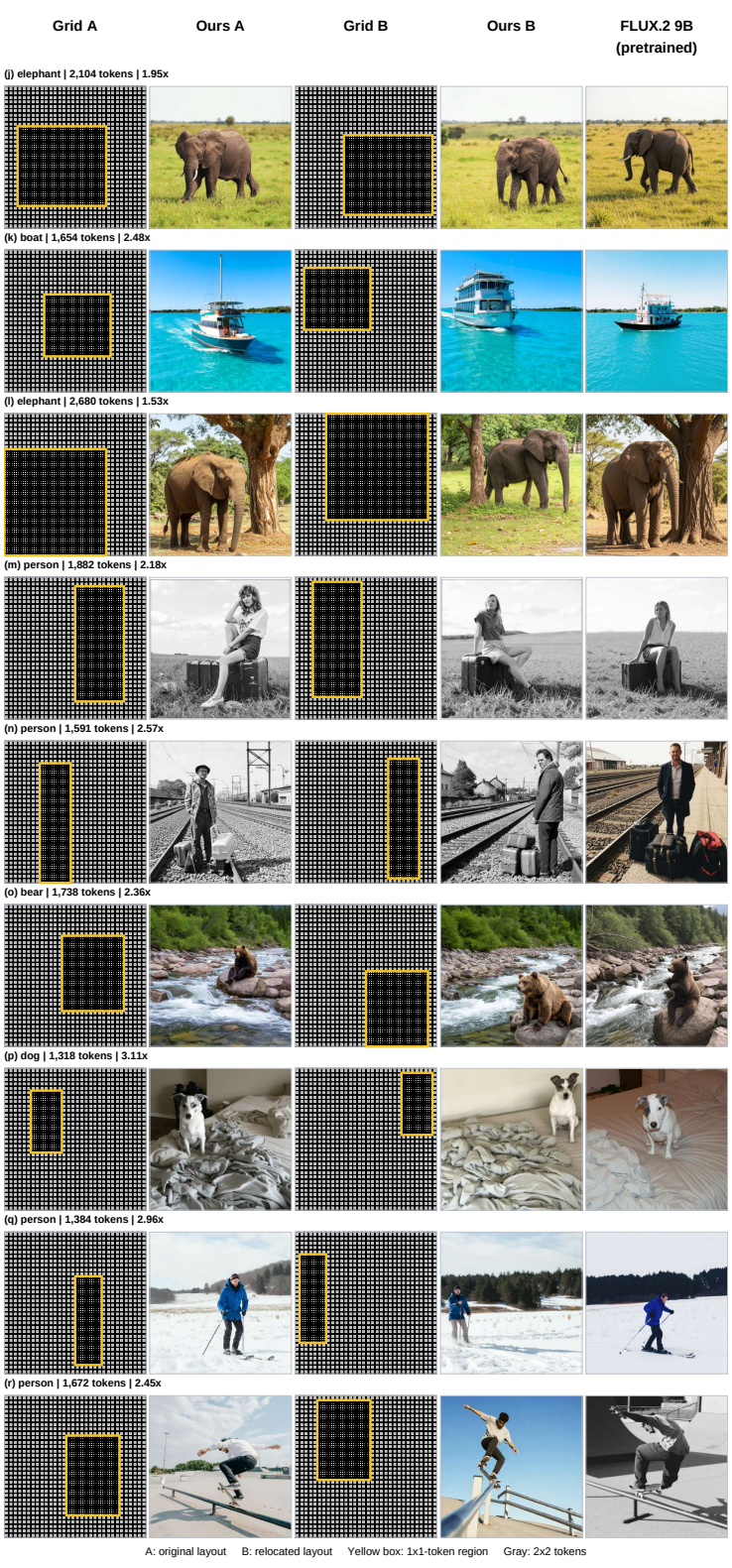


Figure 20: **Layout Controllability: LoT-FLUX.2 9B and Dense FLUX.2 9B.** Selected comparisons (2/2), following the same arrangement and protocol as Figure 19.

### A.10 Video Generation Training Details

We fine-tune Wan2.1-T2V-14B (Team Wan et al., 2025) with its native latent normalization and frozen UMT5-XXL text encoder and VAE.

**Training data and source mixture.** We use 81-frame clips from Vchitect-T2V-Dataverse (Fan et al., 2025). The layout mixture comprises semantic masks or bounding boxes (50%, split equally), VRS layouts (25%), and DoF layouts (25%). All training clips use $1280 \times 720$ resolution. Grids retain $e_t = 1$ and $e_h, e_w \in \{1, 2, 4\}$, with at most 3,072 tokens per latent time slice to avoid out-of-memory errors.

**Training setup.** We reuse the LoRA rank/dropout, AdamW coefficients, adapter learning rate and weight decay, clamped clean-data loss, and EMA schedule from Section A.4. Wan-specific LoRA targets are the query/key/value/output projections in self- and cross-attention, the feed-forward linear layers, and the timestep embedding/projection layers. We fully fine-tune the per-extent input embeddings, output heads, and patch-size MLP at $10^{-5}$, and the final output normalization and modulation at $10^{-4}$. These fully fine-tuned groups have zero weight decay. We train for 4,000 steps with a total batch size of 32 videos, using FSDP2, gradient checkpointing, and gradient-norm clipping at 200. Unlike the image model's length-dependent dynamic shift, video training uses a fixed noise-schedule shift factor of 5, shared by all tokens within each clip.

### A.11 Video Generation Baseline Comparison Details

**Inference parameters.** For all baselines, we generate with 50 sampling steps, classifier-free guidance scale 4, orthogonal guidance strength 1, and a noise-schedule shift factor of 5. We evaluate on 200 held-out DataVerse test prompts, generating 81-frame videos at $1280 \times 720$ and 15 fps. We use the test prompts to generate reference videos with MiniMax-H3 (MiniMax, 2026) and derive the evaluation VRS layouts from these generated videos, rather than the original held-out clips. The original clips were mostly static, so we use generated references to evaluate temporally varying token allocations. Only the derived layouts, not the reference RGB videos, are supplied to the evaluated models. At each token budget, ours, Foveated Diffusion, and ToMe-SD use the same VRS-derived grids, prompts, and seeds.

**Training Details.** We adapt Foveated Diffusion (Chao et al., 2026) to Wan2.1 14B and the same VRS-derived token layouts as ours, with spatial extents up to $4 \times 4$. We evaluate our VRS-only model and Foveated Diffusion at their 1,000-step checkpoints. ToMe-SD (Bolya & Hoffman, 2023) requires no training: we adapt token merging to pretrained Wan2.1 14B by mean-merging tokens within each VRS leaf for self-attention, cross-attention, and MLP computation, then copy-unmerging the outputs before adding them to the full-resolution residual stream. Reported Full Res. Wan2.1 14B is fine-tuned using the same training budget and parameters on the same dataset for fairness.

**Metric details.** We evaluate six VBench (Huang et al., 2024) dimensions using its custom-video evaluation on our held-out prompts. For image quality, aesthetic quality measures visual appeal, while imaging quality assesses perceptual degradation such as blur and noise. For temporal consistency, subject consistency measures the stability of subject appearance across frames, while background consistency measures the stability of the scene background. For motion, motion smoothness measures the continuity of movement, while dynamic degree measures the prevalence of substantial motion. Higher quality, consistency, and smoothness scores are better; a higher dynamic degree indicates more motion, not necessarily more realistic motion.

For efficiency, runtime is the mean H100 generation time per video that includes both denoising and VAE decoding, similar to the image generation case.

### A.12 Additional Video Results

Section 4 of the supplementary website demonstrates custom video layout control: users paint a static background detail map and keyframed detail maps to specify where fine and coarse tokens are allocated as subjects move.

| Method | Compression | Time (s) ↓ | Speed ↑ | IR ↑ | HPSv2.1 ↑ | HPSv3 ↑ | FID ↓ | pFID ↓ | TOPIQ ↑ | MUSIQ ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| Ours w/o size embedding | 1.523× | 4.372 | 1.322× | **0.9294** | 0.2717 | 10.0284 | 14.584 | 17.472 | 0.6030 | 69.433 |
| Ours w/ direct HR prediction | 1.523× | 4.264 | 1.355× | 0.6301 | 0.2601 | 7.0595 | 26.487 | 36.072 | 0.5134 | 63.346 |
| Ours w/ binary grid | 1.518× | **4.167** | **1.387**× | 0.7472 | 0.2520 | 8.1955 | 20.603 | 25.073 | 0.5517 | 66.443 |
| Ours | 1.523× | 4.366 | 1.323× | 0.9105 | **0.2764** | **10.3477** | **13.800** | **17.019** | **0.6170** | **70.410** |
| Ours w/o size embedding | 1.893× | 3.718 | 1.554× | **0.9008** | 0.2665 | 9.7299 | 13.934 | 17.320 | 0.5888 | 68.563 |
| Ours w/ direct HR prediction | 1.893× | 3.609 | 1.601× | 0.4335 | 0.2503 | 5.3532 | 37.414 | 55.443 | 0.4581 | 59.576 |
| Ours w/ binary grid | 1.879× | **3.501** | **1.651**× | 0.7086 | 0.2478 | 7.8377 | 21.638 | 27.603 | 0.5180 | 64.668 |
| Ours | 1.893× | 3.704 | 1.560× | 0.8833 | **0.2728** | **10.1211** | **13.361** | **16.393** | **0.6080** | **69.691** |
| Ours w/o size embedding | 2.406× | 3.137 | 1.842× | 0.8337 | 0.2590 | 9.1663 | 14.378 | 18.822 | 0.5758 | 67.968 |
| Ours w/ direct HR prediction | 2.406× | 3.071 | 1.882× | 0.1066 | 0.2351 | 2.8754 | 54.762 | 89.686 | 0.3876 | 54.603 |
| Ours w/ binary grid | 2.336× | **2.958** | **1.953**× | 0.6952 | 0.2464 | 7.6611 | 23.111 | 31.550 | 0.4871 | 63.340 |
| Ours | 2.406× | 3.178 | 1.818× | **0.8513** | **0.2672** | **9.7533** | **13.308** | **17.046** | **0.5978** | **69.420** |
| Ours w/o size embedding | 2.935× | 2.789 | 2.071× | 0.8042 | 0.2543 | 8.7670 | 14.933 | 20.497 | 0.5636 | 67.277 |
| Ours w/ direct HR prediction | 2.935× | 2.704 | 2.137× | -0.0843 | 0.2268 | 1.3967 | 68.508 | 111.952 | 0.3497 | 51.809 |
| Ours w/ binary grid | 2.774× | **2.634** | **2.194**× | 0.6793 | 0.2460 | 7.5753 | 24.004 | 34.786 | 0.4674 | 62.508 |
| Ours | 2.935× | 2.832 | 2.040× | **0.8326** | **0.2639** | **9.5522** | **13.345** | **17.822** | **0.5873** | **68.760** |

Table 5: **Ablation Study Across Different Token Budgets.**

Additional qualitative video generation results are available in Section 5 of the supplementary website: Section 5.1 shows depth-of-field layouts, Section 5.2 shows mask-based layouts (semantic masks and bounding boxes), and Section 5.3 shows texture-variance layouts (variable-rate shading). Each example pairs its source-map/tokenization visualization with the generated video. Section 6 (Video Generation Qualitative Baseline Comparisons) provides synchronized comparisons with the full-resolution model and the baselines, together with per-method token compression and generation time.

### A.13 Additional Ablation Study Results

Tab. 5 reports ablation results across different token budgets, extending the main-paper results in Tab. 3, which reports only a single threshold setting at approximately $1.52\times$ token compression. We use the same training and evaluation setup as the image generation baseline comparison. We introduce the ablation variants considered below.

**Ours w/ direct HR prediction.** We bypass the analytic patch-wise asymmetric flow recovery in equation 9, introduced in Section 4.1, and directly interpret the per-extent output heads as the full-resolution flow prediction. The compressed input representation, transformer, and patch-size conditioning are retained. The heads are trained against the full-resolution flow objective without the analytic recovery step.

**Ours w/ binary grid.** We restrict the layout to square $1 \times 1$ and $2 \times 2$ tokens. This replaces the multilevel, potentially anisotropic layout during both training and evaluation; patch-size conditioning and asymmetric flow recovery remain unchanged.

**Ours w/o size embedding.** We disable the learned patch-size embedding introduced in Section 4.2, removing explicit extent-dependent conditioning of the transformer. We retain the extent-specific input and output heads, single-scale RoPE at each token's geometric center, and asymmetric flow recovery. This isolates the effect of explicitly informing the transformer of each token's spatial extent.

We show image generation qualitative results of the architectural ablation studies in Fig. 21.

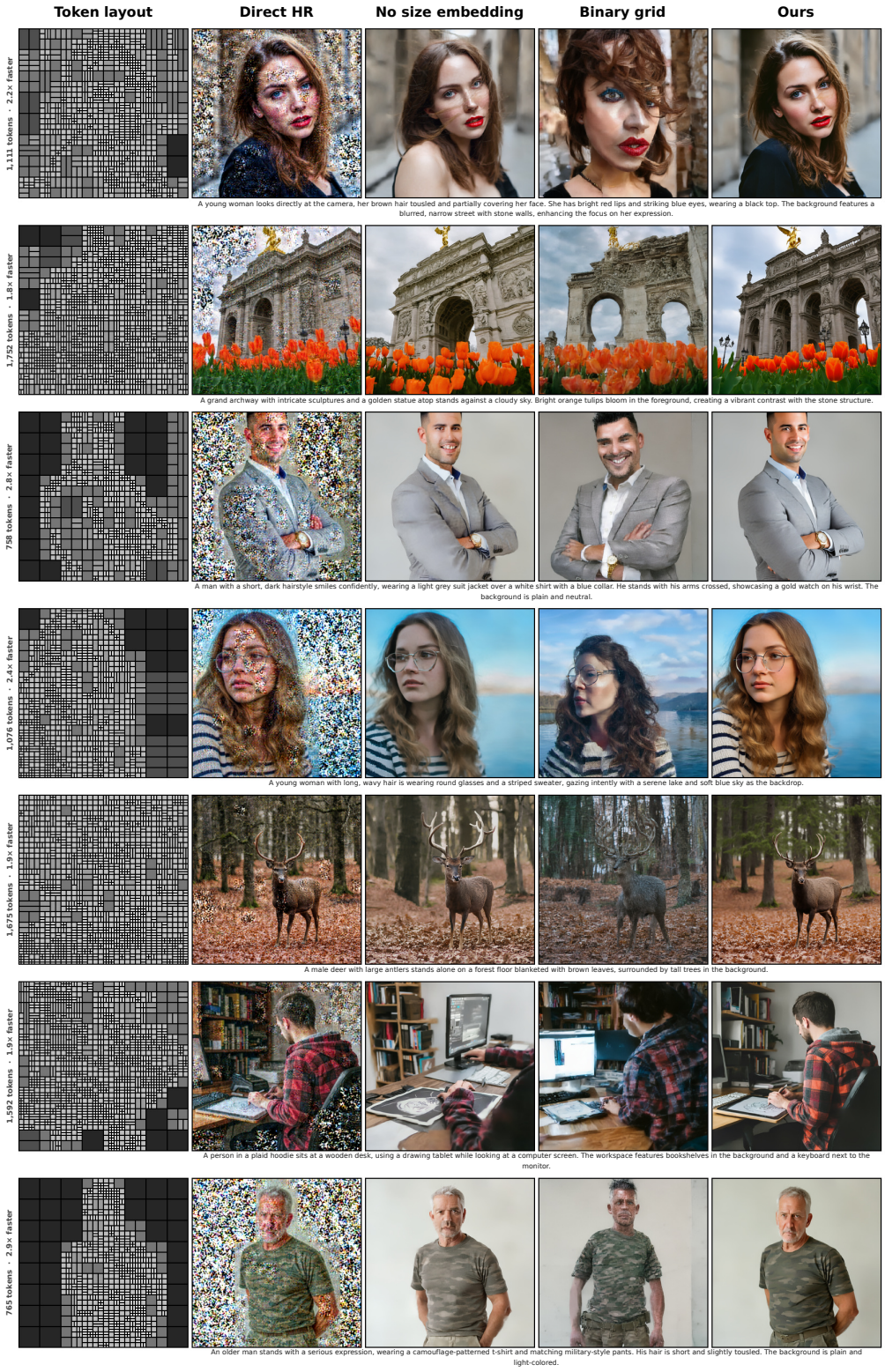


Figure 21: **Ablation Study Qualitative Results.**

### A.14 LIMITATIONS

LoT gains efficiency by allocating fewer tokens to regions that can be synthesized more coarsely. When fine structures or textures extend across most of an image, as in dense musical notation or close-up views of grass or fabric, preserving them may require a fine token layout throughout, leaving limited room for computational savings. Enforcing a lower token budget in these cases can degrade structural fidelity. As shown in Fig. 22, the lower-budget result exhibits more fragmented and distorted staff lines than the higher-budget result generated with the same prompt and noise seed. Thus, the achievable savings depend on the scene's detail requirements, and layout adaptation does not guarantee faithful fine structures under aggressive token reduction.

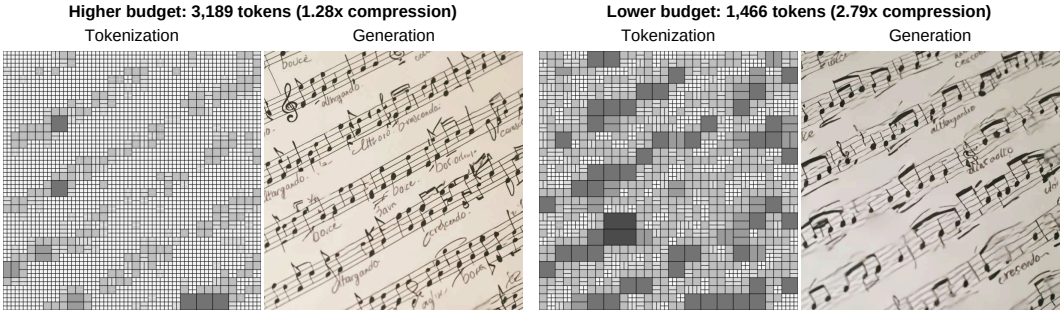


Figure 22: **Limitation: Fine-Structure Degradation at Lower Token Budgets.** The lower-budget result shows more broken staff lines.