# POLYLORA: DISCRETE-CONTINUOUS TRANSVERSAL UPDATES FOR QUANTIZED DIFFUSION ADAPTATION

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

## ABSTRACT

Large-scale diffusion models are often customized via low-rank adaptation and then quantized for deployment to reduce memory costs. However, a fundamental mismatch remains. The adapter is optimized in continuous space but deployed after discrete integer merge. This leads to two failures. First is semantic facet sticking, where updates that reduce the training loss may not cross a quantization boundary and leave the deployed model unchanged. Second is asymmetric merging distortion, where even when a boundary is crossed, the base integer anchor non-uniformly suppresses or amplifies the update. We propose PolyLoRA to reshape the adaptation trajectory along transversality in polyhedral space for deployable semantic changes. It introduces three lightweight mechanisms. Forward-Aligned Crossing promotes transverse boundary crossing in the actual pre-merge adapter, directly countering facet sticking. Merge-Anisotropy Calibration weights these crossings according to the integer-merge geometry, steering updates toward coordinates where post-merge leverage is strong to mitigate distortion. Semantic Transversality Shaping further uses an activation-derived Jacobian-Gram proxy to align merge-retained displacements with denoising-relevant response patterns, ensuring that survived updates actually benefit generation. These mechanisms are coupled with the denoising loss via a dual-barrier strategy at no inference overhead. By enforcing transversality, PolyLoRA ensures that optimizer updates cross quantization boundaries and survive integer merge in semantically relevant directions. Under W4A8 DreamBooth, PolyLoRA further improves DINO and CLIP-I over IntLoRA by 0.0162 and 0.0083, respectively, while preserving the direct integer-merge inference path. The code is available at `https://anonymous.4open.science/r/PolyLora_code-2B75/`.

## 1 INTRODUCTION

Large-scale text-to-image diffusion models Podell et al. (2024); Peebles & Xie (2023) have shown strong generation capability in personalized visual tasks, including subject-driven generation Chen et al. (2024); LI et al. (2023), style customization Sohn et al. (2023); Frenkel et al. (2024), and controllable image synthesis Li et al. (2023b); Mou et al. (2024). Using rich visual priors encoded in pretrained models, these systems can produce high-quality customized content from a small number of examples. However, the increasing scale of pretrained diffusion models has made practical deployment increasingly expensive in computation and memory. Although full-parameter fine-tuning provides strong adaptation quality, its high memory footprint and training cost restrict the rapid iteration and multi-task reuse on consumer-level GPUs Shah et al. (2024). For example, loading the FP32 weights of FLUX.1-dev requires 24+GB GPU memory. Parameter-efficient fine-tuning has therefore become an emerging approach for diffusion model customization Qiu et al. (2023). Among these methods, LoRA Hu et al. (2022) reduces trainable parameters via low-rank decomposition, enabling lightweight deployment and making it a standard choice.

Recent quantized LoRA methods Xu et al. (2024); Li et al. (2024); Guo et al. (2025) further push this line of work by maintaining low-bit adaptation during training and supporting direct integer merge during inference. This design reduces the deployment discontinuity introduced by a separate PTQ stage and can achieve competitive performance on customization tasks while lowering training and inference costs Kim et al. (2023). Despite these advantages, very low-bit adaptation still exposes a



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.

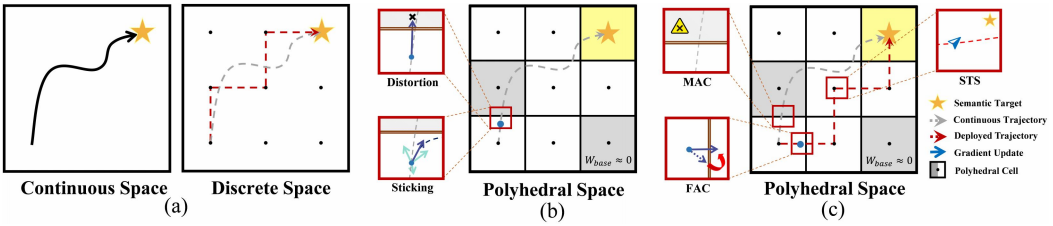



Figure 1: (a) Optimization path deformation when transitioning from continuous to quantized discrete spaces; (b) Reshaping the adaptation trajectory along transversality in polyhedral space reveals two deployment failures, semantic facet sticking and asymmetric merging distortion; (c) PolyLoRA enforces transverse crossings over cell boundaries during optimization through MAC, FAC, and STS, ensuring discrete, merge-retained, and semantically relevant parameter updates.

structural conflict between continuous optimization and discrete deployment. A low-rank adapter is optimized in a continuous parameter space, whereas the deployed model is determined by quantized integer states after merge. The optimization path shifts from its continuous optimum, as shown in Fig. 1 (a), causing updates that reduce training loss to either leave the deployed weights unchanged or produce distorted effects after merge.

Existing methods mitigate this conflict from three directions. Scale-aware adaptation He et al. (2024) improves the learnability of small-magnitude updates, but they only provide numerical compensation without explicitly modeling whether updates cross the discrete boundaries of the deployed quantizer. Consequently, they amplify both useful signals and ineffective perturbations. Merge-friendly LoRA quantization methods Xu et al. (2024) improve the compatibility between low-rank adaptation and quantized weights, yet they are designed for general low-bit LoRA and do not account for the semantic gap induced jointly by multi-timestep diffusion denoising and direct integer merge. Integer adaptation methods such as IntLoRA Guo et al. (2025) reduce arithmetic inconsistency between training and deployment via integer adapters and multiplicative merge. However, the base integer anchor introduced to preserve merge also makes update effects strongly anisotropic across coordinates, further widening the mismatch between continuous low-rank optimization and discrete deployment.

To clarify this deployment failure, we reinterpret quantized LoRA from a polyhedral transversality perspective as trajectory optimization over a cell complex induced by adapter quantization and integer merge. As illustrated in Fig 1 (b), the quantizer partitions the high-dimensional adapter-code space into discrete polyhedral cells. LoRA restricts the search trajectory to a low-dimensional subspace, while diffusion denoising provides a time-dependent optimization field across noise levels. Together, these three factors force the adaptation trajectory to move inside a discretized low-rank geometry, where a reduction in continuous loss does not necessarily translate into a deployable semantic change. Two failure modes emerge. Semantic facet sticking: the optimizer shifts the continuous code within a quantization bin without changing the rounded integer state. Training loss drops, but the deployed adapter is untouched. Asymmetric merging distortion: even when a boundary is crossed, the base integer anchor scales each coordinate differently. Updates on weakly-anchored coordinates (*e.g.*, anchor=1) are suppressed after merge, while those on strong anchors (*e.g.*, 127) are amplified. Hence, the same continuous change can have drastically different deployed effects.

Motivated by the above analysis, we propose PolyLoRA, a framework that reshapes the adaptation trajectory to achieve deployable semantic changes (Fig. 1 (c)). Rather than indiscriminately increasing the magnitude of continuous LoRA updates, PolyLoRA introduces two mechanisms that directly target the two earlier failure modes: Forward-Aligned Crossing (FAC) and Merge-Anisotropy Calibration (MAC). Specifically, FAC encourages the continuous pre-merge adapter code to explicitly cross quantization boundaries, ensuring that the update overcomes semantic facet sticking. MAC weights boundary crossings by their expected survival after the integer merge, steering the trajectory toward coordinates where a crossing genuinely affects the deployed weights, thereby addressing asymmetric merging distortion. However, even updates that cross boundaries and survive integer merging may remain irrelevant to diffusion denoising. To this end, PolyLoRA further incorporates Semantic Transversality Shaping (STS), which aligns merge-retained displacements with denoising-sensitive response patterns. STS uses an activation-derived Jacobian-Gram proxy to favor directions that meaningfully change generation output. Together, these three mechanisms ensure learned updates are discrete, merge-retained, and semantically relevant. They are coupled with the standard diffusion loss through a multiplicative dual-barrier strategy, requiring no additional inference-time module.

**Contributions**

- We identify and formally characterize two previously overlooked deployment failures in quantized diffusion adaptation, namely semantic facet sticking and asymmetric merging distortion, through a novel polyhedral transversality lens. This reveals that continuous loss reduction does not guarantee deployable semantic change.
- We propose PolyLoRA, a framework that reshapes the adaptation trajectory to explicitly enforce three necessary conditions for deployability: crossing quantization boundaries, surviving integer merge, and aligning with denoising sensitive directions. Unlike prior methods that only compensate for quantization error, PolyLoRA directly optimizes for discrete, merge retained, and semantically relevant updates.
- We demonstrate that PolyLoRA improves both editability and fidelity under extreme low-bit settings on subject-driven generation and style-customized image generation, achieving state-of-the-art performance across personalized diffusion tasks. Under W4A8 DreamBooth, PolyLoRA improves DINO and CLIP-I over IntLoRA by 0.0162 and 0.0083, respectively, while preserving the direct integer-merge inference path.

## 2 Related Work

**Diffusion Model Quantization** Deploying multi-step diffusion models is notoriously bottlenecked by memory and latency, making quantization a critical area of study. Early post-training quantization (PTQ) frameworks, such as PTQ4DM Shang et al. (2023) and PTQD He et al. (2023), revealed that the severe activation shifts occurring across diffusion timesteps cause naive uniform quantization to fail. To address this, subsequent methods introduced time-step-aware calibration Li et al. (2023a); Huang et al. (2025) to dynamically adjust quantizer parameters. More recently, as architectures shift toward Diffusion Transformers (DiTs), specialized techniques like Q-DiT Chen et al. (2025) and TaQ-DiT Liu et al. (2026) have been developed to handle the extreme dynamic range variations inherent to transformer blocks. While extreme compression paradigms—including binarization Zheng et al. (2025), product quantization Shao et al. (2024), and low-precision floating-point formats Zhao et al. (2025)—have pushed efficiency further, they universally treat the quantized backbone as a static, finalized artifact. Crucially, they overlook how subsequent parameter-efficient adaptation perturbs this carefully calibrated discrete space, often leading to unpredictable generation artifacts.

**Parameter-efficient Customization** To avoid the prohibitive costs of full-parameter fine-tuning, parameter-efficient fine-tuning (PEFT) has emerged as the de facto standard for customizing large-scale diffusion models. Driven largely by Low-Rank Adaptation (LoRA) Hu et al. (2022), a robust ecosystem of methods now exists for subject-driven generation Ruiz et al. (2023; 2024), concept mixing Kumari et al. (2023); Ye et al. (2023), and precise spatial control Zhang et al. (2023); Gandikota et al. (2024). Beyond standard low-rank structures, architectures have been expanded to include token-level embeddings Gal et al. (2023) and more complex convolutional factorizations YEH et al. (2024). However, the success of these methods inherently relies on the assumption of a continuous weight space where infinitely small, high-precision updates can be safely accumulated. When the underlying model is aggressively quantized, this fundamental assumption collapses. Fine-grained continuous updates are routinely erased by discretization grids or severely distorted during weight merging, crippling the expressiveness of the adapter in low-bit deployment scenarios.

**LoRA over Quantized Diffusion Models** The paradigm of training adapters over a frozen, quantized backbone, pioneered by QLoRA Dettmers et al. (2023) in language modeling, has recently been adapted to diffusion models. Efforts to stabilize this pipeline include aligning quantization grids with adapter ranks Xu et al. (2024), optimizing initialization to offset quantization error Li et al. (2024); Guo et al. (2024), and decoupling weight magnitude from direction Liu et al. (2024). However, applying this paradigm to diffusion models is uniquely challenging because the adapter must maintain perceptually accurate semantic edits across a multi-timestep denoising trajectory. EfficientDM He et al. (2024) globally scales low-rank updates to help them cross quantization thresholds, but this uniform amplification indiscriminately magnifies both meaningful semantic shifts and optimization noise. IntLoRA Guo et al. (2025) resolves floating-point arithmetic mismatch via integer adapters and multiplicative merging, yet introduces a strongly anisotropic efficacy field that makes the merged outcome highly unpredictable across different weight coordinates.

# 3 PRELIMINARY

We formalize three components of quantized diffusion adaptation: diffusion denoising, low-rank adaptation, and integer deployment.

**Diffusion Denoising** Let $x_0$ be a clean latent sample, $y$ the text condition, and $t \in \{1, \dots, T\}$ the denoising timestep. Following denoising diffusion models, the forward process adds noise:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{1}$$

The denoiser $\epsilon_\theta(x_t, t, y)$ is trained with

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{x_0,\epsilon,t,y}\left[\|\epsilon - \epsilon_\theta(x_t, t, y)\|_2^2\right]. \tag{2}$$

During sampling, each step follows $x_{t-1} = \Phi_t(x_t, \epsilon_\theta(x_t, t, y))$ with sampler-specific update rule $\Phi_t$. A deployed adapter must therefore remain effective across multiple noise regimes, not only at a single static input distribution.

**Low-Rank Adaptation** For an adapted layer with frozen weight $W_\ell \in \mathbb{R}^{m_\ell \times n_\ell}$, LoRA introduces a trainable low-rank update: $W'_\ell = W_\ell + \lambda B_\ell A_\ell$, where $B_\ell \in \mathbb{R}^{m_\ell \times r}$ and $A_\ell \in \mathbb{R}^{r \times n_\ell}$ are optimized. In quantized diffusion, a continuous low-rank update is useful only if it changes the integer-deployed layer after merging.

**Quantized Integer Deployment** For affine quantization with scale $\delta_{W,\ell}$, zero-point $z_{W,\ell}$, and bit-width $b$, the integer code of the base weight is

$$K_{W,\ell} = \text{clip}\left(\left\lfloor \frac{W_\ell}{\delta_{W,\ell}} \right\rceil + z_{W,\ell}, 0, 2^b - 1\right). \tag{3}$$

Define the base integer anchor $a_\ell = K_{W,\ell} - z_{W,\ell}$, and $a_{\ell,\text{safe}}$ as $a_\ell$ with zeros replaced by ones. Under the AQS/MLA parameterization Guo et al. (2025), the continuous pre-merge adapter code is

$$u_\ell = o_\ell + \frac{R_\ell \odot \zeta_\ell + B_\ell A_\ell}{a_{\ell,\text{safe}}}, \tag{4}$$

where $R_\ell$ is the auxiliary matrix from adaptation-quantization separation, $\zeta_\ell$ is a bounded trainable rescaling tensor broadcastable to the shape of $R_\ell$, and $o_\ell$ is the fixed offset induced by the base quantizer. In the implementation, $o_\ell$ is the broadcasted base weight scale $\delta_{W,\ell}$. Since $o_\ell$ does not affect code displacements, it is irrelevant for transversality. The variable $u_\ell$ is therefore the continuous pre-merge code to be rounded before multiplicative integer merge, not the raw LoRA product. Let $\delta_{u,\ell}, z_{u,\ell}$ be the quantizer for the adapter code. The deployed layer is

$$W_{\text{dep},\ell} = \delta_{u,\ell}\left(\left\lfloor \frac{u_\ell}{\delta_{u,\ell}} + z_{u,\ell} \right\rceil - z_{u,\ell}\right) \odot a_\ell. \tag{5}$$

The central question is not whether $B_\ell A_\ell$ changes in a continuous space, but whether the change produces a discrete, merge-retained, denoising-relevant update in $W_{\text{dep},\ell}$.

# 4 METHOD

Reinterpreting quantized LoRA as cell-constrained movement, PolyLoRA addresses semantic facet sticking and asymmetric merging distortion via Forward-Aligned Crossing, Merge-Anisotropy Calibration, and Semantic Transversality Shaping.

## 4.1 QUANTIZATION-INDUCED CELL DECOMPOSITION

The deployment equation (Eq. (5)) is piecewise constant in $u_\ell$. For each integer adapter code $K_{u,\ell}$, its preimage defines a quantization cell:

$$\mathcal{P}_\ell(K_{u,\ell}) = \left\{ u_\ell : \frac{(u_\ell)_{ij}}{\delta_{u,\ell}} + z_{u,\ell} \in \left[(K_{u,\ell})_{ij} - \frac{1}{2}, (K_{u,\ell})_{ij} + \frac{1}{2}\right] \right\}. \tag{6}$$

These cells form a polyhedral partition of the adapter-code space. Any continuous update confined to a single cell leaves the rounded code unchanged. The deployed integer code changes only when the update crosses at least one cell facet.

**Proposition 1 (Deployment cell invariance).** Fix $\delta_{u,\ell}$, $z_{u,\ell}$, and $a_\ell$. If $u_\ell^{(0)}$ and $u_\ell^{(1)}$ lie in the same cell $\mathcal{P}_\ell(K_{u,\ell})$, then $W_{\mathrm{dep},\ell}(u_\ell^{(0)}) = W_{\mathrm{dep},\ell}(u_\ell^{(1)})$. If the rounded adapter code changes by $\Delta q_\ell$, the deployed weight changes by

$$\Delta W_{\mathrm{dep},\ell} = \delta_{u,\ell} \Delta q_\ell \odot a_\ell. \tag{7}$$

The proof (see Appendix A for details) follows by substituting the rounded codes before and after crossing. **Proposition 1** provides the geometric motivation for PolyLoRA: effective quantized adaptation requires traversing the facet boundaries of the adapter-code partition, and the base integer anchor modulates the contribution of each such traversal.

### 4.2 Deployment Failure under Cell-Constrained Adaptation

The cell decomposition (**Proposition 1**) reveals two deployment-specific failure modes.

**Semantic facet sticking.** Straight-through estimation propagates gradients through the rounding operation, even when the rounded code remains unchanged. Consequently, the optimizer may reduce the training objective by either moving the continuous parameter $u_\ell$ within its current partition cell $\mathcal{P}_\ell(K_{u,\ell})$, or approaching a facet boundary without crossing it. As illustrated in Fig. 1, such movement is observable in the continuous parameter space but leaves the deployed weight matrix $W_{\mathrm{dep},\ell}$ invariant. This explains why a quantized LoRA adapter may exhibit apparent training progress while yielding negligible improvement in deployed editability.

**Asymmetric merging distortion.** **Proposition 1** establishes that the deployed parameter update is scaled coordinate-wise by the base integer anchor $a_\ell$. Specifically, facet crossings along coordinates with zero or small-magnitude anchors contribute minimally after integer merging, whereas those along high-magnitude anchors are proportionally amplified. Consequently, an identical continuous displacement can induce vastly different deployed effects depending on the anchor magnitudes, and raw crossing count fails to capture this anisotropy. Even when a perturbation survives merging, it may reside in directions that exert negligible influence on the denoising prediction $\epsilon_\theta(x_t, t, y)$. This accounts for the observed discrepancy between apparent weight updates in $W_{\mathrm{dep},\ell}$ and the absence of measurable gains in generation quality.

### 4.3 PolyLoRA: Polyhedral Transversality-Driven Adaptation

Since **Proposition 1** establishes that the deployed parameter state changes exclusively through transverse crossings of quantization facet boundaries, PolyLoRA introduces a transversality regularization term $\mathcal{R}_{\mathrm{tr}}$ designed to enforce a three-stage criterion, mitigating both deployment failure modes. Specifically, (i) the update must induce a transverse crossing of a quantization facet; (ii) the crossing must yield a non-negligible contribution after integer merging; and (iii) the retained perturbation must align with score-space directions that meaningfully influence the denoising prediction $\epsilon_\theta(x_t, t, y)$.

**Forward-Aligned Crossing (FAC).** To enforce the first requirement that the update must induce a transverse crossing of a quantization facet boundary, FAC evaluates the normal displacement component with respect to the actual forward code $u_\ell$, rather than relying solely on the linear combination $B_\ell A_\ell$. Let $u_\ell^{\mathrm{sh}}$ denote an exponential moving average of the shadow code, and define the displacement $\Delta u_\ell = u_\ell - u_\ell^{\mathrm{sh}}$. The soft crossing score is then given by

$$\chi_\ell = \sigma\left(\frac{|\Delta u_\ell|/\delta_{u,\ell} - \theta_\chi}{\tau_\chi}\right), \tag{8}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\theta_\chi$ denotes the normalized crossing threshold, and $\tau_\chi > 0$ governs the smoothness of the soft threshold. By default, $\theta_\chi = 0.5$, corresponding to the half-cell threshold. This term incentivizes the optimization trajectory to exit the current partition cell, rather than remaining confined within it. Because $\chi_\ell$ is computed directly on $u_\ell$, the crossing criterion is forward-aligned. It acts on the same continuous parameter that is rounded during deployment, ensuring consistency between training-time regularization and inference-time quantization.

**Merge-Anisotropy Calibration (MAC).** To satisfy the second requirement that facet crossings yield non-negligible contributions after integer merging, MAC down-weights crossings along coordinates whose anchors confer minimal influence post-merge. In the strict setting, where $a_{\ell,m} = a_\ell$, we define the coordinate-wise geometric gain factor and the merge-aware crossing score

$$g_\ell = \mathrm{clip}\left(\frac{|a_{\ell,m}|}{\mathrm{mean}(|a_{\ell,m}|) + \varepsilon}, 0, g_{\max}\right), \qquad C_{\ell,\mathrm{merge}} = \frac{\sum_{i,j}(g_\ell)_{ij}(\chi_\ell)_{ij}}{\sum_{i,j}(g_\ell)_{ij} + \varepsilon}. \tag{9}$$

The strict formulation aligns precisely with the deployment quantization equation. For ablation studies, a relaxed variant may substitute $a_{\ell,\mathrm{safe}}$, which is a modified anchor vector that zeroes out entries corresponding to zero-anchor coordinates, to isolate the impact of anchor sparsity. The displacement-based instantiation of $\mathcal{R}_{\mathrm{tr}}$ minimizes the module-weighted penalty $\sum_\ell \pi_\ell \, (1 - C_{\ell,\mathrm{merge}})$, where $\pi_\ell$ denotes normalized importance weights across adapted modules. This criterion effectively transforms raw facet crossings into merge-visible crossings, *i.e.*, those whose geometric scaling ensures measurable impact in the deployed integer-weight space.

**Semantic Transversality Shaping (STS).** To satisfy the third requirement that the retained perturbation aligns with meaningful score-space directions, STS establishes a principled link between polyhedral crossing geometry and diffusion semantics. A facet crossing that survives integer merging may still yield a functionally weak perturbation if the resulting displacement is nearly orthogonal to score-sensitive directions. Directly optimizing the full denoiser Jacobian with respect to $W_{\mathrm{dep},\ell}$ is computationally prohibitive. PolyLoRA thus constructs a lightweight, activation-derived surrogate inspired by the Jacobian-Gram structure. Let $H_\ell \in \mathbb{R}^{d_p \times n_\ell}$ denote a normalized basis extracted from the activation matrix of the adapted module. Taking the top $d_p$ right singular vectors computed over a mini-batch activation buffer, the merge-weighted directional response is then given by

$$P_\ell = \left(g_\ell \odot \frac{\Delta u_\ell}{0.5\,\delta_{u,\ell} + \varepsilon}\right) H_\ell^\top. \tag{10}$$

PolyLoRA derives a semantic activation vector $v_\ell \in \mathbb{R}^{d_p}$ from $P_\ell$ by applying a soft threshold to the output-averaged energy of each projected direction:

$$(v_\ell)_k = \sigma\left(\frac{\sqrt{\frac{1}{m_\ell}\sum_{i=1}^{m_\ell}(P_\ell)_{ik}^2 + \varepsilon} - 1}{\tau_\chi}\right). \tag{11}$$

Stacking $v_\ell$ across the $L$ adapted modules yields $V \in \mathbb{R}^{L \times d_p}$. With normalized module weights $\pi$, PolyLoRA forms the module-response Gram matrix

$$G = \frac{1}{d_p}\left(\mathrm{diag}(\sqrt{\pi})\,V\right)\left(\mathrm{diag}(\sqrt{\pi})\,V\right)^\top, \tag{12}$$

where $G \in \mathbb{R}^{L \times L}$ quantifies whether adapted modules generate non-redundant semantic response patterns within the activation-derived subspace. $G$ serves as a measure of cross-module semantic diversity, not simply a covariance matrix over the $d_p$ basis directions.

**Summary.** Combining the above consideration, the transversality penalty is defined as

$$\mathcal{R}_{\mathrm{tr}} = -\log\det(I + \eta G) + \lambda_{\mathrm{bal}}\,\mathrm{Var}(\mathrm{diag}(G)) + \lambda_{\mathrm{scale}}\sum_\ell \pi_\ell\,\mathrm{mean}\big((\zeta_\ell - 1)^2\big), \tag{13}$$

where $\eta > 0$ governs the Gram reward strength; $\lambda_{\mathrm{bal}}$ and $\lambda_{\mathrm{scale}}$ control response balance across modules and scale regularization of the crossing modulation, respectively. The log-determinant term incentivizes low-rank but diverse semantic responses across modules; the diagonal variance term discourages concentration of response activity in only a few modules; and the scale penalty prevents $\zeta_\ell$ from becoming the sole degree of freedom for satisfying the crossing criterion. Since $G$ has rank at most $d_p$, the stabilized form $\log\det(I + \eta G)$ operates exclusively on the nonzero response modes. This merge-weighted semantic transversality criterion thus tightly couples the polyhedral deployment geometry to the functional semantics of the diffusion denoiser.

Given $\mathcal{R}_{\mathrm{tr}}$, PolyLoRA couples it to the base training objective via a multiplicative dual-barrier formulation. Let $\mathcal{L}_{\mathrm{base}}$ denote the denoising loss augmented with optional prior-preservation or

Table 1: Results on subject-driven generation averaged over 75 cases.

| Method | Bit-width | DINO ↑ | CLIP-I ↑ | CLIP-T ↑ | LPIPS ↓ |
|---|---|---|---|---|---|
| QLoRA Dettmers et al. (2023) | W8A8 | 0.3026 | 0.6556 | 0.2726 | 0.8027 |
| IR-QLoRA Qin et al. (2024) | W8A8 | 0.3144 | 0.6481 | 0.2736 | 0.8139 |
| QA-LoRA Xu et al. (2024) | W8A8 | 0.3157 | 0.6388 | 0.2784 | 0.8137 |
| EfficientDM He et al. (2024) | W8A8 | 0.4572 | 0.7023 | 0.2762 | 0.7962 |
| QuEST Wang et al. (2025) | W8A8 | 0.4949 | 0.7145 | 0.2796 | 0.7935 |
| IntLoRA Guo et al. (2025) | W8A8 | 0.5321 | 0.7329 | 0.2811 | 0.7952 |
| **PolyLoRA (Ours)** | W8A8 | **0.5467** | **0.7385** | **0.2864** | **0.7865** |
| QLoRA Dettmers et al. (2023) | W4A8 | 0.1749 | 0.5447 | 0.2624 | 0.8274 |
| IR-QLoRA Qin et al. (2024) | W4A8 | 0.1701 | 0.6057 | 0.2619 | 0.8308 |
| QA-LoRA Xu et al. (2024) | W4A8 | 0.1857 | 0.5938 | 0.2643 | 0.8306 |
| EfficientDM He et al. (2024) | W4A8 | 0.4011 | 0.6782 | 0.2793 | 0.8249 |
| QuEST Wang et al. (2025) | W4A8 | 0.4639 | 0.6976 | 0.2779 | 0.8264 |
| IntLoRA Guo et al. (2025) | W4A8 | 0.4596 | 0.6952 | 0.2785 | 0.8175 |
| **PolyLoRA (Ours)** | W4A8 | **0.4758** | **0.7035** | **0.2801** | **0.8166** |

consistency terms. The scalar transversality quality metric used by the barrier is

$$q_{\mathrm{tr}} = \sum_{\ell=1}^{L} \pi_\ell \left( \frac{1}{d_p} \sum_{k=1}^{d_p} (v_\ell)_k \right), \tag{14}$$

The module-weighted average of semantic crossing scores, which serves as a compact, differentiable proxy for deployed editability. The training objective is $\mathcal{L} = \mathcal{L}_{\mathrm{base}} (1 + m_{\mathrm{tr}}) (1 + b_{\mathrm{tr}})$, where

$$m_{\mathrm{tr}} = \rho_{\mathrm{m}} \frac{\lambda_{\mathrm{tr}} \omega(t,y) \mathcal{R}_{\mathrm{tr}}}{\mathrm{sg}(|\mathcal{L}_{\mathrm{base}}|) + \varepsilon}, \quad b_{\mathrm{tr}} = \kappa_{\mathrm{b}} \mu_{\mathrm{tr}} \omega(t,y) \,\mathrm{softplus}\left( \frac{q^\star - q_{\mathrm{tr}}}{\tau_{\mathrm{b}}} \right). \tag{15}$$

Here, $m_{\mathrm{tr}}$ dynamically matches the transversality penalty magnitude to the scale of $\mathcal{L}_{\mathrm{base}}$; $b_{\mathrm{tr}}$ implements a smooth, adaptive barrier activated when $q_{\mathrm{tr}}$ falls below the target threshold $q^\star$; and $\mu_{\mathrm{tr}}$ is updated via dual ascent to enforce the constraint asymptotically. The hyperparameters $\rho_{\mathrm{m}}$, $\kappa_{\mathrm{b}}$, and $\tau_{\mathrm{b}}$ govern matching fidelity, barrier aggressiveness, and transition sharpness, respectively. An additive coupling $\mathcal{L}_{\mathrm{base}} + \lambda_{\mathrm{tr}} \omega(t,y) \mathcal{R}_{\mathrm{tr}}$ is retained solely as an ablation baseline.

Overall, PolyLoRA reshapes quantized diffusion adaptation toward updates that are discrete, merge-retained, and semantically aligned.

# 5 Experiments

## 5.1 Datasets and Experimental Setup

**Datasets and Tasks** We evaluate PolyLoRA across two distinct personalized generation paradigms. For subject-driven generation, we sample a subset of 15 text-subject pairs from the DreamBooth dataset, testing each against five semantic prompts to yield 75 unique evaluation cases. For style-customized generation, we use a 12-style StyleDrop evaluation subset with six corresponding text prompts per style; the quantitative style results are reported in Table 2.

**Implementation Details** All experiments use Stable Diffusion v1.5 Rombach et al. (2022) on a single NVIDIA RTX 3090 GPU with uniform affine quantization Nagel et al. (2021) under W8A8 and W4A8 settings. PolyLoRA adapts attention projection layers through the integer multiplicative LoRA path of IntLoRA, while FP16 adapter baselines are quantized with PTQ before low-bit inference; further training details and hyperparameters are provided in Appendix C.

## 5.2 Main Results

**Subject-driven generation.** Table 1 details the quantitative performance of PolyLoRA against quantized adaptation baselines. Across DINO, CLIP-I, CLIP-T, and LPIPS, PolyLoRA achieves the




Figure 2: Qualitative comparison on subject-driven generation tasks.

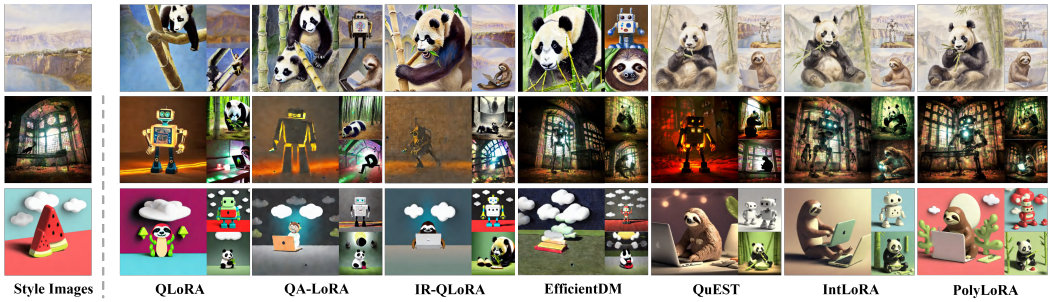



Figure 3: Qualitative comparison of style-customized generation. The text prompts are, respectively, "A friendly robot in [V] style," "A panda eating bamboo in [V] style," and "The letter 'G' in [V] style." More results are provided in Appendix G.

strongest overall performance. Under W8A8, it reaches a DINO score of 0.5467, CLIP-I of 0.7385, CLIP-T of 0.2864, and LPIPS of 0.7865.

The advantage becomes more pronounced under the aggressive W4A8 regime. PTQ-based baselines such as QLoRA and quantization-aware methods such as QA-LoRA exhibit substantial metric degradation, indicating the difficulty of preserving continuously optimized updates after highly discretized deployment. In contrast, PolyLoRA maintains robust editability and outperforms the structurally analogous IntLoRA by encouraging updates that cross quantization boundaries and remain effective after integer merging.

Additional multi-seed robustness results over 30 subjects are reported in Appendix F.1. PolyLoRA remains consistently better than IntLoRA across the evaluated training seeds.

**Style-customized generation.** PolyLoRA achieves a balance between style-image influence and text content, whereas several baselines struggle with semantic alignment and visual quality, as shown in Fig. 3. For example, QLoRA and IR-QLoRA in the first row tend to copy the reference style image when prompted with "A friendly robot in [V] style." They also introduce visual artifacts and omit essential semantic features, such as mechanical components of the robot.

Table 2: Style customization performance averaged over 12 styles. Higher is better.

| Method | CSD ↑ | Content CLIP-T ↑ |
|---|---|---|
| PolyLoRA (Ours) | **0.602** | **0.2463** |
| IntLoRA | 0.566 | 0.2231 |
| QuEST | 0.509 | 0.2107 |

For quantitative evaluation, we use a 12-style subset and measure style similarity with CSD and text-content alignment with Content CLIP-T. Higher values indicate better performance. As shown in Table 2, PolyLoRA achieves the highest CSD score of

Table 3: Efficiency comparison.

| Method | Step Time (s) | Memory (MB) |
|---|---|---|
| QLoRA Dettmers et al. (2023) | 0.635 | 12191.85 |
| EfficientDM He et al. (2024) | 0.458 | 14334.43 |
| QuEST Wang et al. (2025) | 0.680 | 15748.88 |
| IntLoRA Guo et al. (2025) | 0.628 | 8479.47 |
| **Ours** | 0.631 | 8475.97 |

Table 4: Ablation study averaged over 15 subjects under W4A8 quantization.

| Variant | DINO↑ | CLIP-I↑ | CLIP-T↑ | LPIPS↓ |
|---|---|---|---|---|
| No-Poly Control | 0.5360 | 0.7340 | **0.2602** | 0.8030 |
| w/o FAC | 0.5408 | 0.7395 | 0.2115 | 0.7992 |
| w/o MAC | 0.5636 | 0.7442 | 0.2340 | 0.7921 |
| w/o STS | 0.5650 | 0.7402 | 0.2496 | 0.7980 |
| Full PolyLoRA | **0.6032** | **0.7814** | 0.2583 | **0.7745** |




Figure 4: Fine-grained semantic sensitivity under a matched-seed clock prompt sequence.

0.602 and the highest Content CLIP-T score of 0.2463, outperforming both IntLoRA and QuEST. These quantitative results complement the qualitative comparisons in Fig. 3.

### 5.3 Efficiency Analysis & Fine-grained semantic effectiveness

Quantized adaptation prioritizes resource efficiency. As shown in Table 3, PolyLoRA achieves the lowest peak memory footprint (8475.97 MB) among all methods, below QuEST (15,000+ MB) and EfficientDM (14,000+ MB), which load FP16 weights during fine-tuning. PolyLoRA matches IntLoRA's step time (0.631 s) while reducing peak memory slightly, confirming that our dual-barrier transversality objective guides optimization effectively, without dynamic architectural changes.

To assess whether low-bit adapters preserve fine-grained semantic sensitivity, we use matched-seed prompt sequences where only one local attribute varies; all else, including subject, background, layout, and seed, remains fixed. In the clock experiment, ring thickness changes across four levels: none, thin, medium, thick. As shown in Fig. 4, PolyLoRA preserves clock face structure while accurately reflecting each ring-thickness level. Baseline results (EfficientDM, IntLoRA, QLoRA) in the appendix show less distinguishable ring changes under the same prompts. In a sneaker task varying black side-stripe width, PolyLoRA again more faithfully preserves subject geometry and captures the intended local semantic change.

### 5.4 Ablation Study

To isolate the impact of each core component, we conduct an ablation study on fifteen DreamBooth subjects under W4A8 (Table 4). Disabling any component degrades performance. Omitting FAC reintroduces semantic facet sticking: updates fail to cross quantization boundaries, dropping DINO from 0.6032 to 0.5408. Removing MAC causes asymmetric merging distortion, preventing boundary-crossing updates from surviving integer merging. Without STS, updates may survive quantization but misalign with denoising-sensitive directions, reducing CLIP-I from 0.7814 to 0.7402. Thus, the full PolyLoRA formulation is necessary to keep updates discrete, merge-retained, and semantically relevant. Intermediate diagnostics for FAC, MAC, and STS are in Appendix D and Tables 9–10.

## 6 Conclusion

We presented PolyLoRA, a polyhedral transversality-driven framework designed to bridge continuous low-rank adaptation and discrete integer deployment in diffusion models. By reshaping the optimization trajectory, PolyLoRA ensures parameter updates cross discrete quantization boundaries and survive integer merging while preserving denoising-sensitive semantics. Extensive evaluations under extreme low-bit regimes confirm that PolyLoRA consistently outperforms quantization-aware baselines without introducing inference overhead. While our current activation-derived subspace incurs minor training overhead and may incompletely capture subtle time-varying dynamics across the diffusion schedule, designing more lightweight regularizers presents a promising avenue for future exploration.

### AI USE STATEMENT

In preparing this manuscript, generative AI tools were used to assist with LaTeX formatting and language editing. No generative AI tool was used to generate experimental data or to replace the authors' verification of the methods, results, claims, and references. The authors reviewed all AI-assisted material and take responsibility for the final content of this work.

### REPRODUCIBILITY STATEMENT

The paper describes the model formulation, training objective, datasets, evaluation settings, and implementation details needed to reproduce the reported experiments. Additional proofs and qualitative results are provided in the appendix. Code and configuration files can be released in an anonymous supplementary archive during review.

### ETHICS STATEMENT

This work uses publicly available image-generation benchmarks and does not involve human participants or private personal data. We encourage responsible use of the resulting generative models and compliance with the licenses of all pretrained models and datasets.

## REFERENCES

Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-dit: Accurate post-training quantization for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28306–28315, June 2025.

Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6593–6602, June 2024.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=OUIFPHEgJU.

Yarden Frenkel, Yael Vinker, Ariel Shamir, and Daniel Cohen-Or. Implicit style-content separation using b-lora. In *European Conference on Computer Vision*, pp. 181–198. Springer, 2024.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NAQvF08TcyG.

Rohit Gandikota, Joanna Materzyńska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XL*, pp. 172–188, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-73660-5. doi: 10.1007/978-3-031-73661-2_10. URL https://doi.org/10.1007/978-3-031-73661-2_10.

Han Guo, Philip Greengard, Eric Xing, and Yoon Kim. LQ-loRA: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xw29VvOMmU.

Hang Guo, Yawei Li, Tao Dai, Shu-Tao Xia, and Luca Benini. IntloRA: Integral low-rank adaptation of quantized diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=K3QoqobsiR.

Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 13237–13249. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/2aab8a76c7e761b66eccaca0927787de-Paper-Conference.pdf.

Yefei He, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. EfficientDM: Efficient quantization-aware fine-tuning of low-bit diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=UmMa3UNDAz.

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

Yushi Huang, Ruihao Gong, Xianglong Liu, Jing Liu, Yuhang Li, Jiwen Lu, and Dacheng Tao. Temporal feature matters: A framework for diffusion model quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(10):8823–8837, 2025. doi: 10.1109/TPAMI.2025.3585692.

Jeonghoon Kim, Jung Hyun Lee, Sungdong Kim, Joonsuk Park, Kang Min Yoo, Se Jung Kwon, and Dongsoo Lee. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 36187–36207. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/7183f4fc87598f6c6e947b96714acbd6-Paper-Conference.pdf.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1931–1941, June 2023.

DONGXU LI, Junnan Li, and Steven Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 30146–30166. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/602e1a5de9c47df34cae39353a7f5bb1-Paper-Conference.pdf.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17535–17545, October 2023a.

Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. Loftq: LoRA-fine-tuning-aware quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LzPWWPAdY4.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22511–22521, June 2023b.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Xinyan Liu, Huihong Shi, Yang Xu, and Zhongfeng Wang. Taq-dit: Time-aware quantization for diffusion transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2026. doi: 10.1109/TCSVT.2026.3652275.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(5):4296–4304, Mar. 2024. doi: 10.1609/aaai.v38i5.28226. URL https://ojs.aaai.org/index.php/AAAI/article/view/28226.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

Haotong Qin, Xudong Ma, Xingyu Zheng, Xiaoyang Li, Yang Zhang, Shouda Liu, Jie Luo, Xianglong Liu, and Michele Magno. Accurate lora-finetuning quantization of llms via information retention. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 79320–79362. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/faacb7a4827b4d51e201666b93ab5fa7-Paper-Conference.pdf.

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

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22500–22510, June 2023.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6527–6536, June 2024.

Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora: Any subject in any style by effectively merging loras. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part I*, pp. 422–438, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-73231-7. doi: 10.1007/978-3-031-73232-4_24. URL https://doi.org/10.1007/978-3-031-73232-4_24.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1972–1981, June 2023.

Jie Shao, Hanxiao Zhang, and Jianxin Wu. Diffusion product quantization. *arXiv preprint arXiv:2411.12306*, 2024.

Kihyuk Sohn, Lu Jiang, Jarred Barber, Kimin Lee, Nataniel Ruiz, Dilip Krishnan, Huiwen Chang, Yuanzhen Li, Irfan Essa, Michael Rubinstein, Yuan Hao, Glenn Entis, Irina Blok, and Daniel Castro Chin. Styledrop: Text-to-image synthesis of any style. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KoaFh16uOc.

Haoxuan Wang, Yuzhang Shang, Zhihang Yuan, Junyi Wu, Junchi Yan, and Yan Yan. Quest: Low-bit diffusion model quantization via efficient selective finetuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15542–15551, October 2025.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, XIAOPENG ZHANG, and Qi Tian. QA-loRA: Quantization-aware low-rank adaptation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=WvFoJccpo8.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arxiv:2308.06721*, 2023.

SHIH-YING YEH, Yu-Guan Hsieh, Zhidong Gao, Bernard B W Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lyCORIS fine-tuning to model evaluation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wfzXa8e783.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824, 2023. doi: 10.1109/ICCV51070.2023.00355.

Maosen Zhao, Pengtao Chen, Chong Yu, Yan Wen, Xudong Tan, and Tao Chen. Pioneering 4-bit fp quantization for diffusion models: Mixup-sign quantization and timestep-aware fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18134–18143, June 2025.

Xingyu Zheng, Xianglong Liu, Haotong Qin, Xudong Ma, Mingyuan Zhang, Haojie Hao, Jiakai Wang, Zixiang Zhao, Jinyang Guo, and Michele Magno. BinaryDM: Accurate weight binarization for efficient diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YaeZwhXJ4k.

# A PROOF OF PROPOSITION 1

In this section, we provide the formal proof for Proposition 1 in Section 4.1 of the main paper.

**Proposition 1** (Deployment cell invariance)**.** *Fix $\delta_{u,\ell}$, $z_{u,\ell}$, and $a_\ell$. If $u_\ell^{(0)}$ and $u_\ell^{(1)}$ lie in the same cell $\mathcal{P}_\ell(K_{u,\ell})$, then $W_{\mathrm{dep},\ell}(u_\ell^{(0)}) = W_{\mathrm{dep},\ell}(u_\ell^{(1)})$. If the rounded adapter code changes by $\Delta q_\ell$, the deployed weight changes by*

$$\Delta W_{\mathrm{dep},\ell} = \delta_{u,\ell} \Delta q_\ell \odot a_\ell. \tag{16}$$

*Proof.* We prove the result for a fixed adapted layer $\ell$. Recall the deployment equation in Eq. (5):

$$W_{\mathrm{dep},\ell}(u) = \delta_{u,\ell} \left( \left\lfloor \frac{u}{\delta_{u,\ell}} + z_{u,\ell} \right\rceil - z_{u,\ell} \right) \odot a_\ell. \tag{17}$$

Define the rounded quantity and the centered rounded adapter code used in deployment as

$$\kappa_\ell(u) := \left\lfloor \frac{u}{\delta_{u,\ell}} + z_{u,\ell} \right\rceil, \qquad q_\ell(u) := \kappa_\ell(u) - z_{u,\ell}. \tag{18}$$

The deployment equation can then be written as

$$W_{\mathrm{dep},\ell}(u) = \delta_{u,\ell} q_\ell(u) \odot a_\ell. \tag{19}$$

We first consider movement inside a single cell. Suppose that $u_\ell^{(0)}$ and $u_\ell^{(1)}$ lie in the same cell $\mathcal{P}_\ell(K_{u,\ell})$. By the definition of $\mathcal{P}_\ell(K_{u,\ell})$ in the main text, this cell fixes the rounded adapter state under the same quantization convention used for deployment. Since $z_{u,\ell}$ is fixed, both continuous codes have the same rounded quantity in Eq. (5):

$$\kappa_\ell(u_\ell^{(0)}) = \kappa_\ell(u_\ell^{(1)}). \tag{20}$$

Because $z_{u,\ell}$, $\delta_{u,\ell}$, and $a_\ell$ are fixed, the centered rounded codes are also identical:

$$q_\ell(u_\ell^{(0)}) = q_\ell(u_\ell^{(1)}). \tag{21}$$

Substituting these rounded codes into the deployment equation gives

$$W_{\mathrm{dep},\ell}(u_\ell^{(0)}) = W_{\mathrm{dep},\ell}(u_\ell^{(1)}). \tag{22}$$

Therefore, continuous movement inside a single adapter-code cell is invisible after deployment.

We next consider movement across a cell boundary. Suppose that the continuous code crosses a boundary of the cell partition, so that the rounded quantity in Eq. (5) changes from

$$\kappa_\ell^{(0)} := \kappa_\ell(u_\ell^{(0)}) \quad \text{to} \quad \kappa_\ell^{(1)} := \kappa_\ell(u_\ell^{(1)}). \tag{23}$$

The corresponding centered code change is

$$\begin{aligned} \Delta q_\ell &:= q_\ell(u_\ell^{(1)}) - q_\ell(u_\ell^{(0)}) \\ &= \left(\kappa_\ell^{(1)} - z_{u,\ell}\right) - \left(\kappa_\ell^{(0)} - z_{u,\ell}\right) \\ &= \kappa_\ell^{(1)} - \kappa_\ell^{(0)}. \end{aligned} \tag{24}$$

By substituting the rounded codes before and after crossing into the deployment equation, we obtain

$$\begin{aligned} \Delta W_{\mathrm{dep},\ell} &:= W_{\mathrm{dep},\ell}(u_\ell^{(1)}) - W_{\mathrm{dep},\ell}(u_\ell^{(0)}) \\ &= \delta_{u,\ell} \left( q_\ell(u_\ell^{(1)}) - q_\ell(u_\ell^{(0)}) \right) \odot a_\ell \\ &= \delta_{u,\ell} \Delta q_\ell \odot a_\ell. \end{aligned} \tag{25}$$

Coordinate-wise, the deployed contribution of a crossing at entry $(i, j)$ is

$$(\Delta W_{\mathrm{dep},\ell})_{ij} = \delta_{u,\ell} (\Delta q_\ell)_{ij} (a_\ell)_{ij}. \tag{26}$$

If $(a_\ell)_{ij} = 0$, the crossing is erased after integer merge. If $|(a_\ell)_{ij}|$ is small, the crossing has a proportionally weak deployed effect. Although $a_{\ell,\mathrm{safe}}$ is used to form the continuous pre-merge code $u_\ell$, deployment still multiplies by the true anchor $a_\ell$. Therefore, zero-anchor coordinates remain invisible in $W_{\mathrm{dep},\ell}$.

This proves both deployment cell invariance and the anchor-modulated displacement formula. The first result formalizes semantic facet sticking, and the second result identifies the coordinate-wise source of asymmetric merging distortion. □

## B ALGORITHM OF POLYLORA

The pseudo code implementation details of the core module PolyLoRA are given below.

**Algorithm 1** PolyLoRA: Polyhedral Transversality-Driven Adaptation

**Require:** Quantized base model with integer anchors $a_\ell$, initialized LoRA adapters with shadow EMA variables $u_\ell^{\mathrm{sh}}$, and training data stream.

**while** training has not converged **do**
  Sample a training batch and execute the forward pass.
  **for** each adapted layer $\ell$ **do**
    Compute continuous update displacement: $\Delta u_\ell \leftarrow u_\ell - u_\ell^{\mathrm{sh}}$
    **Stage 1: Forward-Aligned Crossing**
    Evaluate soft crossing score: $\chi_\ell \leftarrow \sigma\left(\frac{|\Delta u_\ell|/\delta_{u,\ell} - 0.5}{\tau_\chi}\right)$
    **Stage 2: Merge-Anisotropy Calibration**
    Compute coordinate-wise gain: $g_\ell \leftarrow \mathrm{clip}\left(\frac{|a_\ell|}{\mathrm{mean}(|a_\ell|)+\varepsilon}, 0, g_{\max}\right)$
    **Stage 3: Semantic Transversality Shaping**
    Project onto activation basis: $P_\ell \leftarrow \left(g_\ell \odot \frac{\Delta u_\ell}{0.5\delta_{u,\ell}}\right) H_\ell^\top$
    Extract semantic vector: $(v_\ell)_k \leftarrow \sigma\left(\frac{\sqrt{\frac{1}{m_\ell}\sum_{i=1}^{m_\ell}(P_\ell)_{ik}^2 + \varepsilon} - 1}{\tau_\chi}\right)$
    Update activation basis $H_\ell$ via momentum on current batch.
  **end for**
  **Global Aggregation & Optimization**
  Stack $v_\ell$ to form matrix $V \in \mathbb{R}^{L \times d_p}$ and compute Gram matrix: $G \leftarrow \frac{1}{d_p}\left(\mathrm{diag}(\sqrt{\pi})V\right)\left(\mathrm{diag}(\sqrt{\pi})V\right)^\top$
  Compute transversality penalty: $\mathcal{R}_{\mathrm{tr}} \leftarrow -\log\det(I + \eta G) + \lambda_{\mathrm{bal}}\,\mathrm{Var}(\mathrm{diag}(G)) + \lambda_{\mathrm{scale}}\sum_\ell \pi_\ell\,\mathrm{mean}\left((\zeta_\ell - 1)^2\right)$
  Multiplicative dual-barrier coupling: $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{base}}\,(1 + m_{\mathrm{tr}})\,(1 + b_{\mathrm{tr}})$
  Update adapter parameters via backpropagation on $\mathcal{L}$ and update shadow variables via EMA.
**end while**

**Ensure:** Optimized, deployment-ready quantized LoRA adapters.

## C IMPLEMENTATION DETAILS

All experiments are conducted on a single NVIDIA RTX 3090 GPU using Stable Diffusion v1.5 Rombach et al. (2022) as the generative backbone. We apply uniform affine quantization Nagel et al. (2021), encompassing per-channel weight and per-tensor activation quantization under both W8A8 and W4A8 configurations. PolyLoRA adapts the attention projection layers using the integer multiplicative LoRA path established by IntLoRA. Conversely, FP16 adapter baselines undergo standard continuous training followed by Post-Training Quantization (PTQ) prior to low-bit inference. We employ straight-through estimation (STE) for rounding and optimizes the merge-aware dual-barrier transversality objective. Training for all tasks uses rank-4 adapters at a resolution of 512 and runs for 2000 steps with a learning rate of $6 \times 10^{-5}$. We use prior preservation with 200 class images. Unless otherwise specified, we set $\lambda_{\mathrm{bal}} = 0.05$, $\lambda_{\mathrm{scale}} = 0.005$, $\lambda_{\mathrm{tr}} = 0.025$, and $\theta_\chi = 0.5$.

## D MECHANISTIC VALIDATION OF FORWARD-ALIGNED CROSSING

To assess whether Forward-Aligned Crossing (FAC) produces the intended low-bit adaptation behavior, we compare the full PolyLoRA framework with the variant without FAC using two targeted diagnostics, `trans_hard_cross` and `trans_hard_merge_cross`. These diagnostics measure whether the learned update crosses the normalized crossing threshold $\theta_{\mathrm{cross}}$ before and after integer merging, respectively. They therefore directly evaluate the hard-boundary counterpart of the soft FAC crossing score and complement the standard image-level metrics reported in the component ablation.

Let $u_\ell$ denote the continuous pre-merge adapter code of layer $\ell$, namely the code rounded before integer deployment. Let $u_\ell^{\mathrm{sh}}$ denote the corresponding EMA shadow variable, which tracks the local training trajectory.

Following the notation in the main text, we define the forward-code displacement as

$$\Delta u_\ell = u_\ell - u_\ell^{\mathrm{sh}}. \tag{27}$$

Let $\delta_{u,\ell}$ be the quantization step size of the adapter code. For each coordinate $(i, j)$ in the adapted coordinate set $\Omega_\ell$, the normalized displacement is computed as

$$r_{\ell,ij} = \frac{|(\Delta u_\ell)_{ij}|}{\delta_{u,\ell} + \epsilon}. \tag{28}$$

We define a hard-crossing mask using the same half-cell threshold as the FAC crossing score in the main text:

$$h_{\ell,ij} = \mathbb{I}\left[r_{\ell,ij} \geq \frac{1}{2}\right]. \tag{29}$$

The first diagnostic, $\mathrm{trans_hard_cross}$, is the average hard-crossing rate over all adapted coordinates:

$$\mathrm{trans_hard_cross} = \frac{1}{|\Omega|} \sum_\ell \sum_{(i,j)\in\Omega_\ell} h_{\ell,ij}, \qquad \Omega = \{(\ell, i, j) : (i, j) \in \Omega_\ell\}. \tag{30}$$

This metric is the hard-threshold counterpart of the soft FAC crossing score. It measures how often the continuous forward-code trajectory crosses quantization-induced cell facets, rather than remaining confined within a quantization cell.

The second diagnostic, $\mathrm{trans_hard_merge_cross}$, further accounts for the anisotropic integer-merge geometry. Let $a_\ell$ denote the true base integer anchor used during deployment. The normalized merge gain is defined as

$$g_{\ell,ij} = \mathrm{clip}\left(\frac{|(a_\ell)_{ij}|}{\frac{1}{|\Omega_\ell|}\sum_{(p,q)\in\Omega_\ell} |(a_\ell)_{pq}| + \epsilon}, 0, g_{\max}\right). \tag{31}$$

The merge-visible hard-crossing diagnostic is computed as

$$\mathrm{trans_hard_merge_cross} = \frac{\sum_\ell \sum_{(i,j)\in\Omega_\ell} g_{\ell,ij} h_{\ell,ij}}{\sum_\ell \sum_{(i,j)\in\Omega_\ell} g_{\ell,ij} + \epsilon}. \tag{32}$$

Thus, $\mathrm{trans_hard_cross}$ measures whether the continuous forward-code update crosses adapter-code facets, whereas $\mathrm{trans_hard_merge_cross}$ measures whether these crossings occur on coordinates with non-negligible integer-merge leverage. Both diagnostics are averaged over the same training tail and subject set used in the ablation experiments.

Table 5 reports the two diagnostics. Full PolyLoRA increases $\mathrm{trans_hard_cross}$ from $1.84 \times 10^{-4}$ to $5.19 \times 10^{-4}$, yielding a $2.82\times$ relative gain. It also increases $\mathrm{trans_hard_merge_cross}$ from $2.37 \times 10^{-4}$ to $3.38 \times 10^{-4}$, yielding a $1.43\times$ relative gain. These results are consistent with the intended role of FAC: the optimization trajectory crosses quantization-induced cell facets more frequently, and a larger fraction of these crossings remains visible under the integer-merge geometry.

Together with the downstream improvements in the component ablation, these diagnostics support the interpretation that FAC contributes through the proposed low-bit crossing mechanism rather than acting only as an unstructured regularizer.

Table 5: Results on FAC mechanism diagnostics.

| Variant | `trans_hard_cross` ↑ | `trans_hard_merge_cross` ↑ |
|---|---|---|
| Without FAC (keep MAC+STS) | $1.84 \times 10^{-4}$ | $2.37 \times 10^{-4}$ |
| Full PolyLoRA | $\mathbf{5.19 \times 10^{-4}}$ | $\mathbf{3.38 \times 10^{-4}}$ |
| Relative gain | $\mathbf{2.82}\times$ | $\mathbf{1.43}\times$ |

## E Hyperparameter Sensitivity

PolyLoRA has two transversality-specific hyperparameters: the regularization weight $\lambda_{\mathrm{tr}}$ and the normalized crossing threshold $\theta_{\mathrm{cross}}$. The regularization weight controls the strength of the explicit transversality penalty, which steers the low-rank adaptation trajectory toward discrete and merge-retained updates under quantization. The crossing threshold sets the target displacement used by the constraint and determines how strictly updates

are required to cross quantization facet boundaries. Together, these parameters control the strength of the transversality objective and the required degree of boundary crossing.

Table 6 reports a one-factor-at-a-time sensitivity analysis under W4A8 quantization on five DreamBooth subjects, corresponding to 25 subjects. With $\theta_{\text{cross}}$ fixed at 0.5, varying $\lambda_{\text{tr}}$ from 0 to 0.05 changes the four evaluation metrics only modestly. This indicates that the method does not depend on precise tuning of the explicit transversality penalty. The $\lambda_{\text{tr}} = 0$ setting does not disable the full PolyLoRA mechanism: the PolyLoRA parameterization and the multiplicative dual-barrier constraint remain active, while only the explicit penalty term is removed. It is therefore a barrier-only configuration, and its competitive performance suggests that the adaptive barrier provides most of the effective constraint signal in this setting.

With $\lambda_{\text{tr}}$ fixed at 0.025, varying $\theta_{\text{cross}}$ from 0.3 to 0.7 also produces stable DINO, CLIP-I, CLIP-T, and LPIPS scores. Although the default pair is not the strongest aggregate setting in this sweep, all tested configurations train successfully without metric collapse. We therefore interpret these results as evidence of robustness to reasonable choices of the two transversality hyperparameters, rather than as evidence that the default values are optimal.

# F Robustness and Additional Experiments

## F.1 Multi-Seed Stability on 30 Subjects

To assess seed-level stability, we evaluate subject-driven generation on 30 subjects under W4A8 using three training seeds (42, 1024, and 2026). PolyLoRA and IntLoRA use identical training and evaluation settings. Table 7 reports the results for each seed together with the mean $\pm$ standard deviation, while Table 8 reports the paired differences and 95% confidence intervals. The intervals are computed from paired seed-level differences rather than by treating generated images as independent observations. This analysis therefore measures consistency across training seeds rather than population uncertainty across subjects.

Table 7: Multi-seed Performance (W4A8, 30 Subjects)

| Seed | Model | DINO (↑) | CLIP-I (↑) | CLIP-T (↑) | LPIPS (↓) |
|---|---|---|---|---|---|
| 42 | PolyLoRA (Ours) | 0.5402 | 0.7474 | 0.2671 | 0.8054 |
| 42 | IntLoRA | 0.5299 | 0.7375 | 0.2551 | 0.8153 |
| 1024 | PolyLoRA (Ours) | 0.5578 | 0.7423 | 0.2588 | 0.8012 |
| 1024 | IntLoRA | 0.5476 | 0.7305 | 0.2418 | 0.8157 |
| 2026 | PolyLoRA (Ours) | 0.5317 | 0.7382 | 0.2603 | 0.7901 |
| 2026 | IntLoRA | 0.5186 | 0.7253 | 0.2496 | 0.7995 |
| **Mean** | **PolyLoRA (Ours)** | **0.5432 $\pm$ 0.0133** | **0.7426 $\pm$ 0.0046** | **0.2621 $\pm$ 0.0044** | **0.7989 $\pm$ 0.0079** |
| **Mean** | **IntLoRA** | **0.5320 $\pm$ 0.0146** | **0.7311 $\pm$ 0.0061** | **0.2488 $\pm$ 0.0067** | **0.8102 $\pm$ 0.0092** |

Table 6: Hyperparameter sensitivity analysis of PolyLoRA under W4A8 quantization averaged over 20 evaluation cases.

| Loss Weight | Threshold | DINO ↑ | CLIP-I ↑ | CLIP-T ↑ | LPIPS ↓ |
|---|---|---|---|---|---|
| *(a) Varying Loss Weight (Threshold = 0.5)* | | | | | |
| 0.000 | 0.5 | 0.4915 | 0.7447 | 0.2599 | 0.8331 |
| 0.010 | 0.5 | 0.5021 | 0.7441 | 0.2611 | 0.8358 |
| 0.025 | 0.5 | 0.4832 | 0.7393 | 0.2511 | 0.8482 |
| 0.050 | 0.5 | 0.5132 | 0.7356 | 0.2536 | 0.8328 |
| *(b) Varying Threshold (Loss Weight = 0.025)* | | | | | |
| 0.025 | 0.3 | **0.5188** | **0.7586** | **0.2621** | **0.8299** |
| 0.025 | 0.5 | 0.4832 | 0.7393 | 0.2511 | 0.8482 |
| 0.025 | 0.7 | 0.5063 | 0.7464 | 0.2582 | 0.8411 |

Table 8: Paired Significance Analysis (PolyLoRA minus IntLoRA)

| Metric | Mean Paired Difference | 95% Confidence Interval (t-CI) |
|---|---|---|
| DINO (↑) | +0.0112 | [0.0071, 0.0153] |
| CLIP-I (↑) | +0.0115 | [0.0078, 0.0153] |
| CLIP-T (↑) | +0.0132 | [0.0050, 0.0215] |
| LPIPS (↓) | -0.0113 | [-0.0182, -0.0043] |

Across all three seeds, PolyLoRA improves all four metrics relative to IntLoRA: it obtains higher DINO, CLIP-I, and CLIP-T scores and lower LPIPS. The paired intervals in Table 8 exclude zero in the favorable direction for every metric, supporting consistent seed-level improvements.

### F.2 Internal Component Diagnostics

The image-level ablation in Table 4 is complemented by intermediate diagnostics for the three proposed mechanisms. FAC measures whether updates cross quantization-induced facets (Table 5), MAC measures whether these crossings occur on coordinates with useful integer-merge leverage (Table 9), and STS measures whether the retained updates align with denoising-sensitive directions (Table 10). Together, these diagnostics test the three deployment conditions targeted by PolyLoRA: crossing, merge retention, and semantic alignment.

To assess whether Forward-Aligned Crossing (FAC) produces the intended low-bit adaptation behavior, we compare the full PolyLoRA framework with the variant without FAC using two targeted diagnostics, `trans_hard_cross` and `trans_hard_merge_cross`. These metrics measure whether the learned update crosses the normalized threshold before and after integer merging, respectively. They therefore evaluate the hard-boundary counterpart of the soft FAC crossing score and complement the image-level ablation.

The full model increases `trans_hard_cross` from $1.84 \times 10^{-4}$ to $5.19 \times 10^{-4}$, yielding a 2.82× relative gain. It also increases `trans_hard_merge_cross` from $2.37 \times 10^{-4}$ to $3.38 \times 10^{-4}$, yielding a 1.43× relative gain. These results are consistent with the intended role of FAC: it drives the optimization trajectory across quantization-induced cell facets and increases the fraction of crossings that remain visible under the integer-merge geometry.

Table 9: MAC Intermediate Diagnostics

| MAC metric | IntLoRA | PolyLoRA w/o MAC | PolyLoRA |
|---|---|---|---|
| Merge-aware crossing score (↑) | 0.119 | 0.1175 | 0.1245 |
| Zero-anchor update-energy ratio (↓) | 0.297 | 0.3012 | 0.2897 |
| Deployed energy per flip (↑) | $2.82 \times 10^{-5}$ | $2.76 \times 10^{-5}$ | $2.91 \times 10^{-5}$ |

Table 10: STS Intermediate Diagnostics

| STS metric | IntLoRA | PolyLoRA w/o STS | PolyLoRA |
|---|---|---|---|
| Response log-determinant (↑) | 0.043 | 0.0418 | 0.0489 |
| Activation transport gain (↑) | 1.053 | 1.0519 | 1.0587 |
| Relative deployed gain (↑) | -0.019% | -0.033% | +0.021% |

The MAC diagnostics in Table 9 show that the full model has the highest merge-aware crossing score and deployed energy per flip, together with the lowest zero-anchor update-energy ratio. This pattern indicates that MAC directs updates toward coordinates with stronger integer-merge leverage. The STS diagnostics in Table 10 show higher response log-determinant and activation transport gain, while the relative deployed gain changes from negative without STS to positive in the full model. This change is consistent with STS reorienting merge-retained updates toward denoising-sensitive directions.

Taken together, the diagnostics support the intended roles of the three components. FAC increases facet crossings, MAC favors crossings that survive integer merging, and STS shapes the retained updates toward activation-derived response directions. These measurements complement the downstream image-level ablation and provide intermediate evidence for the proposed deployment-aware trajectory.

### F.3 Extended Hyperparameter Sensitivity

Table 11 extends the loss-weight and crossing-threshold sweep in Table 6 with one-factor-at-a-time sweeps over the activation-basis size $d_p$, learning rate, and LoRA rank. The experiments use W4A8 quantization and ten subjects, and report mean $\pm$ standard deviation across the evaluation cases.

The results show limited sensitivity to the activation-basis size over $d_p \in \{1, 2, 4, 8\}$. We use $d_p = 2$ as the default because it provides the best overall balance across fidelity and text alignment without enlarging the activation subspace unnecessarily. The tested learning rates also produce successful training, although the lower rate of $3 \times 10^{-5}$ yields weaker subject-fidelity and alignment scores. We therefore retain $6 \times 10^{-5}$ as the default operating point.

Changing the LoRA rank produces the expected capacity trade-off. Increasing the rank from $r = 4$ to $r = 8$ improves DINO and CLIP-I and reduces LPIPS, while slightly reducing CLIP-T and increasing the trainable parameter count and memory use. We use $r = 4$ for the main comparisons to keep the parameter count aligned with the baselines. All tested configurations train successfully without metric collapse, showing that the method does not depend on a single fragile hyperparameter setting.

Table 11: Extended Sensitivity Analysis (10 Subjects)

| Parameter | Setting | DINO (↑) | CLIP-I (↑) | CLIP-T (↑) | LPIPS (↓) |
|---|---|---|---|---|---|
| Basis Size ($d_p$) | $d_p$ = 2 (default) | $0.497 \pm 0.089$ | $0.741 \pm 0.041$ | $0.276 \pm 0.040$ | $0.814 \pm 0.084$ |
| Basis Size ($d_p$) | $d_p$ = 1 | $0.467 \pm 0.098$ | $0.737 \pm 0.031$ | $0.274 \pm 0.043$ | $0.814 \pm 0.078$ |
| Basis Size ($d_p$) | $d_p$ = 4 | $0.469 \pm 0.106$ | $0.732 \pm 0.049$ | $0.277 \pm 0.045$ | $0.813 \pm 0.081$ |
| Basis Size ($d_p$) | $d_p$ = 8 | $0.473 \pm 0.087$ | $0.729 \pm 0.051$ | $0.276 \pm 0.042$ | $0.816 \pm 0.081$ |
| Learning Rate | 6e-5 (default) | $0.497 \pm 0.089$ | $0.741 \pm 0.041$ | $0.276 \pm 0.040$ | $0.814 \pm 0.084$ |
| Learning Rate | 3e-5 | $0.417 \pm 0.092$ | $0.702 \pm 0.056$ | $0.273 \pm 0.038$ | $0.829 \pm 0.085$ |
| Learning Rate | 1e-4 | $0.490 \pm 0.058$ | $0.731 \pm 0.042$ | $0.264 \pm 0.031$ | $0.822 \pm 0.086$ |
| LoRA Rank | r = 4 (default) | $0.497 \pm 0.089$ | $0.741 \pm 0.041$ | $0.276 \pm 0.040$ | $0.814 \pm 0.084$ |
| LoRA Rank | r = 2 | $0.432 \pm 0.127$ | $0.727 \pm 0.051$ | $0.278 \pm 0.047$ | $0.839 \pm 0.102$ |
| LoRA Rank | r = 8 | $0.582 \pm 0.115$ | $0.789 \pm 0.040$ | $0.258 \pm 0.045$ | $0.805 \pm 0.099$ |

### F.4 Style Statistical Supplement

The main text reports the aggregate 12-style comparison in Table 2. This appendix provides the corresponding paired statistical analysis in Table 12. We evaluate style fidelity with CSD (Contrastive Style Descriptor) and text-content alignment with Content CLIP-T. The statistical unit is the matched style/prompt comparison, and the analysis uses paired differences between PolyLoRA and each baseline.

All 95% confidence intervals exclude zero in the favorable direction, and all Holm-adjusted $p$-values are below 0.05. These results support statistically significant improvements in both style similarity and text-aligned content preservation under the evaluated low-bit setting.

Table 12: Paired Significance Analysis for Style Customization

| Baseline | Metric | Mean $\Delta$ | 95% CI of $\Delta$ | Exact sign-test p | Holm-adjusted p |
|---|---|---|---|---|---|
| IntLoRA | CSD Style Similarity (↑) | +0.0360 | [+0.0118, +0.0602] | 0.0063 | 0.0127 |
| IntLoRA | Content CLIP-T (↑) | +0.0232 | [+0.0078, +0.0386] | 0.0063 | 0.0127 |
| QuEST | CSD Style Similarity (↑) | +0.0930 | [+0.0380, +0.1480] | 0.0005 | 0.0020 |
| QuEST | Content CLIP-T (↑) | +0.0356 | [+0.0136, +0.0576] | 0.0005 | 0.0020 |

### F.5 Cross-Backbone Generalization

We evaluate the same low-bit adaptation path beyond SD v1.5 using two complementary studies. Table 13 reports a five-subject SDXL feasibility study under W4A8. Tables 14–15 report a matched PolyLoRA–IntLoRA comparison on FLUX.1-dev over eight subjects and three training seeds. The complete FLUX per-subject results are given in Tables 16–18, and Table 19 summarizes pairwise wins across the 24 subject-seed pairs.

Table 13: Cross-Backbone Feasibility (W4A8, 5 Subjects)

| Backbone | Model | DINO (↑) | CLIP-I (↑) | CLIP-T (↑) | LPIPS (↓) |
|---|---|---|---|---|---|
| SD v1.5 | PolyLoRA (Ours) | 0.5932 | 0.7018 | 0.2699 | 0.8076 |
| SD v1.5 | IntLoRA | 0.5707 | 0.6905 | 0.2548 | 0.8262 |
| SDXL | PolyLoRA (Ours) | 0.6179 | 0.7253 | 0.2932 | 0.7864 |
| SDXL | IntLoRA | 0.6043 | 0.7122 | 0.2724 | 0.8075 |

The SDXL pilot shows that PolyLoRA outperforms IntLoRA on all four metrics under the same W4A8 setting. We treat this result as feasibility evidence because the study uses five subjects and a single seed.

For the stronger cross-backbone comparison, both methods use identical W4A8 quantization, rank 4, 2K training steps, a learning rate of $6 \times 10^{-5}$, 512 resolution, text prompts, and direct integer-merge deployment. The evaluation covers eight subjects, three training seeds (43, 1024, and 2026), and five fixed generation seeds per subject, yielding 24 subject-seed pairs and 120 evaluated images per method. The backbone and evaluation protocol are fixed across methods, so the comparison isolates the adaptation method.

Table 14: Matched FLUX.1-dev Results over Eight Subjects and Three Training Seeds

| Method | DINO (↑) [95% CI] | CLIP-I (↑) [95% CI] | CLIP-T (↑) [95% CI] | LPIPS-ref (↓) [95% CI] |
|---|---|---|---|---|
| PolyLoRA (Ours) | 0.386 [0.281, 0.491] | 0.695 [0.636, 0.754] | 0.256 [0.236, 0.276] | 0.768 [0.746, 0.789] |
| IntLoRA | 0.374 [0.269, 0.479] | 0.684 [0.624, 0.743] | 0.244 [0.227, 0.261] | 0.779 [0.758, 0.800] |
| Δ (PolyLoRA - IntLoRA) | **+0.012 [+0.008, +0.015]** | **+0.011 [+0.007, +0.015]** | **+0.012 [+0.007, +0.017]** | **-0.011 [-0.015, -0.007]** |

PolyLoRA improves DINO, CLIP-I, and CLIP-T by 0.012, 0.011, and 0.012, respectively, while reducing LPIPS-ref by 0.011. All four paired 95% confidence intervals exclude zero. The matched comparison therefore shows that the relative advantage persists on FLUX.1-dev under the same quantization and deployment protocol.

Table 15: Seed-wise Paired Improvements on FLUX.1-dev over Eight Subjects

| Training seed | Δ DINO (↑) | Δ CLIP-I (↑) | Δ CLIP-T (↑) | Δ LPIPS-ref (↓) |
|---|---|---|---|---|
| 43 | +0.015 | +0.014 | +0.016 | -0.018 |
| 1024 | +0.008 | +0.010 | +0.009 | -0.008 |
| 2026 | +0.011 | +0.010 | +0.012 | -0.008 |

The seed-wise differences remain positive for all three higher-is-better metrics and negative for LPIPS-ref. This consistency indicates that the aggregate FLUX improvement is not driven by a single training seed.

Tables 16–18 provide the complete per-subject results for the three training seeds. They expose the variation across subjects while preserving the same matched comparison used for the aggregate analysis.

Table 16: Per-subject FLUX.1-dev Results under Training Seed 43

| Subject | Method | DINO (↑) | CLIP-I (↑) | CLIP-T (↑) | LPIPS-ref (↓) |
|---|---|---|---|---|---|
| bear_plushie | PolyLoRA (Ours) | 0.541 ± 0.026 | 0.681 ± 0.025 | 0.288 ± 0.010 | 0.719 ± 0.065 |
| bear_plushie | IntLoRA | 0.528 ± 0.097 | 0.675 ± 0.032 | 0.265 ± 0.031 | 0.735 ± 0.066 |
| berry_bowl | PolyLoRA (Ours) | 0.159 ± 0.043 | 0.594 ± 0.028 | 0.224 ± 0.011 | 0.783 ± 0.028 |
| berry_bowl | IntLoRA | 0.144 ± 0.051 | 0.579 ± 0.064 | 0.208 ± 0.034 | 0.802 ± 0.029 |
| cat2 | PolyLoRA (Ours) | 0.323 ± 0.210 | 0.699 ± 0.101 | 0.241 ± 0.024 | 0.742 ± 0.102 |
| cat2 | IntLoRA | 0.296 ± 0.242 | 0.683 ± 0.117 | 0.225 ± 0.055 | 0.766 ± 0.054 |
| clock | PolyLoRA (Ours) | 0.319 ± 0.155 | 0.737 ± 0.128 | 0.263 ± 0.012 | 0.784 ± 0.013 |
| clock | IntLoRA | 0.291 ± 0.169 | 0.712 ± 0.131 | 0.242 ± 0.023 | 0.805 ± 0.020 |
| colorful_sneaker | PolyLoRA (Ours) | 0.244 ± 0.128 | 0.560 ± 0.095 | 0.236 ± 0.042 | 0.701 ± 0.017 |
| colorful_sneaker | IntLoRA | 0.234 ± 0.236 | 0.542 ± 0.101 | 0.228 ± 0.053 | 0.729 ± 0.020 |
| duck_toy | PolyLoRA (Ours) | 0.583 ± 0.064 | 0.734 ± 0.072 | 0.210 ± 0.006 | 0.781 ± 0.039 |
| duck_toy | IntLoRA | 0.581 ± 0.071 | 0.733 ± 0.095 | 0.201 ± 0.025 | 0.785 ± 0.038 |
| rc_car | PolyLoRA (Ours) | 0.515 ± 0.067 | 0.722 ± 0.047 | 0.288 ± 0.030 | 0.752 ± 0.023 |
| rc_car | IntLoRA | 0.509 ± 0.100 | 0.708 ± 0.034 | 0.262 ± 0.031 | 0.761 ± 0.035 |
| teapot | PolyLoRA (Ours) | 0.326 ± 0.100 | 0.791 ± 0.075 | 0.266 ± 0.049 | 0.770 ± 0.033 |
| teapot | IntLoRA | 0.306 ± 0.130 | 0.774 ± 0.074 | 0.258 ± 0.043 | 0.791 ± 0.038 |
| **8-subject mean** | **PolyLoRA (Ours)** | **0.376 [0.271, 0.482]** | **0.690 [0.636, 0.743]** | **0.252 [0.232, 0.272]** | **0.754 [0.732, 0.776]** |
| **8-subject mean** | **IntLoRA** | **0.361 [0.252, 0.470]** | **0.676 [0.622, 0.730]** | **0.236 [0.219, 0.253]** | **0.772 [0.752, 0.792]** |
| **8-subject mean** | **Δ (PolyLoRA - IntLoRA)** | **+0.015 [+0.009, +0.022]** | **+0.014 [+0.009, +0.019]** | **+0.016 [+0.011, +0.021]** | **-0.018 [-0.023, -0.012]** |

Table 17: Per-subject FLUX.1-dev Results under Training Seed 1024

| Subject | Method | DINO (↑) | CLIP-I (↑) | CLIP-T (↑) | LPIPS-ref (↓) |
|---|---|---|---|---|---|
| bear_plushie | PolyLoRA (Ours) | 0.569 ± 0.051 | 0.644 ± 0.009 | 0.292 ± 0.011 | 0.730 ± 0.047 |
| bear_plushie | IntLoRA | 0.558 ± 0.065 | 0.628 ± 0.031 | 0.287 ± 0.014 | 0.743 ± 0.050 |
| berry_bowl | PolyLoRA (Ours) | 0.221 ± 0.145 | 0.608 ± 0.059 | 0.193 ± 0.054 | 0.822 ± 0.040 |
| berry_bowl | IntLoRA | 0.234 ± 0.193 | 0.606 ± 0.069 | 0.209 ± 0.022 | 0.818 ± 0.039 |
| cat2 | PolyLoRA (Ours) | 0.388 ± 0.174 | 0.742 ± 0.069 | 0.281 ± 0.051 | 0.776 ± 0.032 |
| cat2 | IntLoRA | 0.372 ± 0.219 | 0.735 ± 0.085 | 0.260 ± 0.050 | 0.781 ± 0.036 |
| clock | PolyLoRA (Ours) | 0.317 ± 0.098 | 0.782 ± 0.021 | 0.287 ± 0.036 | 0.801 ± 0.019 |
| clock | IntLoRA | 0.309 ± 0.154 | 0.763 ± 0.055 | 0.253 ± 0.041 | 0.823 ± 0.023 |
| colorful_sneaker | PolyLoRA (Ours) | 0.245 ± 0.177 | 0.569 ± 0.085 | 0.266 ± 0.046 | 0.727 ± 0.011 |
| colorful_sneaker | IntLoRA | 0.226 ± 0.240 | 0.553 ± 0.092 | 0.258 ± 0.060 | 0.740 ± 0.023 |
| duck_toy | PolyLoRA (Ours) | 0.598 ± 0.033 | 0.761 ± 0.034 | 0.216 ± 0.026 | 0.791 ± 0.031 |
| duck_toy | IntLoRA | 0.597 ± 0.046 | 0.766 ± 0.028 | 0.217 ± 0.015 | 0.787 ± 0.033 |
| rc_car | PolyLoRA (Ours) | 0.523 ± 0.066 | 0.695 ± 0.028 | 0.267 ± 0.023 | 0.764 ± 0.029 |
| rc_car | IntLoRA | 0.508 ± 0.078 | 0.683 ± 0.032 | 0.256 ± 0.032 | 0.773 ± 0.034 |
| teapot | PolyLoRA (Ours) | 0.358 ± 0.064 | 0.795 ± 0.034 | 0.294 ± 0.023 | 0.789 ± 0.014 |
| teapot | IntLoRA | 0.348 ± 0.074 | 0.783 ± 0.052 | 0.287 ± 0.024 | 0.800 ± 0.036 |
| **8-subject mean** | **PolyLoRA (Ours)** | **0.402 [0.302, 0.503]** | **0.700 [0.641, 0.758]** | **0.262 [0.236, 0.288]** | **0.775 [0.752, 0.798]** |
| **8-subject mean** | **IntLoRA** | **0.394 [0.294, 0.494]** | **0.690 [0.630, 0.749]** | **0.253 [0.234, 0.273]** | **0.783 [0.762, 0.805]** |
| **8-subject mean** | **Δ (PolyLoRA - IntLoRA)** | **+0.008 [+0.001, +0.015]** | **+0.010 [+0.004, +0.015]** | **+0.009 [-0.002, +0.019]** | **-0.008 [-0.014, -0.002]** |

Table 18: Per-subject FLUX.1-dev Results under Training Seed 2026

| Subject | Method | DINO (↑) | CLIP-I (↑) | CLIP-T (↑) | LPIPS-ref (↓) |
|---|---|---|---|---|---|
| bear_plushie | PolyLoRA (Ours) | 0.521 ± 0.126 | 0.634 ± 0.069 | 0.273 ± 0.012 | 0.737 ± 0.047 |
| bear_plushie | IntLoRA | 0.504 ± 0.131 | 0.625 ± 0.068 | 0.258 ± 0.014 | 0.741 ± 0.053 |
| berry_bowl | PolyLoRA (Ours) | 0.117 ± 0.102 | 0.563 ± 0.074 | 0.204 ± 0.034 | 0.806 ± 0.038 |
| berry_bowl | IntLoRA | 0.103 ± 0.118 | 0.553 ± 0.085 | 0.193 ± 0.032 | 0.823 ± 0.051 |
| cat2 | PolyLoRA (Ours) | 0.417 ± 0.125 | 0.787 ± 0.041 | 0.292 ± 0.021 | 0.779 ± 0.038 |
| cat2 | IntLoRA | 0.399 ± 0.198 | 0.766 ± 0.059 | 0.273 ± 0.016 | 0.782 ± 0.044 |
| clock | PolyLoRA (Ours) | 0.345 ± 0.103 | 0.784 ± 0.096 | 0.270 ± 0.011 | 0.804 ± 0.013 |
| clock | IntLoRA | 0.342 ± 0.116 | 0.786 ± 0.104 | 0.264 ± 0.034 | 0.813 ± 0.027 |
| colorful_sneaker | PolyLoRA (Ours) | 0.210 ± 0.161 | 0.559 ± 0.049 | 0.246 ± 0.041 | 0.730 ± 0.018 |
| colorful_sneaker | IntLoRA | 0.204 ± 0.220 | 0.544 ± 0.094 | 0.229 ± 0.039 | 0.744 ± 0.022 |
| duck_toy | PolyLoRA (Ours) | 0.590 ± 0.026 | 0.777 ± 0.063 | 0.224 ± 0.009 | 0.788 ± 0.036 |
| duck_toy | IntLoRA | 0.587 ± 0.035 | 0.778 ± 0.047 | 0.221 ± 0.020 | 0.790 ± 0.040 |
| rc_car | PolyLoRA (Ours) | 0.496 ± 0.078 | 0.681 ± 0.066 | 0.260 ± 0.014 | 0.759 ± 0.036 |
| rc_car | IntLoRA | 0.480 ± 0.120 | 0.668 ± 0.092 | 0.244 ± 0.029 | 0.765 ± 0.037 |
| teapot | PolyLoRA (Ours) | 0.333 ± 0.086 | 0.774 ± 0.051 | 0.265 ± 0.037 | 0.789 ± 0.034 |
| teapot | IntLoRA | 0.321 ± 0.086 | 0.761 ± 0.074 | 0.258 ± 0.041 | 0.800 ± 0.045 |
| **8-subject mean** | **PolyLoRA (Ours)** | **0.379 [0.267, 0.490]** | **0.695 [0.626, 0.764]** | **0.254 [0.235, 0.274]** | **0.774 [0.754, 0.794]** |
| **8-subject mean** | **IntLoRA** | **0.368 [0.256, 0.479]** | **0.685 [0.615, 0.756]** | **0.243 [0.224, 0.261]** | **0.782 [0.761, 0.803]** |
| **8-subject mean** | **Δ (PolyLoRA - IntLoRA)** | **+0.011 [+0.007, +0.015]** | **+0.010 [+0.004, +0.016]** | **+0.012 [+0.008, +0.016]** | **-0.008 [-0.012, -0.004]** |

Table 19: Pairwise Win Counts on FLUX.1-dev

| Metric | PolyLoRA wins |
|---|---|
| DINO | 23 / 24 |
| CLIP-I | 21 / 24 |
| CLIP-T | 22 / 24 |
| LPIPS-ref | 22 / 24 |

Across the 24 subject-seed pairs, PolyLoRA outperforms IntLoRA on 23 pairs for DINO, 21 for CLIP-I, 22 for CLIP-T, and 22 for LPIPS-ref. These pairwise counts complement the aggregate confidence intervals and per-subject tables by showing how consistently the method improves across evaluation conditions.

### F.6 STS Implementation Details

STS is a training-time surrogate and does not modify the integer-merge inference path. For each adapted module, we collect the input activations from the current mini-batch, flatten the batch and token/spatial dimensions, and mean-center the resulting activation matrix. We initialize the activation basis $H_\ell$ with the top $d_p = 2$ right singular vectors, update it across batches with momentum 0.9, and renormalize it after each update.

The fixed module weights $\pi$ are normalized to sum to one and enter the response Gram matrix through $\mathrm{diag}(\sqrt{\pi})V$. All STS operations are performed during training. Thus, STS introduces neither an inference-time module nor an architectural change, and the deployment procedure remains the direct integer-merge path described in Algorithm 1.

## G MORE RESULTS

We present additional qualitative results in the following figures:

- Fig. 5 illustrates all reference styles used in our experiments, providing a visual overview of the diverse style images that guide the style-customized generation task.
- Fig. 6 shows full visual results for Fine-grained semantic effectiveness.
- Fig. 7 shows more subject-driven generation examples.
- Fig. 8 to 13 present more style-customized generation examples.

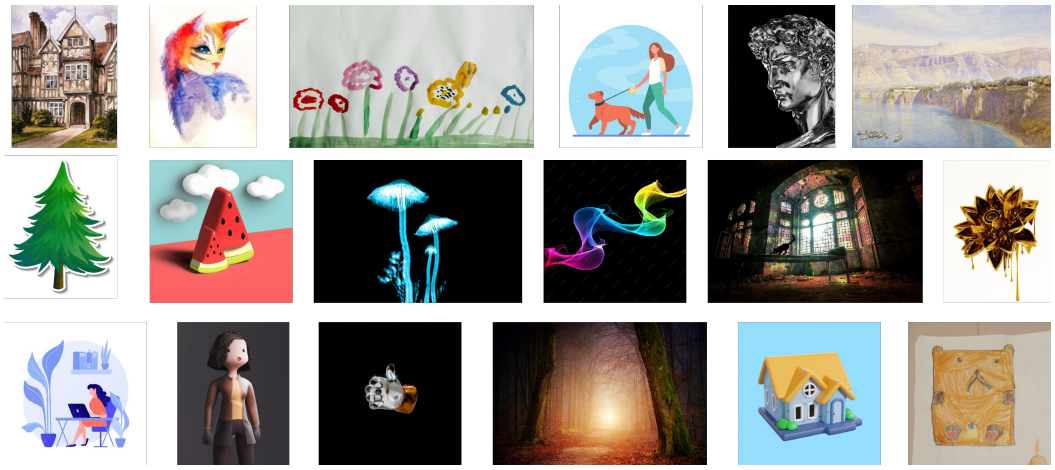


Figure 5: **Style images used for training in the style-customized generation task.** The corresponding text prompts are: “The letter ‘G’ in [V] style,” “A man riding a snowboard in [V] style,” “The Golden Gate Bridge in [V] style,” “A panda eating bamboo in [V] style,” “A fluffy baby sloth with a knitted hat trying to figure out a laptop in [V] style,” and “A friendly robot in [V] style.”

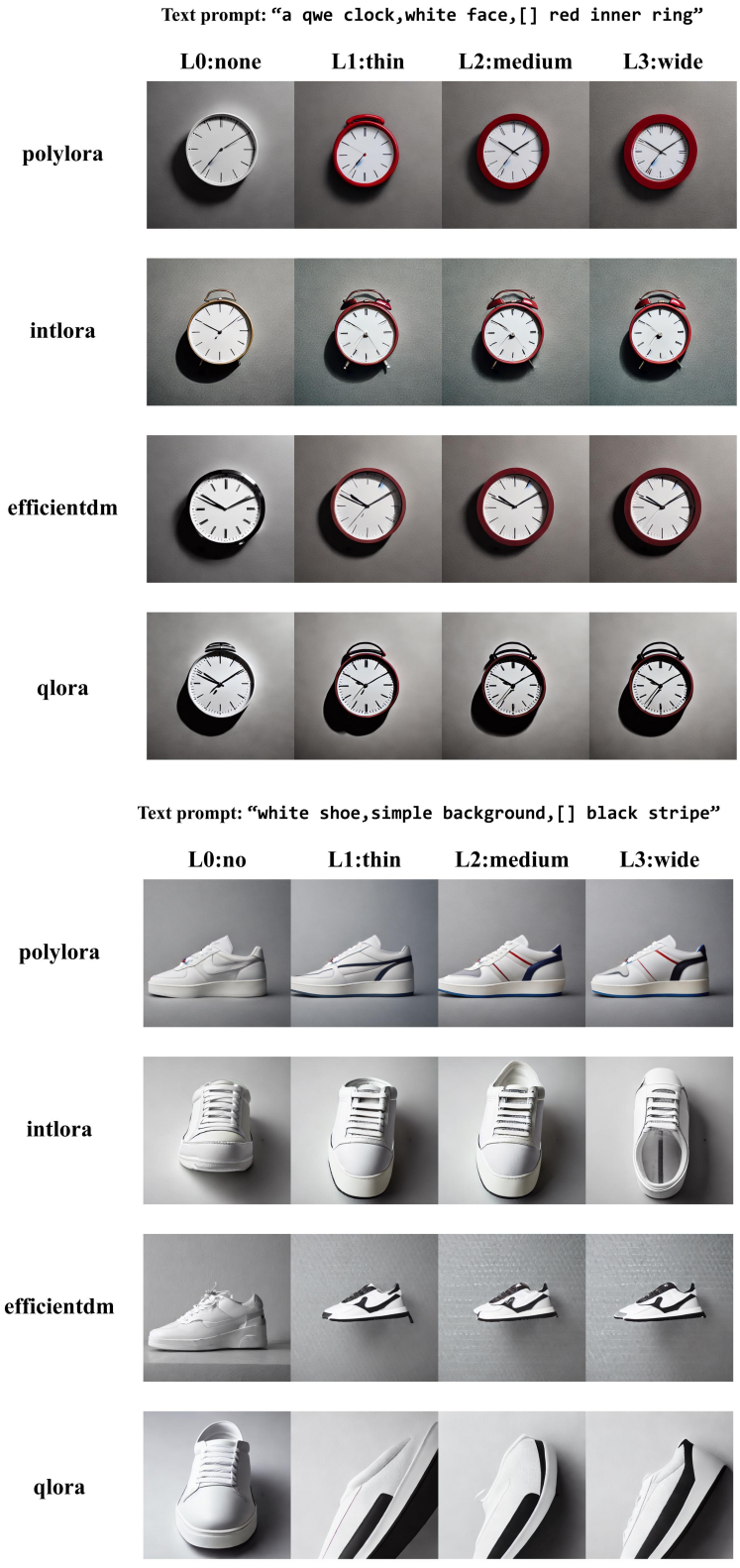



Figure 6: Full visual results for fine-grained semantic control: top – clock, bottom – shoe.




Figure 7: More qualitative comparison results on subject-driven generation.




Figure 8: More qualitative comparison results on style-customized generation.




Figure 9: More qualitative comparison results on style-customized generation.




Figure 10: More qualitative comparison results on style-customized generation.

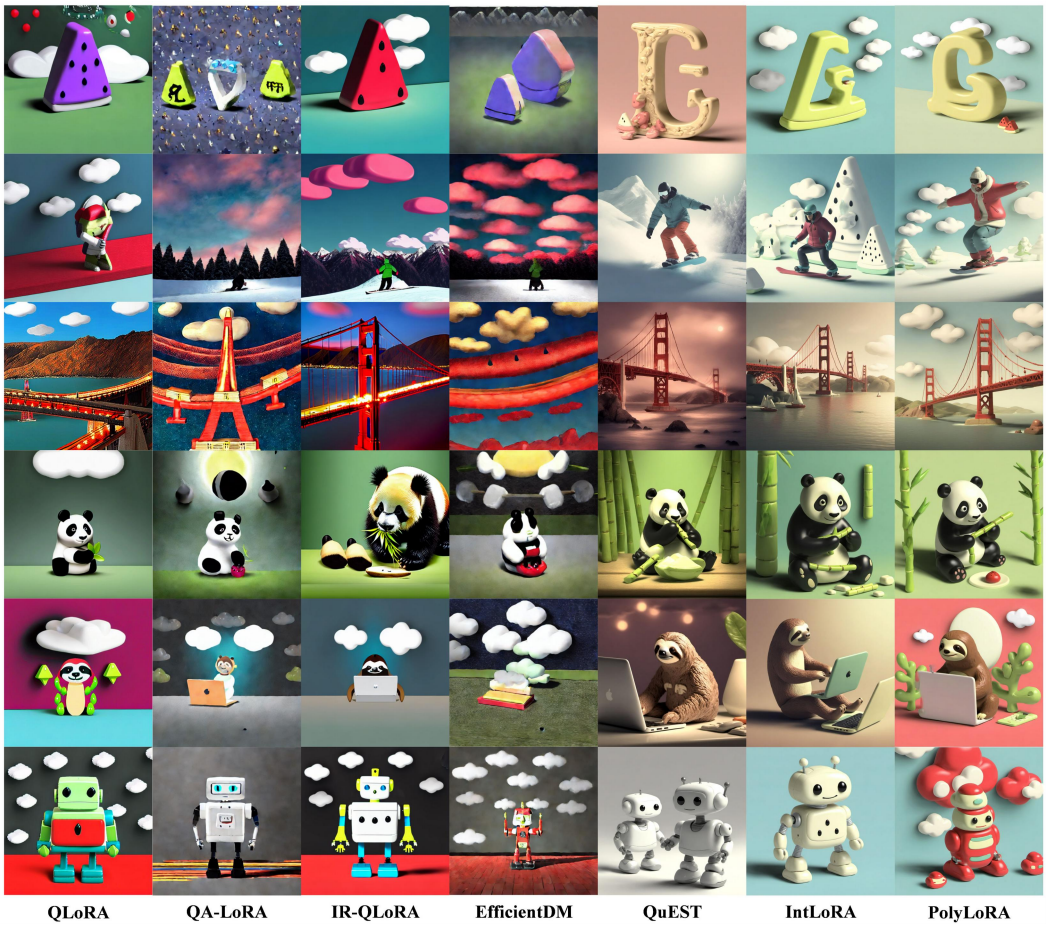



Figure 11: More qualitative comparison results on style-customized generation.




Figure 12: More qualitative comparison results on style-customized generation.




Figure 13: More qualitative comparison results on style-customized generation.