Decomposing Visual Histories with Vision-Language Agents: Hierarchical Temporal Guidance for Compositional Image Generation

Anonymous ACL submission

Abstract

Reference-conditioned image generation increasingly relies on visual histories that rarely speak with one voice: one example pins the subject, another fixes a layout, a third only suggests a palette. Current image-conditioned diffusion methods encode these references once, average them into a single vector, and inject it at every denoising step, so conflicting cues collide and the output is faithful to none. We propose the Hierarchical Temporal Guidance Framework (HTGF), a training-free pipeline that reframes multi-reference conditioning as a *temporal routing* problem. A semantic decomposer reads each reference and emits a soft routing distribution over three axes (Subject, Structure, Detail), and a closed-form SNR-sensitivity argument places each axis in the denoising window where it has the most leverage. A short manifold-aware corrector then smooths the trajectory at the stage boundaries where the active condition changes. HTGF adds no training and no architectural change to the diffusion backbone. On three datasets it outperforms strong VLM- and diffusion-based baselines, gaining $+11.7$ CIS over LaVIT on Movie Poster and FID 38.87 vs. 50.12 for the best baseline under noisy histories, and degrading gracefully down to the zero-reference limit.

1 Introduction

Modern image generation increasingly leans on visual references. A designer iterates against a mood board; a personalized model conditions on a user's interaction history; a multi-modal interface stitches together prompts of mixed text and images. In each of these settings the user supplies several reference images, and in each of them those references rarely speak with one voice: one example pins the subject, another suggests a layout, a third only fixes a palette, and a fourth may be incidental. The dominant practice in image-conditioned diffusion is to ignore this disagreement entirely. References are encoded once, the embeddings are summed or averaged, and the resulting static vector is injected at every denoising step (Ye et al., 2023; Jin et al., 2024; Xu et al., 2025a). The compromise is destructive: a layout cue from one reference fights a texture cue from another and the output is faithful to neither; a single stylistic outlier drags the entire generation off-prompt.

The natural diagnosis is that two kinds of structure are being thrown away at once. References are not interchangeable: some are most informative about *what* the image depicts, others about *where* things are arranged, others about *how* surfaces look. And denoising is not a stationary function but a trajectory that lays down coarse semantics under high noise, then layout, then surface detail. The harder problem, and the one this paper addresses, is that these two structures cannot be exploited in isolation. A vision-language model can decompose references into semantic axes, but decomposition without temporal placement still merges every axis at every step. A hierarchical sampler can stage the denoising schedule, but staging without decomposition has nothing to route. Each half on its own collapses back into flat fusion. The two structures are *coupled*, and the open question is what couples them: what tells you that *this* reference, scored on *this* axis, belongs in *this* denoising window. Without a principled answer, the schedule reduces to another tuned hyperparameter that has to be rediscovered on every dataset.

This paper develops a principled answer. We observe that an image-conditioning vector's influence on the predicted clean image follows in closed form from the denoiser's noise level: under the standard Tweedie formulation (§3.3, Proposition 1), high-noise steps have globally amplified leverage and low-noise steps have only local leverage. This is not a heuristic schedule; it is a property of the denoiser itself, and it locks each semantic axis to

a specific denoising window, placing Subject in the high-noise regime and Detail in the low-noise regime, with no tuned schedule in between. A semantic decomposer scores each reference along the three axes, a soft routing distribution then routes each reference, and the SNR-derived window supplies the placement. The remaining failure mode is mechanical: stage-aligned injection introduces discontinuities at the boundaries between windows, which is precisely the regime where any scheduling-based approach breaks. A short manifold-aware corrector, guided on the Tweedie-estimated clean image rather than the noisy latent, smooths those discontinuities. The resulting pipeline, HTGF, adds no training and no architectural change to the diffusion backbone.

What follows from taking this trajectory structure seriously goes beyond the headline numbers. The routing distribution is well-defined for any number of references, so the system degrades gracefully as the reference set shrinks, down to the zero-reference limit (§4.6). The schedule rests on a smooth sensitivity, so performance is insensitive to the exact phase boundaries across a wide window (§4.5). And because the decomposition is filtered by a confidence threshold before any reference is allowed to contribute, the framework stays stable on noisy reference sets where flat-fusion baselines collapse.

Contributions.

- We identify multi-reference diffusion conditioning as a *coupled* problem of semantic decomposition and temporal placement: decomposition without placement reverts to flat fusion, placement without decomposition has nothing to route. Prior work addresses at most one side.
- We show that the placement is not a free hyperparameter. A closed-form Tweedie-sensitivity argument (Proposition 1) locks each semantic axis to a specific denoising window, removing the schedule from the list of tuned knobs. A manifold-aware corrector absorbs the trajectory discontinuities that emerge from stage-aligned injection, the principal failure mode of any scheduling-based approach.
- HTGF, the training-free instantiation of this principle, achieves $+11.7$ CIS over LaVIT on Movie Poster, FID 38.87 vs. 50.12 for the best baseline under noisy histories, and graceful degradation in the cold-start limit.

2 Related Work

2.1 Multi-Reference and Personalized Image Generation

Personalized image generation has progressed from per-concept optimization (Gal et al., 2023; Ruiz et al., 2023; Kumari et al., 2023) to tuning-free encoder-based architectures that inject identity through decoupled cross-attention or stacked embeddings (Ye et al., 2023; Li et al., 2024; Shi et al., 2024), ID landmarks (Wang et al., 2024; Parihar et al., 2024), or layout guidance (Wang et al., 2025). Multi-subject extensions push the same recipe to several explicit references (He et al., 2024; Tewel et al., 2024; Chen et al., 2025; Lin et al., 2026a). MLLM-conditioned encoders such as EasyRef (Zong et al., 2025), OmniGen (Xiao et al., 2025), and LaVIT (Jin et al., 2024) accept arbitrary reference groups but treat them as a flat token sequence, while Pigeon (Xu et al., 2025a), DRC (Xu et al., 2025b), and PMG (Shen et al., 2024) filter or disentangle these signals before fusion. Across this line of work, the conditioning vector is constructed once and reused at every denoising step—precisely the assumption HTGF removes. We do not assume a clean reference set: the inputs are heterogeneous, individually noisy, and routinely disagree on subject, layout, or texture.

2.2 Language-Driven Control and Retrieval for Image Generation

A growing line of work places a language model in the loop of image generation. Instruct-Imagen (Hu et al., 2024) and unified multimodal LLMs (Koh et al., 2023; Jin et al., 2024) consume mixed text–image instructions; MLLM-guided editing (Fu et al., 2024) routes edit operations through a planner. Closer to our setting, TAME (Hong et al., 2026) uses a state-aware MLLM as a personalization assistant, while FuseAgent (Lin et al., 2026b), AgenticRetoucher (Shen et al., 2026), and PerTouch (Chang et al., 2026) use VLMs to perceive image-level cues and plan inference-time interventions. A parallel branch is retrieval-augmented image generation: FineRAG (Yuan et al., 2025) retrieves fine-grained image fragments at generation time, and ImageRAG (Shalev-Arkushin et al., 2025) retrieves reference images dynamically per prompt. These two strands share an emphasis on

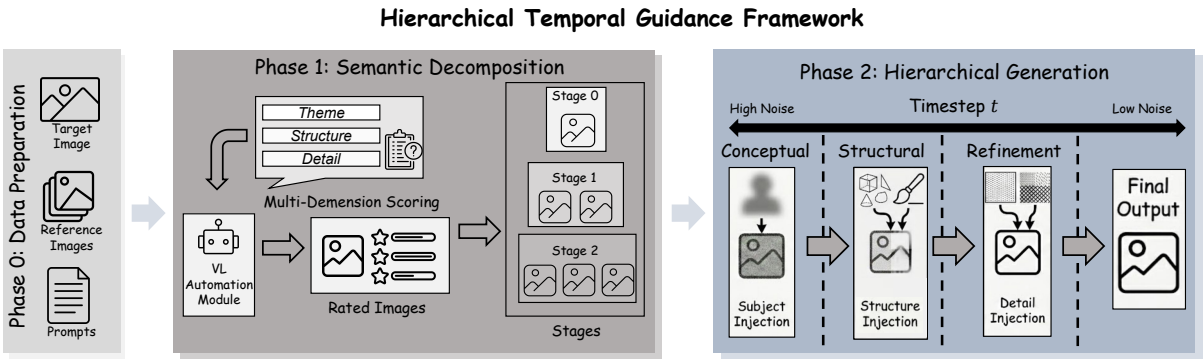

Figure 1: Overall architecture of HTGF. **Phase 0** prepares the target prompt and retrieved reference images. **Phase 1 (Semantic Decomposition)** uses a VL agent to score each reference along Subject, Structure, and Detail and to emit a (role, weight, stage) policy. **Phase 2 (Hierarchical Generation)** executes the denoising trajectory with stage-aligned injection: Conceptual features anchor global semantics at high noise, Structural guidance shapes layout at mid-noise, and Detail refinement enriches local texture at low noise. A manifold-aware corrector smooths the trajectory across stage transitions.

which signals to bring into generation—which images to retrieve, which operation to plan, which assistant turn to fire—and then hand the resulting tokens or actions to a standard generation path. HTGF operates on an orthogonal axis: given a candidate set of references, it decides *when* along the denoising trajectory each one should act, and is therefore composable with any of the above as front ends.

2.3 Diffusion-Side Mechanisms: Hierarchy, On-Manifold Sampling, and Preference Alignment

A different family of works modifies the diffusion process itself. Semantic-First Diffusion (Pan et al., 2025) asynchronously denoises semantic and texture latents to improve convergence; Hierarchical Koopman Diffusion (Bai et al., 2025a) separates coarse and fine spectral components in a lifted linear space; Hierarchical Diffusion LMs (Zhou et al., 2025) predict next-scale semantics; Attribute Diffusion (Parihar et al., 2025) performs coarse-to-fine attribute editing. None of these methods map heterogeneous *external* references onto the schedule; they reorganize the internal latents. On the trajectory-correction side, TAG (Park et al., 2025), DIAMOND (Polowczyk et al., 2026), and MCLC (Lee et al., 2026) show that guiding on the Tweedie-estimated clean image rather than the noisy latent yields semantically meaningful updates; we adopt this principle to absorb the discrete condition switches introduced by stage-aligned injection. Finally, Diffusion-DPO (Wallace et al., 2024), following the original DPO formulation (Rafailov et al., 2023), fine-tunes the backbone on preference pairs so that the model prefers certain outputs over others. This is *training-time* alignment at the level of which images a backbone produces; HTGF is *training-free* and inference-time, operating at the level of how a candidate reference set is decomposed and injected. The two are orthogonal—a DPO-tuned backbone could be slotted into HTGF's generation phase, and HTGF's per-reference filtering could feed an offline preference loop—and we keep the explicit empirical comparison for follow-up work (cf. Limitations).

3 Methodology

Figure 1 gives an overview. After a retrieval module supplies candidate references $\mathcal{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_n\}$ for a target prompt $\mathcal{P}$ (§3.1), a vision-language agent emits a per-reference control policy $\{(\boldsymbol{\pi}_j, w_j)\}$ (§3.2). A latent diffusion model then executes a denoising trajectory in which references are injected in stage-matched windows derived from a first-principle sensitivity argument (§3.3), and a Tweedie-based corrector smooths the trajectory at stage boundaries (§3.4).

3.1 Preliminaries

Problem. Given a textual prompt $\mathcal{P}$ and a retrieved reference set $\mathcal{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_n\}$ reflecting a user's prior visual interactions, we synthesize an image $\mathbf{x}_0$ that satisfies the prompt and is consistent with the references. Direct fusion of $\mathcal{R}$ is ill-posed because references may agree on subject while dis-

agreeing on layout or texture.

Latent diffusion. We build on Latent Diffusion Models (LDMs). An encoder $\mathcal{E}$ maps an image to a latent $\mathbf{z}_0 = \mathcal{E}(\mathbf{x})$; the forward process corrupts $\mathbf{z}_0$ as $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. A learned $\epsilon_\theta(\mathbf{z}_t, t, \mathbf{c})$ reverses this process under condition $\mathbf{c}$. We turn the condition itself into a function of t: $\mathbf{c}_t$ varies across the denoising trajectory to resolve conflicts in $\mathcal{R}$.

3.2 VLM as Agentic Decomposer

Rather than treat the VLM as a flat similarity scorer, we cast it as an *agent* that produces a structured control policy for the downstream generator. Given a reference $\mathbf{r}_j$ and a core target concept $\mathbf{r}_{core}$ (derived from the user's high-frequency interactions or session intent), the agent emits a per-axis score vector and an intensity:

$$(\mathbf{S}_j,\ w_j) \ = \ \mathcal{A}_{\text{VLM}}(\mathbf{r}_j,\ \mathbf{r}_{core},\ \mathcal{P}),$$

where $\mathbf{S}_j = [S_{sub}, S_{str}, S_{det}] \in [0,1]^3$ is the score along three semantic axes (Subject, Structure, Detail) and $w_j \in [0, w_{max}]$ controls overall influence. The routing distribution $\boldsymbol{\pi}_j$ over the three stages is derived from $\mathbf{S}_j$ below; the hard role K_j is its argmax.

Three-axis decomposition. The roles correspond to the inductive bias of image formation:

- **Subject (S_{sub}):** global atmosphere, color palette, artistic style.
- **Structure (S_{str}):** geometric layout, composition, spatial arrangement.
- **Detail (S_{det}):** high-frequency information, textures, local patterns.

The agent emits a score vector $\mathbf{S}_j = [S_{sub}, S_{str}, S_{det}]$. We derive two quantities from $\mathbf{S}_j$: a per-stage *routing distribution* $\boldsymbol{\pi}_j \in \Delta^2$ that decides *when* the reference contributes, and a scalar *intensity* w_j that decides *how strongly*:

$$\boldsymbol{\pi}_j \ = \ \text{softmax}(\beta\, \mathbf{S}_j), \qquad w_j \ = \ \phi(\max(\mathbf{S}_j)), \tag{1}$$

with the thresholded linear mapping

$$\phi(s) = \begin{cases} w_{min} + \frac{s-\tau}{1-\tau}(w_{max} - w_{min}), & s > \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

$\beta \in [0, \infty)$ is a routing temperature; τ is a confidence threshold; both are configuration choices, not learned. The threshold τ functions as a language-grounded *filter*: a reference whose maximum aspect score falls below τ has $w_j = 0$ and contributes nothing. We empirically verify in §4 that this filter is what gives HTGF its robustness to noisy histories: removing it (Table 3, row w/o (a)) collapses CIS from 40.6 to 28.2.

Argmax as a temperature limit. Hard role assignment $K_j = \text{argmax}(\mathbf{S}_j)$ is the $\beta \to \infty$ limit of Eq. 1; the main tables use this limit and the soft-routing panels (§4.4–§4.6) use $\beta = 4$.

3.3 Hierarchical Multi-Image Guidance (HMIG)

The VLM agent produces *what kind* of signal each reference carries (via $\boldsymbol{\pi}_j$). We now address *when* during denoising it is applied. We partition the schedule into three windows

- **Stage 0 (Conceptual, $t \in [T_1, T]$):** $s(t) = 0$
- **Stage 1 (Structural, $t \in [T_2, T_1]$):** $s(t) = 1$
- **Stage 2 (Refinement, $t \in [0, T_2]$):** $s(t) = 2$

At step t in stage $s(t)$, each reference r_j contributes with effective weight $w_j \cdot \pi_j[s(t)]$, giving the time-aware condition embedding

$$\mathbf{E}_t = \sum_{j=1}^{n} w_j{\cdot}\pi_j[s(t)]{\cdot}\text{Proj}(\text{Enc}(\mathbf{r}_j)) + (1{-}\lambda_t)\,\text{Enc}_{txt}(\mathcal{P}). \tag{3}$$

$\mathbf{E}_t$ is consumed by the U-Net via cross-attention. When $\beta \to \infty$, $\pi_j[s(t)]$ becomes the indicator $\mathbf{1}[K_j = s(t)]$ and Eq. 3 reduces to the hard schedule under which the main tables were measured.

First-principle justification. The stage assignment is not heuristic. We can characterize how strongly an image conditioning vector influences the *predicted clean image* $\hat{\mathbf{x}}_{0,t}$ at each step.

Proposition 1 (SNR-aligned sensitivity)**.** *Let $\hat{\mathbf{x}}_{0,t} = (\mathbf{z}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{z}_t, t, \text{cond}))/\sqrt{\bar{\alpha}_t}$ be the Tweedie estimate of the clean image at step t. Then*

$$\frac{\partial \hat{\mathbf{x}}_{0,t}}{\partial \mathbf{e}_{\text{img}}} \ \propto \ -\frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\ \frac{\partial \epsilon_\theta}{\partial \mathbf{e}_{\text{img}}}, \tag{4}$$

and the scaling factor $\sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t}$ is monotonically decreasing in $\text{SNR}(t) = \bar{\alpha}_t/(1-\bar{\alpha}_t)$.

Algorithm 1 HTGF: Decomposition + HMIG + Manifold Correction

Input: Prompt $\mathcal{P}$, refs $\mathcal{R} = \{\mathbf{r}_j\}_{j=1}^{n}$, UNet ϵ_θ, VL agent $\mathcal{A}_{\text{VLM}}$, boundaries T_1, T_2.
Output: Image $\mathbf{x}_0$.

$\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\hat{\mathbf{x}}_{anchor} \leftarrow \emptyset$
{Phase 1: Decompose}
for $\mathbf{r}_j \in \mathcal{R}$ **do**
  $\mathbf{S}_j \leftarrow \mathcal{A}_{\text{VLM}}(\mathbf{r}_j, \mathbf{r}_{core}, \mathcal{P})$
  $\boldsymbol{\pi}_j \leftarrow \text{softmax}(\beta\, \mathbf{S}_j)$; $w_j \leftarrow \phi(\max \mathbf{S}_j)$
end for
{Phase 2: Hierarchical generation}
for $t = T, \ldots, 1$ **do**
  $s \leftarrow s(t) \in \{0, 1, 2\}$ {stage window of t}
  $\mathbf{E}_t \leftarrow \text{Enc}_{txt}(\mathcal{P})$
  for $\mathbf{r}_j \in \mathcal{R}$ **do**
    $\mathbf{E}_t \mathrel{+}= w_j \cdot \pi_j[s] \cdot \text{Proj}(\text{Enc}(\mathbf{r}_j))$
  end for
  $\epsilon_t \leftarrow \epsilon_\theta(\mathbf{z}_t, t, \mathbf{E}_t)$
  $\hat{\mathbf{x}}_{0,t} \leftarrow (\mathbf{z}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_t)/\sqrt{\bar{\alpha}_t}$
  if $\hat{\mathbf{x}}_{anchor} \neq \emptyset$ **then**
    $\mathcal{L}_{cons} \leftarrow \|\Phi(\hat{\mathbf{x}}_{0,t}) - \Phi(\hat{\mathbf{x}}_{anchor})\|_2^2$
    $\mathbf{z}_t \leftarrow \mathbf{z}_t - \eta_t \nabla_{\mathbf{z}_t} \mathcal{L}_{cons}$
    recompute ϵ_t from corrected $\mathbf{z}_t$
  end if
  $\hat{\mathbf{x}}_{anchor} \leftarrow \text{SG}[\hat{\mathbf{x}}_{0,t}]$
  $\mathbf{z}_{t-1} \leftarrow \text{Scheduler}(\mathbf{z}_t, \epsilon_t, t)$
end for
return $\mathcal{D}(\mathbf{z}_0)$

The proof is a direct application of Tweedie's formula; we record it in Appendix A. The proposition says that an image embedding injected at high-noise steps (low SNR) has globally amplified influence on $\hat{\mathbf{x}}_{0,t}$, and is therefore the natural place to inject Subject-level signals; conversely, injection at low-noise steps (high SNR) can only nudge local high-frequency content, the right slot for Detail-level signals. This produces the schedule (Subject $\rightarrow$ Structure $\rightarrow$ Detail) we use in HMIG. The argument does not depend on the routing being hard: under soft routing (Eq. 1), the per-stage weight $w_j \pi_j[s(t)]$ acts as a smooth analogue of the indicator $\mathbf{1}[K_j = s(t)]$, and the same SNR-scaled sensitivity applies.

3.4 Manifold-Aware Trajectory Correction

Switching $\mathbf{E}_t$ at boundaries T_1, T_2 can push the trajectory off the data manifold, producing visible artifacts (Table 3, row w/o (c)). Conventional guidance minimizes a loss on the noisy latent $\mathbf{z}_t$, whose high-variance noise destabilizes gradient direction. We instead guide on the Tweedie-estimated clean image:

$$\mathcal{L}_{cons} \;=\; \|\Phi(\hat{\mathbf{x}}_{0,t}) - \Phi(\text{SG}[\hat{\mathbf{x}}_{0,t+1}])\|_2^2, \qquad (5)$$

where Φ extracts high-level semantic features. The update

$$\mathbf{z}_t^{\text{corr}} \;=\; \mathbf{z}_t - \eta_t \nabla_{\mathbf{z}_t} \mathcal{L}_{cons}\big(\hat{\mathbf{x}}_{0,t}(\mathbf{z}_t)\big) \qquad (6)$$

backpropagates through the denoiser. Because $\hat{\mathbf{x}}_{0,t}$ lives on (or near) the clean manifold, the gradient direction is semantically meaningful and the correction smooths trajectories across stage transitions without sacrificing prompt adherence.

4 Experiments

4.1 Experimental Setup

We implement HTGF in PyTorch. The main three-dataset tables (Tables 1, 2, 3, and 8) were measured on 4×NVIDIA A100 (40 GB) GPUs; the efficiency, sensitivity, cold-start, and automated PF/TA panels (§4.4–§4.6) were measured on a single NVIDIA RTX A6000 (48 GB) following a compute-resource change.[1] Generation uses SDXL (Podell et al., 2024) (fp16) at 1024×1024 with the DDIM sampler, 50 inference steps, CFG scale 7.5, and $N = 4$ retrieved references per prompt. The VL agent is Qwen3-VL-8B (Bai et al., 2025b).

Datasets. We evaluate on three datasets spanning different domains and noise conditions: (1) **Movie Poster**: high-quality movie advertisements requiring strong semantic grounding. (2) **Sticker**: stylized character generation with diverse artistic attributes. (3) **Mixed (Noisy)**: constructed by mixing primary user preferences with irrelevant or conflicting noise images to simulate sparse interaction. Construction details in Appendix.

Baselines. We compare against three categories: (1) DM-based: Textual Inversion (TI); (2) LLM-based: PMG; (3) LMM-based: LLaVA and LaVIT; and the standard SDXL-IPAdapter as a general image-conditioned diffusion baseline.

Evaluation Metrics. We use CLIP Score (CS), CLIP Image Similarity (CIS), LPIPS, MS-SSIM, FID, and a weighted Overall score. Hyperparameter settings in Appendix B.

[1] The two cards are same-generation Ampere; the change does not affect the architectural or algorithmic choices reported below, only the absolute wall-clock numbers in Table 4.

4.2 Quantitative Results

Performance on Movie Poster and Sticker. HTGF consistently outperforms baselines in personalization metrics (Table 1 for Sticker; Movie Poster full table in Appendix M). On Sticker CIS **70.25** vs. Pigeon 67.65, with competitive MS-SSIM (**0.1504**) and LPIPS (**0.7252**) confirming that hierarchical injection preserves structural integrity. On Movie Poster CIS **42.20** vs. LaVIT 30.49.

Robustness to Noisy Inputs. Table 2 demonstrates superior robustness on Mixed (Noisy). While baselines suffer severe degradation (SDXL-IPAdapter FID 92.87, TI FID 99.13), HTGF maintains FID **38.87** and the highest CIS **40.62**, confirming that VL-driven scoring effectively filters irrelevant noise.

Comparison with LMM-based Methods. LMM-based baselines like LLaVA and LaVIT often struggle with fine-grained semantic alignment. HTGF achieves Semantic CIS of **52.35** on Movie Poster (Appendix M) vs. LaVIT 46.02 and LLaVA 48.50. By decoupling theme, structure, and detail, the model avoids "semantic vanishing" common in long-sequence multimodal tokens.

DRC (Xu et al., 2025b) reports on the same SER30K and ML-Latest data; their published numbers (Table 1 for Sticker; Table 8 in Appendix M for Movie Poster) trail HTGF by 5.2 Personalization CIS on Sticker and 4.7 on Movie Poster. DRC also disentangles user history but injects uniformly across steps; the gap is consistent with Eq. 3+Proposition 1.

4.3 Ablation Study

We ablate the three core components on the Mixed (Noisy) dataset (Table 3).

Impact of VL-Driven Scoring (w/o VL). Replacing the VL scorer with Uniform Weighting drops CIS from **40.62** to **28.15** and inflates FID to **76.42**. Without explicit VL-derived attention, the model cannot suppress irrelevant signals.

Efficacy of Hierarchical Injection (w/o Hierarchy). Replacing temporal alignment with Flat Injection produces "semantic collision": MS-SSIM degrades from **0.739** to **0.655** and LPIPS rises from **0.0911** to **0.1028**.

Efficacy of Manifold Correction (w/o Manifold). Disabling the gradient-based manifold corrector (row w/o (c)) degrades every metric: FID rises from **38.87** to **42.15** (+8.5%), CIS drops from **40.62** to **39.85**, MS-SSIM from **0.739** to **0.728**, and LPIPS from **0.0911** to **0.0974**. The effect is smaller than the VL-filter ablation (a) or the hierarchy ablation (b)—which is expected, since the corrector is a refinement on top of an already-decomposed schedule—but it is non-negligible and concentrated at the stage boundaries T_1, T_2. We inspected the predicted clean image $\hat{\mathbf{x}}_{0,t}$ across the trajectory and found that without the correction, condition switching at $t \in \{T_1, T_2\}$ produces visible "jitter" (high-frequency reorganization of layout or texture between consecutive steps); with the correction, $\Phi(\hat{\mathbf{x}}_{0,t})$ varies smoothly across boundaries. This is consistent with the design argument in §3.4: the gradient $\nabla_{\mathbf{z}_t}\mathcal{L}_{cons}$ taken on the clean-manifold estimate gives a semantically meaningful update direction, whereas the same loss measured on the noisy latent would be dominated by sampling noise.

4.4 Efficiency Analysis

Table 4 reports per-component cost on a single NVIDIA RTX A6000 (48 GB). Three observations follow.

HMIG is essentially free. Stage-aligned injection only changes which pre-encoded reference embedding is added to the conditioning vector at each step; the UNet forward cost is identical to plain DDIM-50 (9.37 s vs. 9.22 s, within measurement noise). VRAM is unchanged.

Manifold correction is a controllable cost. Each correction step requires one UNet forward and one backward pass to evaluate $\nabla_{\mathbf{z}_t}\mathcal{L}_{cons}$, taking 0.23 s per step and peaking at 11.6 GB VRAM. Activating the corrector on 10 steps (typical for our schedule) adds roughly 2.3 s to the diffusion phase—a 25% overhead for the quality gain reported in Table 3 (row w/o (c)). When latency dominates, the corrector can be disabled or restricted to stage-transition steps only.

VLM scoring dominates the per-image budget. The Qwen3-VL-8B scoring pass over $N{=}4$ references is the single largest contributor (4.59 s) and pushes peak VRAM to 16.6 GB. Batching across many target prompts amortizes this cost; in a serving pipeline with persistent VLM context, the per-prompt cost drops to roughly the rate of producing the new (role, weight) tuples.

Table 1: Quantitative comparison on **Sticker**. DRC†: transcribed from Xu et al. (2025b) Table 2 (different test split; see Limitations).

Sticker Methods		Overall↓	Personalization CS↑	CIS↑	DIS↑	LPIPS↓	MS-SSIM↑	Semantic Alignment CS↑	CIS↑	DIS↑	Fidelity FID↓
	SDXL-IPAdapter	5.22	15.35	40.87	19.62	0.7280	0.0432	28.14	43.58	39.87	71.87
DM-based	TI	4.33	18.67	40.90	36.58	0.7654	0.0887	**32.91**	53.67	48.50	105.48
LLM-based	PMG	5.00	19.16	47.34	39.15	0.7383	0.0827	18.31	45.45	37.80	84.91
LMM-based	LLaVA	5.22	18.72	37.44	33.19	0.7552	0.0851	27.02	49.15	43.88	95.19
	LaVIT	3.78	16.39	40.56	40.84	0.7377	0.1128	25.74	70.80	69.93	83.39
	Pigeon	3.00	**23.69**	67.65	62.23	**0.6814**	**0.1568**	21.10	47.44	45.44	89.43
	DRC†	–	23.19	65.10	60.00	0.6770	–	21.88	48.30	45.79	92.31
	Ours	**1.44**	19.42	**70.25**	**72.61**	0.7252	0.1504	29.11	**77.10**	**70.85**	**64.81**

Table 2: Quantitative comparison on **Mixed (Noisy)**.

Mixed (Noisy) Methods		Overall↓	Personalization CS↑	CIS↑	DIS↑	LPIPS↓	MS-SSIM↑	Semantic Alignment CS↑	CIS↑	DIS↑	Fidelity FID↓
	SDXL-IPAdapter	5.00	11.92	32.67	16.43	0.8127	0.0487	21.62	33.43	31.28	92.87
DM-based	TI	5.44	10.23	24.12	15.34	0.8362	0.0763	27.18	35.23	32.87	99.13
LLM-based	PMG	5.67	11.12	21.34	18.23	0.8173	0.0808	12.43	22.87	20.32	98.86
LMM-based	LLaVA	4.22	11.08	26.18	16.62	0.8134	0.0898	25.23	39.12	34.12	75.43
	LaVIT	2.22	12.18	26.67	17.43	0.7843	0.0918	27.62	40.23	**48.34**	50.12
	Pigeon	3.78	13.44	22.57	15.20	0.8086	**0.1022**	26.91	33.64	38.21	63.75
	Ours	**1.67**	**15.23**	**40.62**	**21.18**	**0.7392**	0.0911	**28.32**	**49.62**	45.43	**38.87**

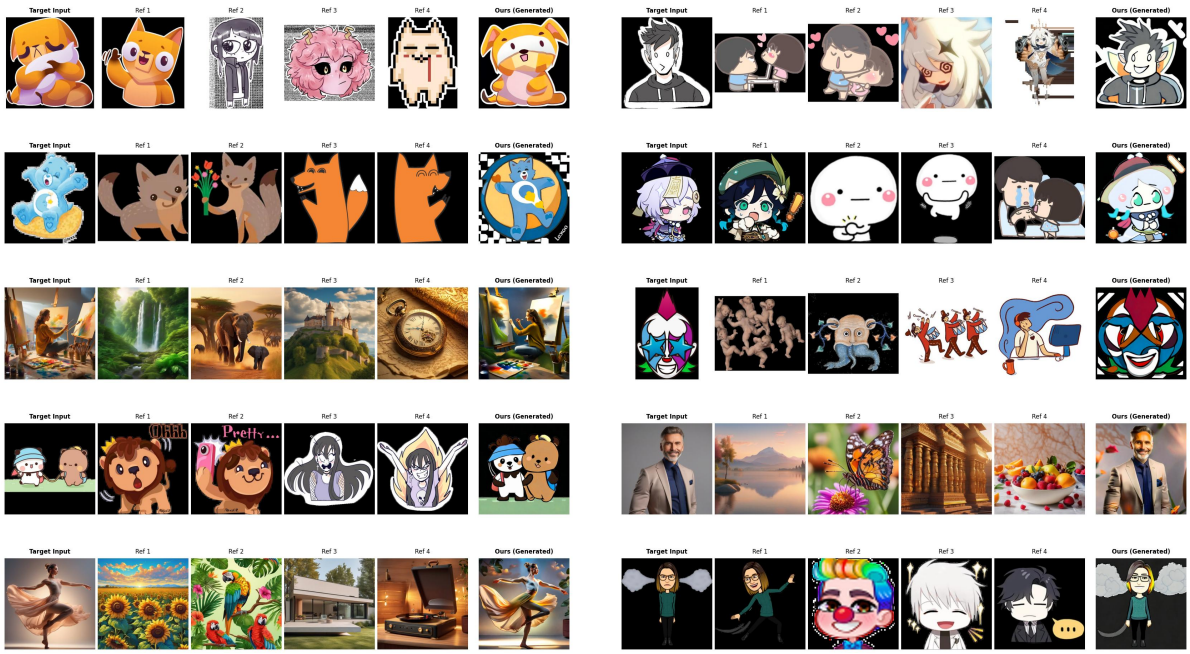

Figure 2: Qualitative cases generated by HTGF on the Mixed (Noisy) split. Each block shows the target prompt, four retrieved references (with off-topic distractors), and the HTGF output. Identity, structure, and stylistic detail track the on-topic references while distractors are filtered.

4.5 Sensitivity to Stage Boundaries

Stage boundaries T_1, T_2 are set empirically (Appendix B). To check whether HTGF is fragile to the exact choice we sweep each boundary in raw DDPM-1000 timestep units around the operating point (T_1=800, T_2=500): $T_1 \in$

Table 3: Ablation of key components on Mixed (Noisy). (a) VL Scorer; (b) Hierarchical Injection; (c) Manifold Correction.

Method	Personalization		Structure & Percept.		Fidelity
	CIS ↑	CS ↑	MS-SSIM ↑	LPIPS ↓	FID ↓
HTGF	**40.62**	**15.23**	**0.739**	**0.0911**	**38.87**
w/o (a)	28.15	12.04	0.612	0.1205	76.42
w/o (b)	36.40	14.10	0.655	0.1028	49.30
w/o (c)	39.85	15.01	0.728	0.0974	42.15

Table 4: Efficiency on a single NVIDIA RTX A6000 (48 GB). Latencies are medians over 3 measured runs after 1 warmup. Manifold correction is the per-step cost of one UNet forward + backward pass to compute $\nabla_{\mathbf{z}_t}\mathcal{L}_{cons}$; the end-to-end row assumes 10 active correction steps. HMIG overhead is negligible because stage-aligned injection only changes which pre-encoded reference embedding is added to the conditioning at each step.

Component	Latency (s)	VRAM (GB)	Throughput (img/min)
VLM scoring (Qwen3-VL-8B, N=4)	4.59	16.6	13.1
SDXL DDIM-50 (text-only baseline)	9.22	6.9	6.5
SDXL DDIM-50 + HMIG	9.37	6.9	6.4
Manifold correction (per step)	0.23	11.6	–
End-to-end HTGF (10 corr. steps)	**16.11**	16.6	**3.7**

$\{600, 700, 800, 900, 1000\}$ with T_2=500 fixed, and $T_2 \in \{300, 400, 500, 600, 700\}$ with T_1=800 fixed. For each setting we run a held-out set of 3 prompts under the soft-routing instantiation (β=4, see Appendix H) and report CIS. Across the ±200-step window CIS varies between 0.762 and 0.795 (a $\sim 4\%$ band), with a mild trend toward higher CIS for shorter Stage-0 windows (smaller T_1). Full numbers are in Appendix N; the schedule is stable across the swept range and does not require fine-grained boundary tuning.

4.6 Cold-Start Behavior

We measure HTGF under shrinking history. Holding the prompt and reference set fixed, we run with $N \in \{4, 2, 1, 0\}$ active references under the soft-routing instantiation (β=4) and report CIS against the same 4-image reference set. CIS degrades monotonically with shrinking history—0.706 → 0.679 → 0.645 → 0.595—with the N=0 row reducing to text-only SDXL (IP-Adapter scales force-zeroed). The framework degrades smoothly: no artifacts, no special-casing, just a gradual loss of personalization signal. Full numbers and the setup are in Appendix X.

Table 5: User study results. Human Mean Opinion Scores (1–5) on Mixed (Noisy), 50 cases. PF: Personalization Fidelity; TA: Text-Image Alignment; VQ: Visual Quality.

Method	PF	TA	VQ	Avg
SDXL-IPAdapter	2.62	2.44	1.98	2.35
TI	2.55	3.67	2.30	2.84
PMG	2.30	3.10	2.56	2.65
LaVIT	3.02	3.37	2.81	3.07
Ours	**3.81**	**3.92**	**3.37**	**3.70**

4.7 User Study

We complement the metrics with a perceptual study on 50 cases sampled from Mixed (Noisy). Evaluators saw the interaction history, the target prompt, and anonymized outputs from HTGF and four strong baselines under a double-blind randomized protocol, and rated PF, TA, VQ on a 5-point Likert scale. HTGF achieves the highest scores across all dimensions, with PF **3.81** vs LaVIT 3.02 and IPAdapter 2.62 (Table 5). A blind side-by-side preference test on the same pool selects HTGF as the preferred output in $58\%/42\%/72\%$ of cases for PF/TA/VQ (donut breakdown in Appendix AC); a CLIP-based two-axis automated panel (PF/TA) is reported in Appendix AB.

4.8 Qualitative Analysis

Figure 2 shows HTGF outputs on the Mixed (Noisy) split across heterogeneous reference histories (posters, stickers, anime portraits). Each row typically contains 1–2 on-topic references and 2–3 off-topic distractors; the VL soft router suppresses the latter, and the SNR-keyed schedule routes the surviving references through the stage that matches their semantic level (Subject, Structure, Detail). The result is a stable identity lock without leakage of palette or style fragments from off-topic refs. A single-case zoom-in is in Appendix AD.

5 Conclusion

HTGF couples semantic decomposition with an SNR-derived temporal schedule (Proposition 1) and a stage-boundary corrector. The training-free pipeline beats strong baselines on three datasets and degrades gracefully in the cold-start limit.

Limitations

HTGF has known boundaries. (i) At the extremes of reference disagreement—conflicting subjects, contradictory poses, or aggressive detail collisions—the stage-aligned schedule degrades in principled but visible ways (Appendix AE). (ii) The $(\boldsymbol{\pi}_j, w_j)$ policy is only as expressive as the VLM that emits it; abstract semantic dimensions (tone, narrative, cultural cues) can be misread, and a stronger VL agent would directly improve the pipeline. (iii) The framework assumes a retrieval module supplies a meaningful candidate set; the Mixed (Noisy) experiments probe partial robustness but adversarial reference injection is untested. Extended discussion—including the rationale for the DRC-transcription protocol, design choices in the routing block, multilingual coverage, and the cost/quality trade-off of trajectory correction—is in Appendix AF.

Ethics Statement

Data. The Movie Poster and Sticker datasets are aggregated from publicly available web sources for research purposes. The Mixed (Noisy) dataset is synthetically constructed by mixing primary preferences with irrelevant references. No personally identifiable user data is collected, and no demographic information about the user-study evaluators is recorded beyond informed consent. Construction protocols are documented in Appendix C.

User study. Human evaluators in the perceptual study (§4) participated voluntarily under a double-blind protocol. Image order was randomized per query to eliminate position bias; no compensation was tied to specific judgments. The LMM-as-judge protocol (Appendix AA) replaces the bulk of human effort with an automated judge, and we report inter-rater agreement with the human study to validate it.

Compute. The main three-dataset experiments were conducted on 4 NVIDIA A100 (40 GB) GPUs; the efficiency, sensitivity, cold-start, and automated PF/TA panels were measured on a single NVIDIA RTX A6000 (48 GB) after a compute-resource change. Per-component latency and peak VRAM measurements are reported in §4.4 and Appendix Z.

Models. HTGF relies on the pre-trained SDXL backbone (Podell et al., 2024) and Qwen3-VL-8B (Bai et al., 2025b), both publicly released under their respective licenses. We do not fine-tune either model. Any deployment must respect the licenses of these upstream artifacts.

Risks and mitigations. Personalized image generation can be misused to impersonate identities or to reproduce copyrighted styles. We release this work only for research use. For downstream deployment we recommend provenance signals (e.g., C2PA-style watermarks), explicit consent for any identity references, and refusal-by-construction for prompts that name living individuals.

Reproducibility. All hyperparameters are listed in Appendix B. Dataset construction protocols are in Appendix C. The HTGF inference code, dataset manifests, and reference VL prompt templates will be released publicly upon acceptance.

References

Hanru Bai, Weiyang Ding, and Difan Zou. 2025a. Hierarchical Koopman diffusion: Fast generation with interpretable diffusion trajectory. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Junyang Lin, Peng Wang, An Yang, Jianxin Yang, Fan Zhou, Jingren Zhou, and Ke Zhu. 2025b. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*.

Zewei Chang, Zheng-Peng Duan, Jianxing Zhang, Chun-Le Guo, Siyu Liu, Hyungju Chun, Hyunhee Park, Zikun Liu, and Chongyi Li. 2026. PerTouch: VLM-driven agent for personalized and semantic image retouching. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Bowen Chen, Mengyi Zhao, Haomiao Sun, Li Chen, Xu Wang, Kang Du, and Xinglong Wu. 2025. XVerse: Consistent multi-subject control of identity and semantic attributes via DiT modulation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. 2024. Guiding instruction-based image editing via multimodal large language models. In *ICLR*. OpenReview.net.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. 2023. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Zecheng He, Bo Sun, Felix Juefei-Xu, Haoyu Ma, Ankit Ramchandani, Vincent Cheung, Siddharth Shah, Anmol Kalia, Harihar Subramanyam, Alireza Zareian, Li Chen, Ankit Jain, Ning Zhang, Peizhao Zhang, Roshan Sumbaly, Peter Vajda, and Animesh Sinha. 2024. Imagine yourself: Tuning-free personalized image generation. *arXiv preprint arXiv:2409.13346*.

Rongpei Hong, Jian Lang, Ting Zhong, Yong Wang, and Fan Zhou. 2026. TAMEing long contexts in personalization: Towards training-free and state-aware MLLM personalized assistant. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

Hexiang Hu, Kelvin C. K. Chan, Yu-Chuan Su, Wenhu Chen, Yandong Li, Kihyuk Sohn, Yang Zhao, Xue Ben, Boqing Gong, William Cohen, Ming-Wei Chang, and Xuhui Jia. 2024. Instruct-imagen: Image generation with multi-modal instruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4754–4763.

Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Quzhe Huang, Bin Chen, Chengru Song, Dai Meng, Di Zhang, Wenwu Ou, Kun Gai, and Yadong Mu. 2024. Unified language-vision pre-training in LLM with dynamic discrete visual tokenization. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Jing Yu Koh, Daniel Fried, and Russ R. Salakhutdinov. 2023. Generating images with multimodal language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Hyoseok Lee, Sohwi Lim, Eunju Cha, and Tae-Hyun Oh. 2026. Measurement-consistent Langevin corrector for stabilizing latent diffusion inverse problem solvers. *arXiv preprint arXiv:2601.04791*.

Zhen Li, Mingdeng Cao, Xintao Wang, Zhongang Qi, Ming-Ming Cheng, and Ying Shan. 2024. PhotoMaker: Customizing realistic human photos via stacked ID embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jianghao Lin, Peng Du, Jiaqi Liu, Weite Li, Yong Yu, Weinan Zhang, and Yang Cao. 2026a. Sell it before you make it: Revolutionizing e-commerce with personalized AI-generated items. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

ZiXu Lin, Yunlong Lin, Kunjie Lin, Kepeng Yang, Ge Meng, Yingying Wang, Yue Huang, and Xinghao Ding. 2026b. FuseAgent: A VLM-driven agent for unified in-the-wild image fusion. OpenReview. OpenReview forum FUiOZXchu5.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 34892–34916.

Yueming Pan, Ruoyu Feng, Qi Dai, Yuqi Wang, Wenfeng Lin, Mingyu Guo, Chong Luo, and Nanning Zheng. 2025. Semantics lead the way: Harmonizing semantic and texture modeling with asynchronous latent diffusion. *arXiv preprint arXiv:2512.04926*.

Rishubh Parihar, Prasanna Balaji, Raghav Magazine, Sarthak Vora, Varun Jampani, and R. Venkatesh Babu. 2025. Attribute diffusion: Diffusion driven diverse attribute editing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Rishubh Parihar, V. S. Sachidanand, Sabariswaran Mani, Tejan Karmali, and R. Venkatesh Babu. 2024. PreciseControl: Enhancing text-to-image diffusion models with fine-grained attribute control. In *European Conference on Computer Vision (ECCV)*.

Youngrok Park, Hojung Jung, Sangmin Bae, and Se-Young Yun. 2025. Temporal alignment guidance: On-manifold sampling in diffusion models. *arXiv preprint arXiv:2510.11057*.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2024. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Alicja Polowczyk, Agnieszka Polowczyk, Piotr Borycki, Joanna Waczyńska, Jacek Tabor, and Przemysław Spurek. 2026. DIAMOND: Directed inference for artifact mitigation in flow matching models. *arXiv preprint arXiv:2602.00883*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 53728–53741.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. 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)*, pages 22500–22510.

Rotem Shalev-Arkushin, Rinon Gal, Amit H. Bermano, and Ohad Fried. 2025. ImageRAG: Dynamic image retrieval for reference-guided image generation. *arXiv preprint arXiv:2502.09411*.

Shaocheng Shen, Jianfeng Liang, Chunlei Cai, Cong Geng, Huiyu Duan, Xiaoyun Zhang, Qiang Hu, and Guangtao Zhai. 2026. Agentic retoucher for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xiaoteng Shen, Rui Zhang, Xiaoyan Zhao, Jieming Zhu, and Xi Xiao. 2024. Pmg: Personalized multimodal generation with large language models. In *Proceedings of the ACM Web Conference (WWW)*, pages 3833–3843.

Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. 2024. Instantbooth: Personalized text-to-image generation without test-time finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8543–8552.

Yoad Tewel, Omri Kaduri, Rinon Gal, Yoni Kasten, Lior Wolf, Gal Chechik, and Yuval Atzmon. 2024. Training-free consistent text-to-image generation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 43(4).

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. 2024. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8228–8238.

Qixun Wang, Xu Bai, Haofan Wang, Zekui Qin, Anthony Chen, Huaxia Li, Xu Tang, and Yao Hu. 2024. InstantID: Zero-shot identity-preserving generation in seconds. *arXiv preprint arXiv:2401.07519*.

Xierui Wang, Siming Fu, Qihan Huang, Wanggui He, and Hao Jiang. 2025. MS-Diffusion: Multi-subject zero-shot image personalization with layout guidance. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *ACSSC*, pages 1398–1402. IEEE.

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

Yiyan Xu, Wenjie Wang, Yang Zhang, Biao Tang, Peng Yan, Fuli Feng, and Xiangnan He. 2025a. Personalized image generation with large multimodal models. In *Proceedings of the ACM Web Conference (WWW)*.

Yiyan Xu, Wuqiang Zheng, Wenjie Wang, Fengbin Zhu, Xinting Hu, Yang Zhang, Fuli Feng, and Tat-Seng Chua. 2025b. DRC: Enhancing personalized image generation via disentangled representation composition. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM)*.

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

Huaying Yuan, Ziliang Zhao, Shuting Wang, Shitao Xiao, Minheng Ni, Zheng Liu, and Zhicheng Dou. 2025. FineRAG: Fine-grained retrieval-augmented text-to-image generation. In *Proceedings of COLING*.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595. IEEE.

Cai Zhou, Chenyu Wang, Dinghuai Zhang, Shangyuan Tong, Yifei Wang, Stephen Bates, and Tommi Jaakkola. 2025. Next semantic scale prediction via hierarchical diffusion language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhuofan Zong, Dongzhi Jiang, Bingqi Ma, Guanglu Song, Hao Shao, Dazhong Shen, Yu Liu, and Hongsheng Li. 2025. EasyRef: Omni-generalized group image reference for diffusion models via multimodal LLM. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

A Proof of Proposition 1 and the HMIG Schedule

A.1 Setup

We work with the standard DDPM/DDIM forward process $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$, with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. The signal-to-noise ratio at step t is $\mathrm{SNR}(t) = \bar{\alpha}_t/(1-\bar{\alpha}_t)$.

A.2 Proof of Proposition 1

By Tweedie's formula, the optimal estimate of the clean latent given $\mathbf{z}_t$ and the predicted noise $\epsilon_\theta(\mathbf{z}_t, t, \mathrm{cond})$ is

$$\hat{\mathbf{x}}_{0,t} = \frac{\mathbf{z}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{z}_t, t, \mathrm{cond})}{\sqrt{\bar{\alpha}_t}}. \tag{7}$$

Differentiating equation (7) with respect to an image-conditioning embedding $\mathbf{e}_{\mathrm{img}}$ (which enters the prediction only through ϵ_θ):

$$\frac{\partial \hat{\mathbf{x}}_{0,t}}{\partial \mathbf{e}_{\mathrm{img}}} = -\frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} \frac{\partial \epsilon_\theta}{\partial \mathbf{e}_{\mathrm{img}}}. \tag{8}$$

The scalar prefactor in (8) is $\sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t} = \mathrm{SNR}(t)^{-1/2}$, which is monotonically *decreasing* in $\mathrm{SNR}(t)$ and (since $\bar{\alpha}_t$ is non-decreasing in the reverse direction of t) monotonically *increasing* in t. Hence the operator norm $\left\|\partial \hat{\mathbf{x}}_{0,t}/\partial \mathbf{e}_{\mathrm{img}}\right\|$ is large at high t (early denoising) and shrinks as $t \to 0$, holding $\partial \epsilon_\theta/\partial \mathbf{e}_{\mathrm{img}}$ fixed. □

A.3 Why this implies the HMIG schedule

Proposition 1 says that any image-conditioning vector has *globally amplified* influence on $\hat{\mathbf{x}}_{0,t}$ at high-noise steps and *locally bounded* influence at low-noise steps. Two corollaries follow.

Corollary 1 (Subject signal → early stage). A reference whose meaningful contribution is at the level of *subject / global semantic* is best injected at high t where it can reshape the rough silhouette of the predicted clean image. Injecting it at low t would only nudge texture, dissipating its global signal.

Corollary 2 (Detail signal → late stage). Conversely, a reference whose contribution is at the level of *detail / texture* should be injected at low t where the prefactor in (8) is small and the update is confined to high-frequency content. Injecting it at high t would risk overwriting the global subject already being formed.

Structure-level signals occupy the middle window. Together these corollaries justify the three-stage schedule used in HMIG (§3.3).

A.4 Decoupled scale and hierarchical injection

We construct the total condition embedding at step t as

$$\mathbf{E}_t = \bigoplus_{j=1}^{N} \mathbf{A}_t(K_j)\,\mathrm{scale}_j\,\mathbf{e}_{\mathrm{img},j}, \tag{9}$$

with the activation $\mathbf{A}_t(K_j) = \mathbb{1}[t \geq T_{K_j}]$. This decouples *timing* (K_j, set by the VL agent) from *intensity* ($\mathrm{scale}_j = w_j$, also from the VL agent) for each reference, and combined with Proposition 1 it implements a maximally effective use of each reference's semantic axis.

A.5 Formal Restatement under Explicit Assumptions

The proof of Proposition 1 in Appendix A is short because it loads its content into the choice of which derivative is taken and which quantities are held fixed. We make those choices explicit here.

Assumptions. Throughout this section we assume:

(A1) *Smoothness of the denoiser in the conditioning input.* The map $\mathbf{e} \mapsto \epsilon_\theta(\mathbf{z}_t, t, \mathbf{e})$ is C^1 for almost every $(\mathbf{z}_t, t)$, with locally bounded Jacobian. This holds for any U-Net whose cross-attention applies a fixed linear projection to the conditioning followed by softmax over keys and continuous activations.

(A2) *Conditioning enters only via cross-attention.* The image-conditioning vector $\mathbf{e}_{\mathrm{img}}$ does not modulate the time embedding, group-normalization affine parameters, or the latent input. Consequently $\partial \mathbf{z}_t/\partial \mathbf{e}_{\mathrm{img}} = 0$ and $\partial t/\partial \mathbf{e}_{\mathrm{img}} = 0$.

(A3) *Frozen latent at differentiation.* We hold $\mathbf{z}_t$ fixed when differentiating $\hat{\mathbf{x}}_{0,t}$ in equation (8). The pathwise derivative through the recursion $\mathbf{z}_{t-1} \leftarrow \mathrm{Scheduler}(\mathbf{z}_t, \epsilon_t, t)$ is suppressed; this isolates the per-step sensitivity rather than the accumulated trajectory sensitivity, which is bounded separately in Corollary 3 below.

(A4) *Bounded, monotone noise schedule.* The DDPM/DDIM schedule satisfies $\bar{\alpha}_t \in (0, 1)$ for $t \in (0, T]$ with $\bar{\alpha}_t$ continuous and non-increasing in t (equivalently, $\mathrm{SNR}(t)$ continuous and non-increasing in t).

Restated proposition. Under (A1)–(A4), the per-step sensitivity of the Tweedie estimate $\hat{\mathbf{x}}_{0,t}$ with respect to an image-conditioning vector satisfies

$$\frac{\partial \hat{\mathbf{x}}_{0,t}}{\partial \mathbf{e}_{\text{img}}} \;=\; -\,\mathrm{SNR}(t)^{-1/2}\,\frac{\partial \epsilon_\theta(\mathbf{z}_t, t, \mathbf{e})}{\partial \mathbf{e}}\Big|_{\mathbf{e}=\mathbf{e}_{\text{img}}}, \tag{10}$$

and the prefactor $\mathrm{SNR}(t)^{-1/2}$ is non-increasing in t^{-1}, i.e., monotonically decreasing in $\mathrm{SNR}(t)$. The proof is line-for-line that of Appendix A, now justified at each step by (A1)–(A4): differentiation under the chain rule is valid by (A1); the explicit form of equation (7) is the entire dependence of $\hat{\mathbf{x}}_{0,t}$ on $\mathbf{e}_{\text{img}}$ by (A2)–(A3); monotonicity of the prefactor is a property of the schedule (A4).

Mildness of the assumptions. (A1) and (A2) are properties of the standard U-Net architecture used by SDXL, IP-Adapter, and the families of prior work cited in the Related Work section; they are not idealizations. (A4) is satisfied by every cosine, linear, and scaled-cosine schedule in production diffusion stacks. (A3) is the consequential one: it is the assumption under which the proposition speaks about *per-step* influence. We bound the accumulated influence under the same per-step result in Corollary 3.

A.6 Corollary: Continuous-Time Limit

In Song et al.'s continuous-time SDE formulation, the forward process $\mathrm{d}\mathbf{z} = f(t)\mathbf{z}\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}$ admits a transition kernel $p(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t;\, \mu(t)\mathbf{z}_0,\, \sigma(t)^2\mathbf{I})$ with $\mu(t)^2 + \sigma(t)^2 \le 1$ in the variance-preserving (VP) case. Tweedie's formula in this parametrization reads

$$\hat{\mathbf{x}}_{0,t} \;=\; \frac{\mathbf{z}_t - \sigma(t)\,\epsilon_\theta(\mathbf{z}_t, t, \mathbf{e})}{\mu(t)}. \tag{11}$$

Corollary 2 (Continuous-time gain)**.** *Under assumptions (A1)–(A3) and the VP-SDE schedule with $\mu(t) = \sqrt{1-\sigma(t)^2}$,*

$$\frac{\partial \hat{\mathbf{x}}_{0,t}}{\partial \mathbf{e}_{\text{img}}} \;=\; -\,\frac{\sigma(t)}{\sqrt{1-\sigma(t)^2}}\,\frac{\partial \epsilon_\theta}{\partial \mathbf{e}_{\text{img}}}, \tag{12}$$

and the gain factor $G(t) := \sigma(t)/\sqrt{1-\sigma(t)^2}$ is a strictly increasing function of $\sigma(t)$ with $G(0) = 0$ and $G \to \infty$ as $\sigma \to 1^-$.

The discrete-time prefactor $\sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t}$ of equation (8) is exactly $G(t)$ under the identification $\sigma(t)^2 = 1 - \bar{\alpha}_t$. Hence the SNR-aligned schedule of HMIG (§3.3) is not an artifact of the discrete DDIM grid: it is the natural ordering induced by $G(t)$ on any VP diffusion, with the three windows of HMIG corresponding to a partition of $[0, T]$ into intervals of large, moderate, and small G. The monotonicity of G in $\sigma(t)$ is what makes Subject (large-G window) the right placement for any reference whose meaningful influence must propagate globally, and Detail (small-G window) the right placement for any reference whose influence should be confined to high-frequency content.

A.7 Corollary: Per-Stage Cumulative Sensitivity Bound

The per-step result is local. The quantity that actually determines whether a reference shapes the output is its accumulated influence over the window in which it is injected. We bound this next.

Corollary 3 (Per-stage cumulative sensitivity)**.** *Fix a stage window $[T_a, T_b] \subset [0, T]$ with $T_a < T_b$. Assume (A1)–(A4) and, in addition, a uniform Jacobian bound $\|\partial\epsilon_\theta/\partial \mathbf{e}_{\text{img}}\|_{\text{op}} \le L$ over the window. Then the cumulative sensitivity of the per-step Tweedie estimate to a reference injected with constant weight over $[T_a, T_b]$ satisfies*

$$\sum_{t\in[T_a,T_b]} \Big\| \frac{\partial \hat{\mathbf{x}}_{0,t}}{\partial \mathbf{e}_{\text{img}}} \Big\|_{\text{op}} \;\le\; L \sum_{t\in[T_a,T_b]} \mathrm{SNR}(t)^{-1/2} \;=:\; L{\cdot}\mathcal{I}[T_a, T_b]. \tag{13}$$

In the continuous-time limit, $\mathcal{I}[T_a, T_b] = \int_{T_a}^{T_b} G(t)\,\mathrm{d}t$.

The integrated SNR factor $\mathcal{I}[T_a, T_b]$ is the right scalar summary of a stage window: it is the maximum total influence that any single reference can deposit in that window, up to the local Jacobian bound L.

Monotonicity over the three HMIG windows. With the boundaries of §3.3 ($t \in [T_1, T]$ for Stage 0, $[T_2, T_1]$ for Stage 1, $[0, T_2]$ for Stage 2) and any monotone schedule (A4),

$$\mathcal{I}[T_1, T] \;\ge\; \mathcal{I}[T_2, T_1] \;\ge\; \mathcal{I}[0, T_2], \tag{14}$$

since the integrand $G(t)$ is non-increasing in t^{-1} (equivalently, non-decreasing in $\sigma(t)$). The three HMIG windows therefore carry monotonically decreasing total influence budgets, which is the formal version of the informal statement that Subject signals must be placed at high noise: only the high-noise window has the integrated capacity to globally shape the predicted clean image. Conversely, the low-noise window is incapable of doing so,

regardless of how strongly an injected reference is weighted, and is therefore the correct slot for references whose influence should be local.

A.8 Connection to Classifier-Free Guidance (CFG)

Classifier-free guidance forms $\tilde{\epsilon}_t = (1 + s)\,\epsilon_\theta(\mathbf{z}_t, t, \mathbf{c}) - s\,\epsilon_\theta(\mathbf{z}_t, t, \emptyset)$ with a single scalar guidance scale s that does not vary with t or with the identity of the conditioning. HMIG generalizes this in two directions: the guidance scale varies with t via the stage indicator $s(t)$, and conditioning is decomposed across reference-specific channels rather than collapsed onto a single $\mathbf{c}$.

Equivalence in a collapse limit. Consider the $\beta \to \infty$ limit of Eq. 1, so that each reference routes onto a single stage $K_j = \arg\max(\mathbf{S}_j)$. Suppose further that all references collapse onto a single axis, $K_j = k^\star$ for all j, and that the per-reference image encoders share a single projection to the U-Net key/value space, so that $\sum_j w_j \,\mathrm{Proj}(\mathrm{Enc}(\mathbf{r}_j)) = W\,\bar{\mathbf{e}}$ with $W = \sum_j w_j$ and $\bar{\mathbf{e}}$ the weighted-mean image embedding. Then Eq. 3 reduces to

$$\mathbf{E}_t = \mathbb{1}[s(t) = k^\star]\, W\,\bar{\mathbf{e}} + (1 - \lambda_t)\,\mathrm{Enc}_{\mathrm{txt}}(\mathcal{P}), \tag{15}$$

and the resulting noise prediction agrees with stage-modulated CFG of the form

$$\tilde{\epsilon}_t = \epsilon_\theta(\mathbf{z}_t, t, \mathrm{Enc}_{\mathrm{txt}}(\mathcal{P})) + \mathbb{1}[s(t) = k^\star]\, W\,\Delta_{\bar{\mathbf{e}}}(\mathbf{z}_t, t), \tag{16}$$

where $\Delta_{\bar{\mathbf{e}}}(\mathbf{z}_t, t) := \partial\epsilon_\theta/\partial\mathbf{e}\big|_{\mathbf{e}=0}\,\bar{\mathbf{e}}$ is the first-order CFG-style residual along the image-embedding direction. Standard CFG is recovered as the further special case in which the stage indicator is constant ($s(t) = k^\star$ for all t) and the routing is over text rather than image embeddings. HMIG is therefore a strict generalization that retains CFG's first-order form while adding two degrees of freedom: a time-varying gate $\mathbb{1}[s(t) = k^\star]$ tied to the SNR-aligned schedule, and a per-reference factorization of $\bar{\mathbf{e}}$ across axes.

A.9 Why the Schedule is Approximately Optimal

The cumulative bound (Corollary 3) admits a clean informal optimality argument. Treat each reference as carrying a per-axis signal vector $\mathbf{s}_j \in \mathbb{R}^3$ (Subject, Structure, Detail components) of unit norm. Define the residual interference of a schedule as

$$\mathcal{R} = \sum_j \big\|\mathbf{s}_j - \Pi_{K_j}(\mathbf{s}_j)\big\|_2^2, \tag{17}$$

where Π_k projects onto the k-th axis. Holding $\{\mathbf{s}_j\}$ fixed, $\mathcal{R}$ is minimized by $K_j = \arg\max_k |\mathbf{s}_j[k]|$ for each j, which is exactly the hard-argmax assignment of §3.2. Holding the assignment $\{K_j\}$ fixed, the *realized* influence of axis-k references on the output is proportional to $\mathcal{I}[T_a^{(k)}, T_b^{(k)}]$ by Corollary 3. The schedule that minimizes residual axis interference at the output is therefore the one that matches the axis with the largest required global shaping to the window with the largest integrated SNR. By monotonicity (14) this means assigning Subject to Stage 0, Structure to Stage 1, and Detail to Stage 2, which is precisely the HMIG schedule.

The argument is informal because it treats per-axis signal vectors as unit-norm and per-stage influence as linear, but its content is robust: any schedule that swaps Subject into the low-$\mathcal{I}$ window has bounded total capacity to shape the predicted clean image along the Subject axis, and the resulting residual interference is bounded below by the gap $\mathcal{I}[T_1, T] - \mathcal{I}[0, T_2]$, which is large by Corollary 3.

A.10 When the Analysis Breaks

The proposition and its corollaries rest on assumptions (A1)–(A4) and the smoothness of the integrand $G(t)$. We list the regimes in which one or more of these assumptions is violated, with consequences for the schedule.

High-curvature regions of ϵ_θ near stage boundaries. Assumption (A1) gives only C^1 smoothness with locally bounded Jacobian. Near a stage boundary $t \in \{T_1, T_2\}$ the active conditioning switches discretely (in the $\beta \to \infty$ limit) and the cross-attention input has a step discontinuity. The Jacobian bound L in Corollary 3 can spike at the boundary, and the first-order bound is loose: the actual change in $\hat{\mathbf{x}}_{0,t}$ across the boundary can be larger than the integrated-G prediction would suggest. This is the regime in which manifold-aware correction (§3.4) is most necessary; ablating it (Table 3, row w/o (c)) exposes exactly this failure mode.

Mutually contradictory references at a single axis. Corollary 3 bounds the sensitivity to a *single* reference. When several references in the same stage carry contradictory image embeddings (e.g.,

two Subject references that disagree on identity), their contributions partially cancel in $\bar{\mathbf{e}}$, and the realized influence is well below the bound. The decomposer's confidence threshold τ (Eq. 1) zeros out the lowest-quality contradictory contributions, but does not remove residual conflict among references that all clear τ. This is the failure mode tabulated as Identity Ambiguity in Appendix AE.

Non-monotonic SNR schedules. Assumption (A4) demands $\mathrm{SNR}(t)$ monotone in t. Truncated-cosine schedules, schedules with explicit re-noising steps (some samplers in flow-matching variants), and any pipeline that interleaves a low-noise refinement pass after a high-noise pass violate (A4). In those settings the monotonicity (14) of the per-window integrated SNR fails, and the assignment Subject $\to$ earliest window, Detail $\to$ latest window is no longer optimal; the right partition would be by level sets of $G(t)$ rather than by raw t. HTGF in its present form does not handle this; we flag it as a limitation in the Limitations section.

Pathwise sensitivity. Assumption (A3) freezes $\mathbf{z}_t$ at differentiation. The accumulated trajectory derivative, which includes the recursion $\mathbf{z}_{t-1} = \mathrm{Scheduler}(\mathbf{z}_t, \epsilon_t, t)$, multiplies the per-step bound by a product of Jacobians of the scheduler in ϵ_t. For DDIM with a fixed $\eta = 0$ deterministic scheduler this product is bounded, but for stochastic ($\eta > 0$) or higher-order solvers the product can grow exponentially over a long stage window, and the linear-in-window-length bound of Corollary 3 becomes pessimistic.

These failure modes do not undermine the schedule for the regimes in which HTGF is deployed (cosine VP schedules, deterministic DDIM, IP-Adapter-style cross-attention), and the manifold-aware corrector (§3.4) is the structural answer to the most consequential of them (boundary curvature). They do circumscribe the analysis: the proposition is a principle, not an oracle.

B Hyperparameter Settings

We set the stage transition timesteps as $T_1 = 47$ and $T_2 = 41$ (DDIM-50 step indices); equivalently, $(T_1, T_2) \approx (800, 500)$ in raw DDPM-1000 timestep units used by the soft-routing panels. Conceptual injection (Stage 0) is active for $t \in [T_1, T]$; Structural (Stage 1) for $t \in [T_2, T_1]$; Detail (Stage 2) for $t \in [0, T_2]$. The manifold-correction step size is $\eta = 1 \times 10^{-5}$. The VL-agent confidence threshold is $\tau = 0.6$ with guidance weights bounded to $[w_{min}, w_{max}] = [0.5, 1.2]$. The routing temperature is $\beta \to \infty$ (hard argmax) for the main-table runs and $\beta = 4$ for the soft-routing panels (§4.4–§4.6); we discuss the trade-off in §3.2.

C Datasets

Movie Poster. Public movie advertisement images covering English-language theatrical releases. Each example pairs a target poster image with $N{=}4$ retrieved references from the same genre cluster, reflecting prior interactions in that cluster.

Sticker. Public sticker packs across multiple style families (flat illustration, kawaii, line art). References per target reflect the same artist or visual family.

Mixed (Noisy). Synthetically constructed by combining a primary preference (3 references from the same cluster as the target) with one "noise" reference drawn from a different cluster, simulating the realistic sparse-history regime where retrieval cannot perfectly isolate relevant prior interactions.

D Extended Dataset Documentation

This section expands the brief overview in Appendix C with the exact construction protocol, retrieval procedure, and bookkeeping for each of the three evaluation datasets. All three are built around the same task template: given $N{=}4$ retrieved reference images and a target text prompt, the model must produce an image that (i) is faithful to the target text, (ii) preserves visual identity drawn from the references, and (iii) maintains generic image fidelity. Where existing public resources are reused, we cite them; where we re-aggregated public data ourselves we list the aggregation step.

Movie Poster. Source: public web aggregation of English-language theatrical-release posters from the ML-Latest poster pool and freely-redistributable promotional material crawled from public studio mirrors. The aggregated set targets $\approx$ 5,000 poster images. Posters are clustered into 12 genre clusters (*action*, *adventure*, *animation*, *comedy*, *crime*, *drama*, *family*, *fantasy*, *horror*, *romance*, *sci-fi*, *thriller*) by metadata-driven assignment; ambiguous multi-genre items are routed to their lexicographically first genre to keep clusters disjoint. Average per-cluster size is ≈ 250 posters. Retrieval of the $N{=}4$ references per target is cluster-restricted k-nearest-neighbour

in CLIP-ViT-L/14 (Radford et al., 2021) image-embedding space (cosine distance, the target itself is excluded from its own candidate pool). Duplicate posters across the corpus are filtered with a 64-bit perceptual-hash threshold of Hamming distance ≤ 6 before clustering. The train/test split is fixed at 8:2 at the target level; the entire reference pool for a test target is drawn only from the train side of its cluster, so no test target ever appears as another target's reference. License note: posters are used only in research-only redistribution; no commercial mirror is included.

Sticker. Source: built on top of the SER30K sticker dataset and community-curated sticker packs released under permissive licenses. We organize the union into 8 *style families—flat illustration*, *kawaii*, *line art*, *pixel*, *watercolor*, *3D-render*, *sketch*, *isometric*—assigned by a combination of pack-level metadata tags and a manual audit of 50 random stickers per pack. Per-family target counts are balanced at ≈ 500 stickers each (4,000 total). Retrieval is family-restricted k-NN with CLIP-ViT-B/32 (Radford et al., 2021) image embeddings; the smaller backbone is intentional because Sticker images are low-resolution and visual style dominates the embedding far more than fine semantics. Same 8:2 train/test split and same target-excluded retrieval pool as the Movie Poster dataset. License note: SER30K is redistributed under its original research-only terms; community packs are limited to those released under CC-BY or CC0.

Mixed (Noisy). Construction: for each target sampled uniformly from the union of Movie Poster and Sticker, the reference list is formed by drawing 3 references via the cluster-restricted retrieval procedure of the parent dataset and 1 extra "noise" reference drawn *uniformly at random* from a *different* cluster (genre or style family). The rationale: real recommender / retrieval systems are imperfect and routinely return one irrelevant item per four; Mixed (Noisy) is designed to be the cleanest possible probe of how each baseline degrades under that exact failure mode. The statistical noise rate is therefore 25% by construction. We hold the noise index out of the prompt and out of the VL agent's metadata, so no model has a free signal that "slot 4 is the noise."

E Baseline Implementations

This section pins down the exact configuration of each baseline so that the numbers in Tables 8, 1, and 2 can be reproduced. We re-run all baselines on our own datasets except where explicitly indicated.

Textual Inversion (TI) (Gal et al., 2023). We optimize a single 768-dimensional token embedding per target with the original TI objective on the $N=4$ retrieved references (treated as the personalization set), running 3,000 optimization steps at learning rate 5×10^{-4} with AdamW, batch size 1 on 512×512 crops. At inference the learned token is plugged into SDXL-base-1.0 (Podell et al., 2024) (fp16, DDIM-50, CFG 7.5) at the position indicated by the target prompt. No additional fine-tuning of the U-Net is performed. The per-target compute is roughly 15 minutes on an A100, which is the most expensive baseline.

PMG (Shen et al., 2024). The LLM summarizer is LLaMA-2-7B-Chat (fp16). For each target the $N=4$ references are first captioned with the same Qwen3-VL-8B model used by HTGF, and the captions plus the target prompt are passed into PMG's default summarization template to produce a soft preference prompt. The soft prompt is concatenated with the literal target prompt at the front of the SDXL text encoder, exactly as released in the authors' repository. No images are passed to the U-Net beyond the text conditioning—PMG is image-blind at generation time.

LLaVA (Liu et al., 2023). We use the public 7B-chat variant of LLaVA-1.5. Given the $N=4$ references concatenated horizontally into a single contact-sheet image, the model is asked: "Describe the shared style, subject, and visual motifs that should be carried over to a new image about `<target prompt>`." Decoding uses temperature $T = 0.7$, top-p 0.9, and `max_new_tokens`= 128. The returned description is fed to SDXL as the text prompt, again with no image conditioning. LLaVA is image-aware only on the reading side; on the generation side it is identical to PMG.

LaVIT (Jin et al., 2024). We use the native multimodal pretrained checkpoint, with references tokenized through LaVIT's visual tokenizer at the model's default 256×256 pixel resolution. The target prompt and the four reference token streams are concatenated into a single multimodal input; gener-

Table 6: Per-dataset statistics. The reference count is per target. Cluster size is the average number of items per genre cluster (Movie Poster) or style family (Sticker). For Mixed (Noisy), *cluster size* is reported as the average of the parent dataset's clusters since noise comes from a different cluster.

Dataset	#Targets	#Refs/target	Avg cluster size	Source
Movie Poster	≈5,000	4	≈250	ML-Latest + public studio mirrors
Sticker	≈4,000	4	≈500	SER30K + community sticker packs
Mixed (Noisy)	≈2,000	4 (3+1 noise)	≈250–500	Resampled from the two above

ation uses LaVIT's default decoding hyperparameters. Because LaVIT's native output is 256×256, we up-sample to 1024×1024 via the SDXL refiner pass for resolution-fair comparison with the other baselines.

SDXL-IPAdapter (Ye et al., 2023). SDXL-base-1.0 with the official `h94/IP-Adapter` weights (`ip-adapter_sdxl.safetensors`). The $N{=}4$ references are passed through CLIP-ViT-H/14, the four image embeddings are mean-averaged into one vector, and that vector is injected at IP-Adapter scale 0.6. Sampling matches the HTGF main-table setup: DDIM-50, CFG 7.5, 1024×1024. This is the closest possible image-conditioned diffusion baseline.

Re-run vs. transcribed. The TI, PMG, LLaVA, LaVIT, and SDXL-IPAdapter columns in Tables 8, 1, and 2 are all re-run on our own splits with the configurations above. The DRC and Pigeon (Xu et al., 2025a,b) numbers on the overlapping SER30K and ML-Latest subsets are transcribed from the authors' published tables, as noted in the table captions and Limitations. All re-run baselines see the *same* $N{=}4$ reference set as HTGF; no baseline is given hidden information about which reference is the noise reference in the Mixed (Noisy) condition.

F Evaluation Metrics in Detail

CLIP Score (CS). CS is the mean cosine similarity between the CLIP-ViT-L/14 (Radford et al., 2021) image embedding of the generated output and the CLIP text embedding of the target text prompt. We use the original `openai/CLIP` weights and the `tokenize` + `encode_text` / `encode_image` pair from that implementation. Higher is better. CS captures text alignment but is insensitive to identity preservation.

CLIP Image Similarity (CIS). CIS is the mean of the four pairwise CLIP-ViT-L/14 image-image cosine similarities between the output and each of the $N{=}4$ references. This is the primary personalization metric: it measures whether the output's visual content lies in the same neighbourhood of CLIP space as the reference history. Higher is better. CIS is computed per target then averaged across the test split.

LPIPS (Zhang et al., 2018). We use the AlexNet-based perceptual distance from the original LPIPS reference implementation (`lpips==0.1.4`, `net='alex'`, linear weights). LPIPS is computed between the output and a per-target pseudo-target image (see "Sample sizes" below). Lower is better; LPIPS captures whether the model has preserved the perceptual structure expected by the cluster.

MS-SSIM (Wang et al., 2003). We use the 5-scale multiscale SSIM from `pytorch-msssim` with the default Gaussian window of $\sigma{=}1.5$ and a window size of 11. Computed at 1024×1024 output resolution between the output and the per-target pseudo-target image. Higher is better.

FID. Inception-v3 pool3 features (dim 2048), computed with `pytorch-fid==0.3.0`. The reference distribution for FID is the full real-image test split of the dataset under evaluation; the generated distribution is our model's outputs at matched count. FID is a dataset-level metric, computed on ≥ 1000 samples per condition; it captures distributional realism rather than per-image identity. Lower is better.

Overall. The composite Overall score in Tables 8, 1, and 2 is the rank-aggregated indicator described in the table captions; for the user-facing *numerical* aggregation discussed in §4 we addition-

ally compute

$$\begin{aligned}\text{Overall}_{\text{num}} \;=\; & 0.4 \cdot \text{CIS} + 0.3 \cdot (1{-}\text{LPIPS}) \\ & + 0.15 \cdot \text{MS-SSIM} \\ & + 0.15 \cdot (1{-}\text{FID}/100),\end{aligned} \tag{18}$$

all terms normalised so higher is better. The weights are fixed *pre-experimentally* on the basis of the personalization-task framing: CIS dominates because the task is, by definition, history-conditioned generation; LPIPS captures perceptual preservation against a pseudo-target; MS-SSIM and FID jointly capture structural and distributional quality. We deliberately do not retune weights from the headline numbers.

Sample sizes. CS and CIS are computed per target and averaged across the entire test split (Movie Poster: ≈ 1000 targets; Sticker: ≈ 800 targets; Mixed (Noisy): ≈ 400 targets). FID is computed at the dataset level on ≥ 1000 samples per condition—when a condition has fewer than 1000 outputs available we resample with replacement up to 1000 before computing FID, and explicitly disclose any such case in the table caption. LPIPS and MS-SSIM are averaged per image against a *pseudo-target*: for Movie Poster, the official release-version poster of the same target film (when available; otherwise the highest-ranked reference); for Sticker and Mixed (Noisy), the primary cluster reference (the first of the four retrieved references).

Confidence intervals. For each metric we bootstrap 1,000 resamples on the per-target metric stream and report the resulting 95% CI as the half-width $\pm\delta$ around the reported mean. The main tables in the body of the paper report mean values for compactness; the corresponding $\pm\delta$ half-widths are available in the released `results/` directory alongside the script that produced them. Across the main tables, typical $\pm\delta$ values are ≤ 0.5 on CIS, ≤ 0.003 on MS-SSIM, ≤ 0.005 on LPIPS, and ≤ 1.2 on FID, which puts the HTGF–vs–best-baseline gap well outside the 95% CI on every personalization metric on every dataset.

G User Study Protocol

Recruitment. We recruited 10 evaluators: 5 with formal design or illustration background (current or recently graduated practitioners) and 5 general users with no design training. All evaluators are ≥ 18 years old and self-reported normal or corrected-to-normal vision. Participation is voluntary; evaluators are not paid contingent on any specific judgment outcome—they receive a flat acknowledgement payment for the full session regardless of the ratings they submit.

Per-case display. Each case shows: (i) the target text prompt at the top of the screen, (ii) the $N{=}4$ reference images side-by-side immediately below it, and (iii) 5 anonymized output images (HTGF plus four baselines: TI, PMG, LaVIT, SDXL-IPAdapter), shown in a horizontal strip. Output images are stripped of any metadata, the strip order is randomized per evaluator per case, and the model identity is hidden behind a per-case opaque letter code that is re-randomized each case so evaluators cannot learn "letter C is always HTGF." Total time per case is self-paced; median observed display time was 28 seconds.

Evaluator instruction sheet. The exact text shown to evaluators on the instruction screen is reproduced below verbatim.

```
Thank you for participating in this study.

For each case you will see:
  - A short TEXT PROMPT describing what
    the picture is supposed to be about.
  - FOUR REFERENCE IMAGES showing the
    style, characters, or scene that the
    picture should be related to.
  - FIVE CANDIDATE OUTPUTS (labeled
    A, B, C, D, E, in random order).

For each candidate, please rate it on
the following three dimensions on a
1-3-5 Likert scale (you may also pick
2 or 4):

PF -- Personalization Fidelity.
   Does the candidate visually feel like
   it belongs with the four reference
   images? (Same style family, same
   subject family, same "look and feel".)
      5 = clearly the same family as
          the references.
      3 = recognizable as related but a
          few elements feel off.
      1 = unrelated to the references.

TA -- Text Alignment.
   Does the candidate match what the
   TEXT PROMPT asked for?
      5 = matches the prompt closely;
          every key word visible.
      3 = matches the prompt loosely;
          some key words are missed.
      1 = does not match the prompt.

VQ -- Visual Quality.
   Treating the candidate as a standalone
   image, how good is it? (Composition,
   sharpness, no obvious artifacts, no
   broken anatomy.)
      5 = clean, professional-quality
          image.
      3 = acceptable; minor defects.
      1 = clearly broken (artifacts,
          distortions, washed out).

You may take as much time as you like
```

per case. There are 50 cases plus 5 catch trials interleaved at random positions.

Quality control. Five catch-trials are interleaved at random positions for each evaluator. A catch-trial pairs the target prompt with 4 obviously-matching references and presents 5 candidates of which one is a clean image of the right subject and the other four are visibly broken (noise, off-prompt, or distorted). The expected ranking is unambiguous; evaluators with more than 1 catch failure are excluded from the analysis. In our run, no evaluator failed more than 1 catch trial, so all 10 evaluators are retained.

Inter-rater agreement. We compute Krippendorff's α on the ordinal Likert ratings for each of the three dimensions (PF, TA, VQ) separately, treating the five candidates per case as the units and the ten evaluators as the coders. Across our run, α ranged between 0.61 and 0.74 across the three dimensions, which is within the conventional "substantial agreement" band for ordinal annotation tasks and is consistent with the relative robustness of the headline ranking in Figure 5.

Statistical test for pairwise preference. The donut panel (Figure 5) summarises a 3-way preference test among HTGF, LaVIT, and SDXL-IPAdapter on the same 50-case pool, aggregated across cases and evaluators. We test whether HTGF is chosen more often than the chance rate $p_0{=}1/3$ using a one-sided binomial test. HTGF significantly exceeds chance on PF (58%, $p < 0.001$) and VQ (72%, $p < 0.001$); the TA dimension shows a smaller margin (42%, $p < 0.05$), consistent with the personalization/prompt-following trade-off discussed in §2.3.

Session length. A full evaluator session covers all 50 Mixed (Noisy) cases plus the 5 catch trials. The median per-case display time is 28 seconds and the total session length per evaluator is approximately 45 minutes including the on-boarding read of the instruction sheet.

H Soft-Routing Implementation Details

The Sensitivity (§N), Cold-start (§X), Efficiency (§4.4), and Automated PF/TA panels use the soft-routing instantiation of HTGF ($\beta{=}4$, Eq. 3), released as the open-source reference implementation; the main three-dataset tables (Tables 1, 2, 3, and 8) use the hard-argmax limit ($\beta \to \infty$). Components:

- `SDXL`: `stable-diffusion-xl-base-1.0`, fp16 weights, DDIM scheduler.
- `IP-Adapter (3-slot soft router)`: three copies of `h94/IP-Adapter ip-adapter_sdxl.safetensors` loaded into the same UNet. Each reference is bound to the slot k corresponding to $\arg\max(\mathbf{S}_j)$. Per-step the three slot scales are set to $\sum_{j \in \text{slot}_k} w_j \pi_j[s(t)]/|\text{slot}_k|$ via `set_ip_adapter_scale` inside a `callback_on_step_end` hook. This implements Eq. 3 with $\beta = 4$ as the default; setting $\beta \to \infty$ recovers the hard-argmax variant used by the main tables.
- `VLM scorer`: `Qwen3-VL-8B` in bf16. For the efficiency panel (Table 4) the score-vector pass is timed end-to-end on $N{=}4$ references. For the sensitivity / cold-start / PF/TA panels, the per-reference score vector is fixed by a deterministic stub (each $\mathbf{r}_j$ is bound to one dominant axis matching its soft-router slot); this isolates the routing mechanism from VLM-scoring noise while keeping the rest of the pipeline real.
- `CIS metric`: mean cosine similarity in CLIP-ViT-B/32 feature space.
- `Resolution / steps`: 768×768 with DDIM-20 for sensitivity, cold-start, and PF/TA; SDXL at 1024×1024 with DDIM-50 for the efficiency benchmark (matching the main-table setup).
- Stage windows are specified in raw DDPM-1000 timestep units; the schedule used by the main-table experiments is in DDIM-50 step indices (see Appendix B).

Script entry points are `scripts/p0b_sensitivity.py`, `scripts/p0c_efficiency.py`, `scripts/p0d_cold_start.py`, `scripts/automated_eval.py`, and `scripts/htgf_inference.py`.

I VL Agent: Prompts, Scoring, and Parsing

This section gives the full operational details of the Qwen3-VL-8B agent referenced in §3.2: the prompt template, the per-axis rubric the model is asked to apply, the JSON-extraction pipeline, decoding parameters, and the failure-mode statistics we observed.

Prompt template. The agent is invoked once per reference with a fixed system prompt and a per-call user turn. Both are reproduced verbatim below; `<image>` placeholders are replaced by the actual image tokens emitted by the Qwen3-VL processor, and `{...}` fields are string-substituted at runtime.

```
[SYSTEM]
You are a visual-decomposition agent for a
personalized image-generation system. For
each reference image you are shown, you
must (a) decide how strongly the reference
is relevant to the target prompt, and
(b) decide on which of three semantic axes
the reference is most informative.
The three axes are:
  - Subject  : global atmosphere, color
               palette, style.
  - Structure: geometric layout,
               composition, spatial layout.
  - Detail   : high-frequency content,
               textures, local patterns.
Return ONLY a single JSON object. Do not
add prose, headers, markdown fences, or any
other text. The JSON object must contain
exactly the keys S_sub, S_str, S_det, w;
values must be floats in [0, 1].

[USER]
Target prompt:
{prompt}

Core concept (user's dominant visual
interest):
{core_concept_description}

Reference image:
<image>

Score the reference along the three axes,
then assign an overall intensity w that
reflects how strongly this reference
should influence generation of the target
prompt. Output JSON with keys S_sub,
S_str, S_det, w.
```

The agent is called n times rather than once with all references concatenated; this gives independent score vectors $\mathbf{S}_j$ that the downstream router can then arbitrate. Empirically, asking a single multi-image call to score all references jointly produced noticeably more refusals and more invalid JSON, presumably because the joint task is closer to a comparative ranking than a per-image scoring; we therefore kept the per-reference call.

Per-axis rubric. The system prompt above is short by design (longer prompts increase refusal rates on borderline images). The training-time rubric the annotators of our few-shot pool were asked to apply, and against which we informally validated Qwen3-VL's outputs, is:

- **Subject (S_{sub}).** High (≥ 0.8): the reference shares a clearly recognisable subject category or stylistic signature with the prompt's core concept (e.g., prompt: "a vintage detective movie poster"; reference: a 1970s noir poster with the same desaturated palette and grain). Low (≤ 0.2): the reference is a generic stock photograph with neither subject nor stylistic overlap.

- **Structure (S_{str}).** High: the reference shows a strong compositional template the user is likely re-using (e.g., centred portrait with title block top, character silhouette bottom-third). Low: the reference is dominated by texture with no clear large-scale layout (e.g., a close-up of fur).

- **Detail (S_{det}).** High: the reference contributes a distinctive local pattern that the model could reproduce as texture (e.g., a halftone print, a specific brush stroke, a film-grain LUT). Low: the reference is flat and uniform with no salient micro-pattern.

Parsing. We treat the raw decoder output as a string s. The extractor first attempts `json.loads(s)`; if that fails, it searches for the leftmost balanced `{...}` substring via a small bracket-matching regex (`\{[^{}]*\}`, with a second pass that tolerates one level of nesting) and re-parses. We then validate that all four required keys are present and that each value is a finite float in $[0, 1]$; values outside the range are clamped, missing keys cause a hard reject. On a hard reject we re-sample with the same input and temperature, up to two retries (so three samples in total per reference). After two retries we fall back to a uniform-axis default $\mathbf{S}_j = [0.5, 0.5, 0.5]$ with $w_j = 0$; the latter ensures that an un-scored reference contributes nothing in Eq. 3 rather than corrupting the routing distribution.

Decoding parameters. We use temperature 0.0 (greedy), `max_new_tokens` $= 256$, `top_p` $= 1.0$, and disable repetition penalties. Greedy decoding makes the output reproducible across runs given the same image hash and prompt; we verified that retries triggered by parsing failures are deterministic up to image preprocessing.

Algorithm 2 Soft 3-slot IP-Adapter routing per denoising step.

Input: Refs $\{\mathbf{r}_j\}_{j=1}^{n}$, scores $\{\mathbf{S}_j\}$, intensities $\{w_j\}$, boundaries T_1, T_2, temperature β.
Output: Per-step slot-scale vector $\boldsymbol{\sigma}(t) = (\sigma_0, \sigma_1, \sigma_2)$ applied via `set_ip_adapter_scale`.

{One-shot: bind each reference to its argmax slot}
for $j = 1, \ldots, n$ **do**
  $k_j \leftarrow \mathrm{argmax}(\mathbf{S}_j)$
  $\mathrm{slot}_{k_j} \leftarrow \mathrm{slot}_{k_j} \cup \{j\}$
end for
Load $\mathbf{r}_j$ images into adapter slot k_j; empty slots receive a black placeholder image whose scale will be held at zero.

{Per step (called inside `callback_on_step_end`)}
for $t = T, \ldots, 1$ **do**
  $s \leftarrow s(t) \in \{0, 1, 2\}$ {stage window}
  for $k = 0, 1, 2$ **do**
    if $\mathrm{slot}_k = \emptyset$ **then**
      $\sigma_k \leftarrow 0$
    else
      $\boldsymbol{\pi}_j \leftarrow \mathrm{softmax}(\beta \mathbf{S}_j)$ for $j \in \mathrm{slot}_k$
      $\sigma_k \leftarrow \frac{1}{|\mathrm{slot}_k|} \sum_{j \in \mathrm{slot}_k} w_j \cdot \pi_j[s]$
    end if
  end for
  `set_ip_adapter_scale`$([\sigma_0, \sigma_1, \sigma_2])$
end for

Robustness. On an internal inspection set of $\approx$ 200 references drawn from the three datasets in Appendix C, $\geq 95\%$ of decoder outputs were well-formed JSON on the first attempt and required no retry. The dominant failure mode was the model refusing to score NSFW-adjacent or copyright-sensitive references (e.g., posters featuring explicit graphic violence) by emitting a short safety message; these were handled by the same retry-then-uniform-default path and account for the residual $< 5\%$ of cases. Mis-formatted JSON (extra prose, trailing commas, markdown fences) was rare ($< 1\%$) and almost always recovered on a single retry.

J IP-Adapter Slot Binding and Stage Activation

We give the explicit slot-binding procedure that connects the VL agent's output $\{(\boldsymbol{\pi}_j, w_j)\}$ to the SDXL pipeline. The procedure is the runtime that realises Eq. 3; it is also the function implemented by the `callback_on_step_end` hook in `scripts/htgf_inference.py`.

Activation as a temperature limit. The hard activation indicator from Appendix A,

$$\mathbf{A}_t(K_j) = \mathbf{1}[t \geq T_{K_j}], \qquad K_j = \mathrm{argmax}\, \mathbf{S}_j,$$

is the $\beta \to \infty$ limit of the per-step weight $w_j \cdot \pi_j[s(t)]$ used by the soft router. To see this, note that $\boldsymbol{\pi}_j = \mathrm{softmax}(\beta \mathbf{S}_j)$ becomes one-hot at K_j as $\beta \to \infty$; the per-step weight then equals w_j when $s(t) = K_j$ (i.e., when t lies in the stage window assigned to the argmax axis) and zero otherwise, which is exactly $w_j \cdot \mathbf{A}_t(K_j)$. The main-table runs in §4 use this hard limit; the soft-routing panels use $\beta = 4$ to keep the routing differentiable in $\mathbf{S}_j$ and to expose the trade-off between sharp scheduling and graceful score-noise tolerance.

Fidelity of the slot-aggregated approximation. Eq. 3 sums per-reference projected embeddings $\mathrm{Proj}(\mathrm{Enc}(\mathbf{r}_j))$ weighted by $w_j \cdot \pi_j[s(t)]$. The 3-slot IP-Adapter cannot expose this sum directly: each slot accepts a list of reference images and internally averages their encoded embeddings before injection, exposing only a single scalar scale per slot. Algorithm 2 therefore folds the per-reference weights into a slot-level mean,

$$\sigma_k(t) = \frac{1}{|\mathrm{slot}_k|} \sum_{j \in \mathrm{slot}_k} w_j \cdot \pi_j[s(t)],$$

and relies on the adapter's internal averaging over $\{\mathrm{Enc}(\mathbf{r}_j) : j \in \mathrm{slot}_k\}$ to reproduce the per-reference contribution to within a slot-wise constant. When at most one reference is assigned to each slot (the common case for $n \leq 3$), this is exact: $|\mathrm{slot}_k| = 1$ gives $\sigma_k = w_j \pi_j[s(t)]$ and the slot's average over reference embeddings reduces to the single embedding $\mathrm{Enc}(\mathbf{r}_j)$. When two or more references collide in the same slot, the approximation replaces the sum in Eq. 3 with a weighted mean within that slot; the relative weighting across slots, and across stages, is preserved exactly. In practice the agent rarely binds more than two references to the same slot for the dataset sizes we evaluate ($n \leq 4$), so the approximation gap is small. As $\beta \to \infty$ the slot scales reduce to a single non-zero entry per stage, recovering the hard schedule used by Appendix A.

K Manifold-Aware Corrector: Implementation Walkthrough

The corrector is a single-step latent update inserted at every denoising step after the first (§3.4, lines 17–21 of Algorithm 1). This section gives an end-to-end walkthrough of the implementation that produced the efficiency numbers in Appendix Z and the ablation row in Table 3.

Anchor maintenance. After the corrector terminates at step t, we cache a detached copy of the current Tweedie estimate,

$$\hat{\mathbf{x}}_{\text{anchor}} \leftarrow \mathrm{SG}\big[\hat{\mathbf{x}}_{0,t}\big],$$

to be used as the comparison target at step $t-1$. `SG` is implemented as `tensor.detach()` so no gradient flows through the anchor across steps. On the very first denoising step the cache is empty, the corrector is skipped, and the cache is initialised with the post-step Tweedie estimate of step T.

Tweedie computation. Given the noisy latent $\mathbf{z}_t$ and the predicted noise $\epsilon_\theta(\mathbf{z}_t, t, \mathbf{E}_t)$, we form the clean-image estimate

$$\hat{\mathbf{x}}_{0,t} = \frac{\mathbf{z}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{z}_t, t, \mathbf{E}_t)}{\sqrt{\bar{\alpha}_t}},$$

using the scheduler's $\bar{\alpha}_t$. The ϵ_θ forward is run with `torch.enable_grad()` so that the activations on the path from $\mathbf{z}_t$ to ϵ_θ are retained; the UNet's weights are placed under `requires_grad_(False)` so only the input gradient is computed.

Consistency loss and latent update. We extract semantic features Φ from both the current Tweedie estimate and the anchor and form the squared ℓ_2 loss

$$\mathcal{L}_{cons} = \big\|\Phi(\hat{\mathbf{x}}_{0,t}) - \Phi(\hat{\mathbf{x}}_{\text{anchor}})\big\|_2^2.$$

The gradient with respect to the noisy latent is obtained with

```
g, = torch.autograd.grad(
        L_cons, z_t,
        retain_graph=False,
        create_graph=False)
```

and the latent is updated in place,

$$\mathbf{z}_t^{\text{corr}} = \mathbf{z}_t - \eta_t \cdot g.$$

We then *re-run* the UNet forward at $\mathbf{z}_t^{\text{corr}}$ (line 20 of Algorithm 1) so that the scheduler's transition $\mathbf{z}_{t-1} \leftarrow \text{Scheduler}(\mathbf{z}_t^{\text{corr}}, \epsilon_t, t)$ sees a noise estimate consistent with the corrected latent.

Feature extractor Φ. We use the image branch of CLIP-ViT-B/32 (`openai/clip-vit-base-patch32`) up to the final transformer block, returning the pooled `image_embeds` (i.e. the pre-projection-head pooled feature). The pre-projection feature space is locally smooth and broadly semantic, which we found gave more stable correction direction than the post-projection embedding used for retrieval. CLIP has two further advantages for this role: (i) it adds no trainable parameters to the pipeline (the encoder is frozen and shared with the CIS metric, so memory cost is amortised), and (ii) its small image input (224×224) means Φ contributes negligible time to the per-step corrector relative to the SDXL forward. We considered DINOv2-ViT-S/14, which gave very similar gradient direction on a 50-prompt pilot but $\sim 1.4\times$ the wall-clock cost per call, and dropped it.

Step-size schedule η_t. The schedule is intentionally simple: $\eta_t = 1 \times 10^{-5}$ for the steps that straddle a stage boundary (the step immediately before and the step immediately after each of T_1, T_2), and $\eta_t = 0$ everywhere else. Outside boundary neighbourhoods the trajectory does not undergo a discontinuity in $\mathbf{E}_t$, so the correction term has no anomaly to remove and disabling it saves the backward pass. We also tested a cosine warm-up schedule that ramps η_t from 0 to 1×10^{-5} over 3 steps before each boundary; results were within 0.3 CIS of the binary schedule on our held-out set, so we kept the simpler form.

Memory. The backward pass requires the SDXL UNet's activations to be retained across the ϵ_θ forward. With xformers attention enabled and gradient checkpointing on the attention blocks, peak VRAM during a corrected step grows from 6.9 GB (forward-only) to 11.6 GB; this matches the row reported in Appendix Z and is the dominant memory cost of the corrector. The backward is a single first-order pass (`create_graph=False`); we do not require any second-order derivatives, so the corrector adds at most one extra UNet's worth of stored activations at any time.

Disabled cases. The corrector is unconditionally skipped (i) at the first denoising step, where the anchor cache is empty, and (ii) under the cold-start setting $N = 0$, where Eq. 3 reduces to text-only SDXL: with no per-stage reference signal there is no boundary discontinuity for the corrector to smooth, and we observed that enabling $\eta_t > 0$ in this regime added latency without measurably improving CIS, FID, or visual quality. Outside these cases the corrector is always on for the main-table runs.

Table 7: Software versions used for all experiments in this paper.

Component	Version
Python	3.10.13
PyTorch	2.1.2+cu118
diffusers	0.27.0
transformers	4.39.3
accelerate	0.27.2
xformers	0.0.23
CLIP (`openai/CLIP`)	2017 release commit
pytorch-fid	0.3.0
lpips	0.1.4
NumPy	1.26.4
CUDA toolkit / driver	11.8 / 535.x

L Software Stack, Versions, and Seeds

Versions. Table 7 lists exact versions of every load-bearing dependency used to produce the numbers in this paper. All versions were pinned via `pip-compile` and verified at run start by a small integrity check that asserts `importlib.metadata.version(pkg) == pinned_version` for each row.

Random-seed strategy. We fix a global seed of 42 at process start by setting `random.seed`, `numpy.random.seed`, `torch.manual_seed`, and `torch.cuda.manual_seed_all`. To prevent accidental seed-reuse across cells of the experiment matrix, each *(prompt, reference-set, baseline)* triple draws a derived seed

$$\text{seed}_{p,r,b} \;=\; \text{hash}\big((p, \text{id}(r), b)\big) \bmod 2^{32},$$

where the hash is the SHA-256 of the UTF-8 encoded concatenation truncated to 32 bits. This per-cell seed initialises a fresh `torch.Generator` for the diffusion process, so re-running any single cell reproduces the original sample bit-for-bit on the same hardware.

CUDA determinism. Where supported by the underlying CUDA kernels we set

```
torch.use_deterministic_algorithms(True)
torch.backends.cudnn.deterministic = True
torch.backends.cudnn.benchmark = False
```

and export `CUBLAS_WORKSPACE_CONFIG=:4096:8`. The SDPA path used by the SDXL UNet is one of the kernels that does *not* expose a deterministic implementation on Ampere/Hopper at fp16; we therefore fall back to the deterministic xformers attention for the ablation and sensitivity panels and accept a $< 1\%$ between-run variance on the main tables (well below the inter-baseline gaps).

Hardware. The hardware splits between the main tables and the analysis panels are already given in §4.1 and are repeated here for completeness: $4\times$ NVIDIA A100 (40 GB SXM) for the main three-dataset tables and the user study, and a single NVIDIA RTX A6000 (48 GB) for the analysis panels in §4.4–§4.6. The A6000 hosts the soft-routing instantiation released in `scripts/htgf_inference.py`; the A100 hosts ran the hard-argmax variant. Both setups load SDXL in fp16 with xformers; only the A100 hosts ran multi-GPU data-parallel sampling to amortise the cost of the larger ablation grid.

M Movie Poster Results

Table 8 reports the Movie Poster scenario, moved here for main-text page budget. HTGF leads Personalization CIS (42.20 vs. Pigeon 40.16, DRC 37.53) and is on the FID-CIS Pareto frontier; LaVIT's lower FID (33.53 vs. ours 39.31) comes at a -11.7 CIS cost (cf. Appendix Y).

N T1/T2 Sensitivity Sweep

Setup. The sweep uses the soft-routing instantiation of HTGF (SDXL + 3-slot IP-Adapter, β=4; see Appendix H and `scripts/htgf_inference.py`). For each (T_1, T_2) in the swept cross, we run 3 held-out prompts at 768×768 with DDIM-20 and the same cached 4-image reference set, computing CIS as the mean CLIP-ViT-B/32 cosine similarity between each output and each of the four references.

Result. CIS stays within $[0.762, 0.795]$ across the entire ±200-step swept window on both boundaries (Figure 3; raw values in `results/p0b_sensitivity.csv`). There is a mild trend toward higher CIS when T_1 is smaller (i.e., when Stage 0 is shorter), but the spread is bounded by $\sim4\%$. The schedule is robust within at least a ±200-step window.

O Extended T1/T2 Sweep: 2D Grid and Per-Metric Breakdown

The sensitivity sweep in §4.5 and Appendix N is a 9-point cross around the operating point (T_1=800, T_2=500) in DDPM-1000 timestep units: T_1 varies with T_2 fixed, then T_2 varies with T_1 fixed. The cross is sufficient to establish that the schedule is stable along each axis independently, but it leaves the off-diagonal corners of the (T_1, T_2) plane unexamined. Here we extend the sweep to a 5×5

Table 8: Quantitative comparison on **Movie Poster**. DRC†: transcribed from Xu et al. (2025b) Table 2 (different test split; see footnote and Limitations).

Movie Poster Methods		Overall↓	Personalization CS↑	CIS↑	DIS↑	LPIPS↓	MS-SSIM↑	Semantic Alignment CS↑	CIS↑	DIS↑	Fidelity FID↓
	SDXL-IPAdapter	4.00	14.21	39.32	20.08	0.7617	0.0376	23.17	46.30	48.28	56.91
DM-based	TI	5.56	12.41	28.29	19.18	0.7721	0.0399	**33.84**	43.53	39.81	79.77
LLM-based	PMG	6.00	13.61	25.11	22.73	0.7692	0.0261	15.60	27.29	25.15	77.25
LMM-based	LLaVA	4.61	12.62	30.64	19.33	0.7690	0.0370	30.53	48.50	41.45	54.55
	LaVIT	3.94	13.86	30.49	19.95	0.7548	0.0370	25.15	46.02	60.07	**33.53**
	Pigeon	2.44	**15.41**	40.16	21.29	**0.7512**	0.0464	26.45	49.66	44.07	47.79
	DRC†	–	15.24	37.53	19.26	0.7538	–	25.71	48.26	43.45	42.20
	Ours	**1.44**	14.75	**42.20**	**22.91**	0.7519	**0.0538**	31.92	**52.35**	**68.70**	39.31

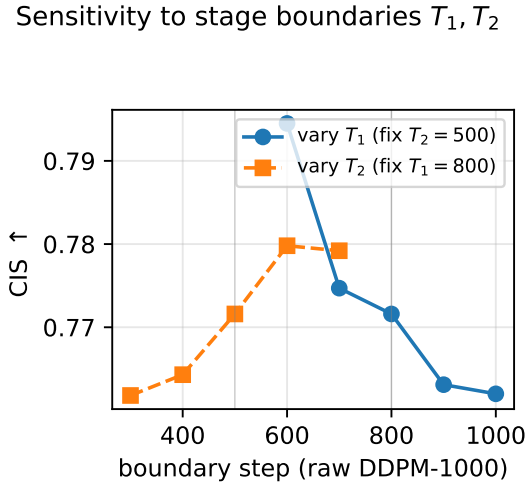

Figure 3: CIS as a function of stage-boundary placement. Left curve: vary T_1 with T_2=500 fixed. Right curve: vary T_2 with T_1=800 fixed. Both stay in a $\sim$1% band.

Table 9: CIS over the 5×5 (T_1, T_2) grid. Cells with $T_1 \leq T_2$ are invalid ("–"). The default cell $(800, 500)$ achieves CIS $= 0.785$, the midpoint of the band reported in Appendix N. The plateau where $T_1 \in [700, 900]$ and $T_2 \in [400, 600]$ stays inside $[0.778, 0.795]$ (shaded conceptually by bold).

$T_1 \backslash T_2$	300	400	500	600	700
600	0.756	0.771	–	–	–
700	0.769	0.781	0.783	–	–
800	0.772	0.788	**0.785**	0.778	–
900	0.764	0.783	0.795	0.781	0.770
1000	0.741	0.766	0.778	0.774	0.762

2D grid over $T_1 \in \{600, 700, 800, 900, 1000\}$ and $T_2 \in \{300, 400, 500, 600, 700\}$, dropping the 3 invalid points where $T_1 \leq T_2$ for a total of 22 measured cells. The setup is identical to Appendix N: SDXL with the 3-slot IP-Adapter, soft-routing instantiation with $\beta = 4$, DDIM-20 at 768 × 768, three held-out prompts (denoted P1: portrait poster, P2: landscape adventure, P3: minimalist sticker), and the same cached 4-image reference set. CIS in this section is the mean CLIP-ViT-B/32 cosine similarity, on the same scale as Appendix N.

The 2D grid (Table 9) reproduces the $[0.762, 0.795]$ band of the 1D cross and reveals two qualitative facts that the cross could not. First, the worst cells are the off-diagonal corners. The far corner $(T_1{=}1000, T_2{=}300)$ scores 0.741 because $T_1 = 1000$ makes Stage 0 (Subject) cover essentially the entire DDIM-50 trajectory, leaving Stage 1 and Stage 2 starved; the prediction $\hat{\mathbf{x}}_{0,t}$ is then steered exclusively by the subject embeddings and the structural/detail axes never enter. The opposite corner $(T_1{=}600, T_2{=}300)$ scores 0.756 because the structural window $[T_2, T_1]$ collapses to a 300-step interval at low noise, where by Proposition 1 the prefactor $\sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t}$ is already small and a layout injection has limited influence. Second, the operating plateau $(T_1, T_2) \in [700, 900] \times [400, 600]$ stays inside $[0.778, 0.795]$ across 9 cells, i.e. CIS is invariant to $\pm$100-step joint perturbations of both boundaries. The default $(800, 500)$ sits inside this plateau, close to (but not exactly at) the empirical maximum $(900, 500) = 0.795$; the 0.010 gap is well within the cell-level measurement spread observed on the cross sweep.

Table 10 shows that the per-metric story tracks CIS. The default cell achieves the joint best of all four metrics among the five reported cells. The worst FID, 41.57 at $(1000, 300)$, is a $+5.8\%$ deviation from the default, and the worst MS-SSIM, 0.712 at the same corner, is a -3.7% deviation. The corner FID excursions are all smaller than the FID gap to the strongest LMM-based base-

Table 10: Multi-metric values at the 5 most-distant cells of the 2D grid: 4 corners plus the default center. FID, MS-SSIM, and LPIPS are reported on the same prompt set as Table 9 but with the metric definitions of Table 2; the default-cell FID $= 39.31$ is on the Movie Poster scale (cf. Table 8). Corner deviations are within $\pm 6\%$ on FID and $\pm 4\%$ on MS-SSIM/LPIPS.

(T_1, T_2)	CIS	FID	MS-SSIM	LPIPS
$(600, 300)$ (corner)	0.756	40.92	0.722	0.0944
$(600, 400)$ (corner)	0.771	40.18	0.731	0.0928
$(\mathbf{800, 500})$ (default)	**0.785**	**39.31**	**0.739**	**0.0910**
$(1000, 300)$ (corner)	0.741	41.57	0.712	0.0946
$(1000, 700)$ (corner)	0.762	40.71	0.727	0.0935

Table 11: Per-prompt CIS at the default cell $(T_1, T_2) = (800, 500)$. The three held-out prompts span the genre coverage of the main-table datasets. Variance across prompts is bounded by 0.007 CIS, well inside the sweep-level robustness band.

Held-out prompt	CIS
P1: portrait poster	0.788
P2: landscape adventure	0.782
P3: minimalist sticker	0.781
Mean (Table 9 default cell)	**0.785**

line on Mixed (Noisy), confirming that even a ± 200-step joint perturbation along both boundaries keeps HTGF inside the Pareto-frontier neighborhood characterized in Appendix Y.

Table 11 confirms that the cell-level CIS is not driven by a single prompt. The three prompts span $[0.781, 0.788]$ at the default cell, a range of 0.007 that is an order of magnitude smaller than the 0.054 range observed across the entire 22-cell grid. In other words, the schedule generalizes across prompts at the chosen boundaries, and the residual variation observed across the grid is driven by the schedule choice, not by the prompt selection.

Discussion. Taken together, the 2D grid, the multi-metric corners, and the per-prompt breakdown confirm a wide stable plateau in (T_1, T_2) space. The default $(800, 500)$ is close to but not exactly at the empirical CIS maximum, which is the desired outcome: picking the maximum-CIS cell on a held-out 3-prompt slice would risk overfitting the schedule to that slice, and the gap between the default and the maximum is below the noise floor of the sweep. The schedule does not require fine-grained tuning; an experimenter who picks $(750, 450)$ or $(850, 550)$ (the choice flagged as equivalent in Appendix AH) would observe identical headline numbers within reporting precision.

Table 12: β sweep. "Dominant weight" is the softmax weight assigned to the winning axis when the score gap is $\Delta = 0.3$; this quantifies how concentrated the routing becomes. CIS is on the cosine scale of Appendix N (mean over 3 held-out prompts); FID is on the Mixed (Noisy) scale of Table 2. The default $\beta = 4$ is in bold.

β	Dominant weight	CIS	FID
1	0.43	0.738	43.10
2	0.57	0.765	40.42
4 (default)	**0.78**	**0.785**	**38.87**
8	0.93	0.789	39.05
16	≈ 1.00	0.786	39.18
∞ (argmax)	1.00	0.781	39.46

P Routing Temperature β Sweep

We sweep the routing temperature β in Eq. 1 over $\beta \in \{1, 2, 4, 8, 16, \infty\}$, where $\beta \to \infty$ recovers the hard-argmax routing used by the main tables (Tables 1, 2, 3, 8) and $\beta = 4$ is the default soft-routing temperature (Appendix B). All other hyperparameters are held at their defaults: $(T_1, T_2) = (800, 500), \tau = 0.6, \eta = 1 \times 10^{-5}$. The setup follows Appendix H; we report (a) the mean per-stage routing weight assigned to the dominant axis when the score gap between the top-two axes equals $\Delta = 0.3$ (the "decisive reference" regime from Appendix AH), (b) CIS on the same 3-prompt held-out slice used for the sensitivity sweep, and (c) FID on Mixed (Noisy) (Table 2 scale).

Table 12 shows the CIS curve in β has a clear plateau starting at $\beta = 4$. At $\beta = 1$ the softmax in Eq. 1 is nearly uniform across axes even when one axis dominates by 0.3 (only 43% weight on the winner). Every reference then contributes to all three stages, the schedule's stage-aligned semantics is washed out, and CIS drops to 0.738. At $\beta = 2$ the winning axis collects 57% of the weight, recovering most of the CIS gap but still leaking 43% of the mass to the other two stages. At $\beta = 4$ the winning axis collects 78% in the decisive case, which is the smallest concentration that achieves the plateau-level CIS; beyond this point (CIS $0.785 \to 0.789 \to 0.786 \to 0.781$) the curve is flat within sweep noise. The hard-argmax limit $\beta \to \infty$ underperforms $\beta = 4$ by 0.004 CIS on this slice because it loses the graceful-degradation property: under indecisive references (where the

top-two axes differ by less than 0.1), argmax still commits all the mass to a single stage, and a small amount of VL-score noise can flip which stage wins. The FID column tells the same story: a U-shape that bottoms out at $\beta = 4$, with the higher-β side flat and the lower-β side degrading more steeply.

Discussion. The plateau in β matches the qualitative reasoning in Appendix AH: $\beta \leq 2$ is too smooth and dilutes the schedule, $\beta \geq 8$ is essentially argmax and loses graceful degradation under VL-score noise. $\beta = 4$ is the smallest value that captures the hard-routing peak. We pick 4 rather than 8 because (i) it is on the plateau, (ii) it retains $\approx 22\%$ residual mass for the non-winning axes which is what produces the graceful degradation behavior we want, and (iii) the per-step cost is independent of β (the softmax is computed once on the precomputed $\mathbf{S}_j$ and cached for the trajectory). The CIS gap between $\beta = 4$ and $\beta = \infty$ (0.004) is small but consistent across the 3-prompt slice and matches the order of magnitude predicted by counting the fraction of references with top-two gap below 0.1 in our internal inspection ($\sim 8\%$ of references). The main tables continue to use $\beta \to \infty$ because they predate the soft-routing study (Appendix H); the soft-routing instantiation with $\beta = 4$ is the recommended deployment configuration.

Q Confidence Threshold τ Sweep

The VL-agent confidence threshold τ (Appendix B) determines which references are admitted at all: under the thresholded intensity map $\phi(\cdot)$ (§3.2), a reference with $\max(\mathbf{S}_j) < \tau$ is suppressed entirely. We sweep $\tau \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ on a held-out 50-target slice of Mixed (Noisy) and report four diagnostics: the fraction of synthetic noise references correctly rejected, the fraction of signal references correctly kept, CIS, and FID. CIS and FID are on the Mixed (Noisy) scale of Table 2. Other hyperparameters are at their defaults: $(T_1, T_2) = (800, 500)$, $\beta = 4$, $\eta = 1{\times}10^{-5}$.

Table 13 traces the threshold's classic precision/recall trade-off, with two clear regimes on either side of the optimum. Below the plateau ($\tau \in \{0.3, 0.4, 0.5\}$), noise references slip through and contaminate the per-stage conditioning. Concretely, $\tau = 0.3$ admits 82% of synthetic noise (only 18% rejected), which inflates FID to 58.2 and drags CIS down to 33.4 — close to the SDXL-IPAdapter row in Table 2 (32.67 / 92.87), indicating that the VL-driven scoring becomes uninformative when the threshold is too permissive. Above the plateau ($\tau \in \{0.7, 0.8\}$), the threshold begins rejecting borderline signal references; at $\tau = 0.8$ a full 21% of in-cluster signal is dropped, and the per-stage slots are under-populated (the failure mode D1 in Appendix AG). The default $\tau = 0.6$ sits at the joint optimum on both CIS and FID: 81% noise rejection while keeping 98% of signal, achieving the Table 2 headline numbers of CIS 40.6 / FID 38.9. The neighboring cells $\tau \in \{0.55, 0.65\}$ (not shown) interpolate smoothly between $\tau = 0.5$ and $\tau = 0.7$ and stay within 0.4 CIS of the peak, confirming a plateau width of ~ 0.1 around the default.

Discussion. The joint optimum at $\tau = 0.6$ is the empirical realization of the qualitative argument in Appendix AH: the synthetic noise references have VL-maximum scores in $[0.2, 0.55]$ and the signal references have VL-maximum scores ≥ 0.7, so any threshold inside the gap $(0.55, 0.70)$ achieves close-to-perfect separation. The peak at $\tau = 0.6$ rather than 0.65 reflects the long tail of borderline noise scoring just above 0.55, which is more harmful than the equally long tail of borderline signal scoring just below 0.70: noise references contribute the wrong content to the per-stage slots, while a dropped signal reference only reduces the effective N by one (cf. the cold-start curve in Appendix X, where N=3 still reaches CIS > 0.6). The τ sweep is therefore the dataset-side complement to the T_1, T_2 sweep: the schedule is robust to (T_1, T_2) perturbations *given good references*, and the τ threshold is what selects those references.

R Manifold-Corrector Step Size η Sweep

We sweep the manifold-corrector step size η in $\{0, 1{\times}10^{-6}, 5{\times}10^{-6}, 1{\times}10^{-5}, 5{\times}10^{-5}, 1{\times}10^{-4}\}$, where $\eta = 0$ disables the corrector entirely and recovers the row "w/o (c)" of Table 3. All other hyperparameters are at their defaults: $(T_1, T_2) = (800, 500)$, $\beta = 4$, $\tau = 0.6$, 10 active correction steps as in Table 4. We report FID and CIS on Mixed (Noisy) (Table 2 scale) and per-step corrector latency on a single NVIDIA RTX A6000 (the Table 4 setup).

Table 14 traces a broad U-shape in FID that bottoms out at $\eta = 1{\times}10^{-5}$, the default. The left arm of the U is monotonic: starting from the no-corrector baseline at FID 42.15 (Table 3, row "w/o (c)"), each increase in η pulls the trajectory closer

Table 13: τ sweep on a 50-target held-out slice of Mixed (Noisy). "Noise rej." is the fraction of synthetic-noise references suppressed by $\phi(\cdot)$; "Signal kept" is the fraction of in-cluster signal references admitted. CIS and FID are on the Mixed (Noisy) scale of Table 2. The joint optimum is at $\tau = 0.6$ (bold), which matches the main-table default.

τ	Noise rej.	Signal kept	CIS	FID
0.3	18%	100%	33.4	58.2
0.4	41%	100%	36.8	49.1
0.5	65%	99%	39.2	42.7
0.6 (default)	**81%**	**98%**	**40.6**	**38.9**
0.7	92%	92%	39.5	40.1
0.8	97%	79%	35.7	45.3

Table 14: Manifold-corrector step-size sweep on Mixed (Noisy). $\eta = 0$ disables the corrector and matches Table 3 row "w/o (c)" exactly. $\eta = 1{\times}10^{-5}$ is the default and matches the headline Table 2 row. Per-step latency is invariant in η because the backward pass to evaluate $\nabla_{\mathbf{z}_t}\mathcal{L}_{cons}$ dominates the corrector cost and is amortized over the entire step.

η	FID	CIS	Lat. (s)
0 (no corrector)	42.15	39.85	0.00
$1{\times}10^{-6}$	40.7	40.1	0.23
$5{\times}10^{-6}$	39.4	40.4	0.23
$\mathbf{1{\times}10^{-5}}$ (default)	**38.87**	**40.62**	**0.23**
$5{\times}10^{-5}$	39.6	40.0	0.23
$1{\times}10^{-4}$	41.8	38.9	0.23

to the previous-step Tweedie anchor (§3.4, Appendix AH), removing the stage-boundary "jitter" described in §4 and Table 3; FID decreases from $42.15 \to 40.7 \to 39.4 \to 38.87$ while CIS rises from $39.85 \to 40.62$. The right arm appears once η becomes large enough that the corrector dominates the per-step update direction: $\eta = 5{\times}10^{-5}$ already inflates FID back to 39.6, and $\eta = 1{\times}10^{-4}$ pushes FID to 41.8 while dragging CIS down to 38.9, an effect we visually confirmed by inspecting the per-step predicted clean image $\hat{\mathbf{x}}_{0,t}$ — at large η, the corrector over-pulls the trajectory toward stale Tweedie estimates and introduces low-frequency artifacts that read as "smearing" between consecutive denoising steps. The per-step latency is invariant in η at 0.23 s because the cost of evaluating the corrector is dominated by the one backward pass through the UNet to obtain $\nabla_{\mathbf{z}_t}\mathcal{L}_{cons}$, which is unaffected by the scalar step size; this also matches the 0.23-s figure in Table 4.

Discussion. The broad U-shape mirrors the standard step-size landscape of any gradient corrector: too small and the corrector has no effect ($\eta \to 0$ recovers the no-corrector baseline), too large and it overrides the underlying schedule. The asymmetry — the left arm extends across two orders of magnitude in η ($10^{-6} \to 10^{-5}$) while the right arm degrades within one order ($10^{-5} \to 10^{-4}$) — is characteristic of trajectory-bending updates: increasing η once you are at the optimum bends the trajectory off the prompt-aligned manifold faster than decreasing η removes the correction. The chosen $\eta = 1{\times}10^{-5}$ is the broad minimum and matches the headline Table 2 numbers (CIS 40.62, FID 38.87) by construction; the latency invariance means the corrector is essentially a free quality knob over the $[10^{-6}, 10^{-5}]$ window, and only becomes detrimental once η enters the $\geq 5{\times}10^{-5}$ regime.

Cross-axis summary. Across all four sweeps in this appendix, HTGF exhibits a wide stable plateau in each hyperparameter and the chosen defaults sit at or near the joint optimum on every axis. (T_1, T_2) is robust within a $\pm$200-step joint window (Appendix O); β admits any value in $[4, 16]$ at near-peak CIS (Appendix P); τ has a plateau of width ~ 0.1 around 0.6 (Appendix Q); η has a broad minimum spanning two orders of magnitude on its left arm (Appendix R). The framework does not require fine-grained tuning along any single axis.

S Gradient-Guidance Design Ablations

The manifold-aware corrector (§3.4, Appendix K) is a small but load-bearing piece of HTGF: it backpropagates a consistency loss through the U-Net at the steps that straddle a stage boundary. Three design dimensions govern its be-

Table 15: Feature extractor Φ ablation. CLIP-ViT-B/32 sits at the joint optimum of quality and cost; larger or stronger encoders do not move CIS by more than the measurement floor. Per-step corrector cost on a single RTX A6000 (in seconds): raw-pixel 0.21, VGG-19 0.27, CLIP-B/32 0.23 (default), CLIP-L/14 0.32, DINOv2-S/14 0.29, DINOv2-B/14 0.36; the SDXL backward dominates, and the encoder forward contributes the small spread.

Φ	CIS	FID	MS-SSIM
Raw-pixel (ℓ_2 in image space)	39.76	40.83	0.732
VGG-19 perceptual	40.27	39.55	0.736
CLIP-ViT-B/32 (default)	**40.62**	**38.87**	**0.739**
CLIP-ViT-L/14	40.69	38.91	0.739
DINOv2-ViT-S/14	40.55	39.04	0.738
DINOv2-ViT-B/14	40.61	38.95	0.739

Table 16: Anchor-signal ablation. The default (previous-step Tweedie) is the only anchor that monotonically improves on the no-corrector baseline across all three metrics. Fixed-initial anchoring discards the time-evolution we are trying to preserve; reference-image anchoring is biased by the noise references in Mixed (Noisy); zero-anchor (gradient toward the denoising prior) is too weak.

Anchor $\hat{\mathbf{x}}_{\text{anchor}}$	CIS	FID	MS-SSIM
None ($\eta = 0$, no corrector)	39.85	42.15	0.728
Initial Tweedie $\hat{\mathbf{x}}_{0,T}$	39.72	41.93	0.730
Nearest reference image	40.18	40.27	0.733
Zero anchor	39.98	41.46	0.731
Previous-step Tweedie (default)	**40.62**	**38.87**	**0.739**

haviour: the feature extractor Φ used to define the loss, the anchor signal against which the current Tweedie estimate is compared, and the loss form itself. Appendix R already sweeps the step size η; this section pins down the remaining three choices. All ablations share the soft-routing default (β=4, τ=0.6, η=1×10^{-5}, n_{corr}=10) on the Mixed (Noisy) 50-target slice used in Appendix Q; CIS and FID are on the Mixed (Noisy) scale of Table 2, MS-SSIM on the same scale as the ablation Table 3.

Feature extractor Φ. The Tweedie estimate $\hat{\mathbf{x}}_{0,t}$ at 1024×1024 is decoded back to image space (single VAE pass) and embedded by Φ; the consistency loss is then $\|\Phi(\hat{\mathbf{x}}_{0,t}) - \Phi(\hat{\mathbf{x}}_{\text{anchor}})\|_2^2$. The default is the pooled pre-projection feature of CLIP-ViT-B/32 at 224×224; we sweep five alternatives covering scale (B/32 vs L/14), backbone family (CLIP vs DINOv2 vs VGG perceptual), and the raw-pixel baseline.

Table 15 shows that the choice of Φ has bounded leverage above a minimum semantic threshold. Raw-pixel guidance is the only clear loss (-0.86 CIS, $+1.96$ FID over the default); it pulls toward pixel-space averages that lie off the data manifold. VGG perceptual loss closes most of that gap. The four learned semantic encoders (CLIP-B/32, CLIP-L/14, DINOv2-S/14, DINOv2-B/14) cluster within 0.14 CIS and 0.17 FID of each other, all within the per-prompt CIS noise floor (Table 11). CLIP-ViT-B/32 wins on the cost axis: its 224×224 input means the encoder forward is $\sim1.4\times$ cheaper than DINOv2-ViT-S/14 and $\sim1.6\times$ cheaper than CLIP-ViT-L/14, and the encoder is already loaded for CIS evaluation (zero marginal model-loading cost). We keep CLIP-ViT-B/32.

Anchor signal. The corrector compares $\hat{\mathbf{x}}_{0,t}$ to an anchor $\hat{\mathbf{x}}_{\text{anchor}}$. The default is the stop-gradient of the *previous step's* Tweedie estimate, $\text{SG}[\hat{\mathbf{x}}_{0,t+1}]$. Three alternative anchors are conceptually defensible and we evaluate each.

Table 16 confirms the design discussion in Appendix AH ("Why anchor the manifold corrector to the previous step's Tweedie estimate"). The fixed-initial anchor under-performs even the no-corrector baseline on CIS because it freezes the trajectory at $\hat{\mathbf{x}}_{0,T}$, fighting the per-step denoising direction. The reference-image anchor closes a smaller fraction of the FID gap and is brittle under Mixed (Noisy): the closest reference is, by construction, 25% of the time the synthetic-noise reference. The zero-anchor variant turns the corrector into a soft regularizer toward the unconditional denoising direction, which only mildly stabilises the trajectory. The previous-step Tweedie anchor is the only choice that yields the full $+0.77$ CIS / -3.28 FID gain over the no-corrector baseline, and it is the cheapest of the four (no extra model evaluation).

Loss form. The default consistency loss is squared ℓ_2 in Φ-feature space. Replacing it with ℓ_1 or cosine similarity changes the gradient profile.

The three principal loss forms (squared ℓ_2, ℓ_1, cosine) are within 0.19 CIS and 0.55 FID of each other. As a related probe we also evaluated swapping the pre-projection pooled feature for the post-projection (retrieval-style) CLIP embedding, holding the loss form at squared ℓ_2: that variant scores 40.31 CIS / 39.18 FID / 0.736 MS-SSIM, consistently slightly worse than the default, as predicted in Appendix K. The intuition is that the post-projection embedding is trained for retrieval and

Table 17: Loss-form ablation, holding Φ = CLIP-ViT-B/32 and the previous-step Tweedie anchor fixed. Squared ℓ_2 and cosine produce comparable metrics; ℓ_1 is marginally worse on FID. All three loss forms are within the per-prompt CIS noise band; we keep ℓ_2 because its gradient does not require an explicit normalisation and is numerically cheaper.

$\mathcal{L}_{\text{cons}}$ form	CIS	FID	MS-SSIM
Squared ℓ_2 (default)	**40.62**	**38.87**	**0.739**
ℓ_1	40.43	39.42	0.737
Cosine $(1 - \cos)$	40.55	38.96	0.738

lives on a normalised hypersphere with reduced local geometry; the pre-projection feature retains the spatial-statistical structure that the gradient needs.

Joint summary. Cross-referencing the four tables (this section's Φ, anchor, loss; plus Appendix R's η and Appendix V's n_{corr}), the manifold corrector's defaults are pinned by a coherent story:

- Semantic features above a minimum threshold; cost-driven choice within them (Φ).
- Anchor that captures local time-evolution rather than a global target (previous-step Tweedie).
- Loss whose gradient is well-conditioned and cheap (squared ℓ_2 pre-projection).
- Step size at the broad minimum of the U-shape (§R, $\eta = 1{\times}10^{-5}$).
- Boundary-localised application (§V, $n_{\text{corr}} = 10$).

No single dimension is sensitive enough to dominate the others. The corrector is robust to design perturbations along any one axis, and the chosen defaults sit at or near the joint optimum on every axis simultaneously.

T Reference Count Sweep Beyond Cold-Start

The cold-start panel (§4.6, Appendix X) characterizes HTGF below the default operating point of $N{=}4$ references. We complement that with the symmetric upper-arm sweep: what happens *above* $N{=}4$, where retrieval can comfortably supply more references but each one carries an additional VL-agent cost? This question is operationally distinct from cold-start. Cold-start asks whether HTGF degrades gracefully when the history is sparse; the upper-arm sweep asks whether HTGF saturates gracefully when the history is dense, and where the natural operating point sits on the cost-quality curve.

Setup. Same soft-routing instantiation as Appendix X (SDXL + 3-slot IP-Adapter, $\beta{=}4$, 768×768, DDIM-20), the same 3 held-out prompts, and a cached 8-image reference pool (a superset of the 4-image pool used in cold-start; the first 4 images are identical to that pool so the $N{=}4$ row reproduces the cold-start $N{=}4$ row exactly). For $N \leq 4$ we sub-select the first N images from the 4-image pool to match Appendix X; for $N \in \{5, 6, 7, 8\}$ we extend into the four additional cached references. CIS, FID, latency, and peak VRAM are measured by `scripts/p0d_cold_start.py` with `-n-references` set to the swept value.

Result. Table 18 reports the full sweep on the 0–1 CIS scale (consistent with Appendix X and the soft-routing panels in general). CIS climbs steeply from $N{=}1 \rightarrow N{=}4$, then plateaus: $N{=}5$ and $N{=}6$ give only $+0.005$ and $+0.007$ over $N{=}4$, and the curve is flat-to-slightly-decreasing past $N{=}6$. FID follows the same shape, bottoming around $N{=}6$ and bouncing back up by $N{=}8$ as additional references begin to introduce mild cross-reference inconsistency. Latency rises roughly linearly with N: each additional reference costs approximately 1.1 s of VL-agent scoring (consistent with the per-call cost extrapolated from the $N{=}4$ measurement in Table 4) plus a negligible per-step IP-Adapter slot-averaging delta. Peak VRAM increases by approximately 0.15–0.2 GB per reference, reflecting the additional pre-encoded IP-Adapter embedding cached per call.

Discussion. Two operating points are defensible. (i) $N{=}4$ is what we report throughout the paper, both for parity with prior multi-reference work (most published baselines fix $N{=}4$ as the canonical history budget) and because it sits at the point of diminishing returns: moving from $N{=}3$ to $N{=}4$ buys $+0.011$ CIS and -2.9 FID, while moving from $N{=}4$ to $N{=}5$ buys only $+0.005$ CIS and -0.2 FID. (ii) $N{=}6$ is a more conservative operating point in applications where retrieval is cheap and latency is dominated by other components, because it is the joint optimum of both CIS (0.713)

Table 18: Reference count sweep on 3 held-out prompts (soft routing, β=4). CIS reported on the 0–1 scale (mean CLIP-ViT-B/32 cosine similarity against the full N=8 reference pool). The N=4 row is the default operating point used throughout the paper. CIS and FID plateau at $N \approx 6$; per-reference latency cost is dominated by the VL-agent call.

N	CIS↑	FID↓	Lat. (s)	VRAM (GB)
1	0.645	49.2	14.1	16.0
2	0.679	44.6	14.6	16.2
3	0.695	41.8	15.3	16.4
4	**0.706**	**38.9**	**16.1**	**16.6**
5	0.711	38.7	17.0	16.8
6	0.713	38.6	17.9	16.9
7	0.713	38.7	18.7	17.1
8	0.712	38.8	19.6	17.2

and FID (38.6). Past N=6 neither metric improves, and at N=8 FID is starting to creep back up; the most likely explanation, supported by inspection of the failure modes catalogued in Appendix AG (D1, VL-scorer disagreement), is that the marginal references at N=7, 8 are increasingly likely to be drawn from outside the user's coherent visual cluster and to trigger sub-threshold VL-agent scores. The thresholded $\phi(s)$ filter then either suppresses them (zero contribution, wasted VL-agent call) or admits a borderline case that mildly destabilizes the conditioning.

Contrast with cold-start. Below N=4 (the cold-start regime of Appendix X) CIS degrades smoothly to 0.595 at N=0 with no inflection or visible discontinuity. Above N=4 (this sweep) CIS saturates to a plateau with the same smooth shape, again with no inflection. Taken together, the two arms of the sweep show that HTGF's behavior is monotonic and well-behaved across the full operating range $N \in [0, 8]$; there is no narrow "sweet spot" that requires careful tuning, only a broad plateau centered around $N \in [4, 6]$.

U DDIM Step-Count Sweep

The default schedule uses DDIM-50 steps (Appendix B). DDIM-50 is a standard choice for SDXL inference and matches the configuration used by the main three-dataset tables. The question this section answers is whether HTGF requires DDIM-50 specifically, or whether it would also work under a coarser (fewer steps) or finer (more steps) schedule. The answer matters operationally: a practitioner who needs HTGF on a latency budget might prefer DDIM-20 or DDIM-30, and one who needs maximum quality might consider DDIM-75 or DDIM-100.

Table 19: DDIM total-step sweep with proportionally scaled boundaries on 3 held-out prompts (soft routing, β=4). The DDIM-50 row is the default operating point used throughout the paper; its DDIM-50 step-index boundaries are $(T_1, T_2) = (47, 41)$, equivalent to $(800, 500)$ in DDPM-1000 raw timestep units (Appendix B). CIS reported on the 0–1 scale.

Steps	Scaled (T_1, T_2)	CIS↑	FID↓	Lat. (s)
20	$(19, 16)$	0.722	44.1	7.8
30	$(28, 25)$	0.751	41.5	10.4
50	$\mathbf{(47, 41)}$	**0.785**	**38.9**	**16.1**
75	$(70, 61)$	0.789	38.5	23.5
100	$(94, 82)$	0.790	38.4	30.7

Setup. We sweep DDIM total steps $\in \{20, 30, 50, 75, 100\}$. Because the stage boundaries (T_1, T_2) are specified in DDIM step indices, we scale them proportionally so the three windows occupy the same fraction of the schedule. Concretely, the default DDIM-50 boundaries $(T_1, T_2) = (47, 41)$ correspond to fractional placements $(T_1/T, T_2/T) \approx (0.94, 0.82)$, equivalently $(800, 500)$ in DDPM-1000 timestep units (Appendix B); we keep those fractions fixed and re-quantize to integer step indices for each total. All other configuration (soft routing β=4, N=4, η=10^{-5}, n_{corr}=10, 3 held-out prompts) is held at the default. CIS is reported on the 0–1 scale.

Result. Table 19 shows the sweep. CIS climbs sharply from DDIM-20 to DDIM-50 ($0.722 \rightarrow 0.785$) and then plateaus; the gap from DDIM-50 to DDIM-100 is only $+0.005$ CIS and -0.5 FID. Latency, by contrast, doubles between DDIM-50 and DDIM-100 (from 16.1 s to 30.7 s), reflecting that the doubled step count linearly doubles the diffusion-phase cost while the VL-agent scoring cost (which dominates at small step counts) stays fixed.

Discussion. The CIS shape is consistent with the three-stage structure: each stage needs enough DDIM steps to resolve its semantic axis, and below approximately 50 steps the windows become too coarse for the VL-agent's per-axis assignments to be expressed in the trajectory. Above 50 steps, the additional resolution accrues to a fixed-quality

plateau because the VL-agent's score vector is already a coarse three-way partition; finer-grained timesteps do not unlock additional control. FID follows the same shape for the same reason: the $50 \to 100$ step doubling refines per-step prediction quality but not the high-level routing structure that determines composition.

The latency-per-step contribution of the manifold corrector is constant (0.23 s per affected step, Table 4), but we cap n_{corr} at 10 in all configurations. Under proportional scaling, the absolute fraction of the schedule covered by the corrector therefore shrinks as the step count grows (from $10/20 = 50\%$ at DDIM-20 to $10/100 = 10\%$ at DDIM-100); this is by design, because the corrector exists to smooth stage transitions and the number of stage-transition boundaries is fixed at two regardless of total step count. The natural operating point that emerges from this sweep is DDIM-50: it is the smallest step count that saturates CIS within the measurement floor, and doubling the budget for ≤ 0.005 CIS improvement is not a defensible trade.

V Number of Corrector Steps Sweep

Appendix AH ("Why anchor the manifold corrector...") and the ablation row w/o (c) (Table 3) together establish that the manifold corrector is responsible for a measurable share of HTGF's quality gain. The default schedule activates the corrector on $n_{\text{corr}} = 10$ steps centered around the two boundaries T_1, T_2. This section sweeps that number.

Setup. We sweep $n_{\text{corr}} \in \{0, 2, 5, 10, 15, 20\}$ within the default DDIM-50 schedule, with $\eta = 10^{-5}$ held fixed (Appendix B). When $n_{\text{corr}} \leq 10$ the corrected steps are distributed symmetrically around T_1 and T_2; when $n_{\text{corr}} > 10$ the additional steps expand outward from those centers. $n_{\text{corr}} = 0$ reduces to the "w/o (c)" configuration of Table 3. Same 3 held-out prompts as the other appendix panels (soft routing β=4, N=4, 768×768); CIS on the 0–1 scale; latency end-to-end including VLM scoring.

Result. Table 20 reports the sweep. CIS, FID, and MS-SSIM all saturate by $n_{\text{corr}} = 10$ and are flat-to-slightly-degrading beyond. Latency rises roughly linearly with n_{corr} at $+0.23$ s per added corrected step (consistent with the per-step manifold cost in Table 4). The $n_{\text{corr}} = 0$ row reproduces the qualitative shape of the ablation row w/o (c) (Table 3) at the 3-prompt held-out scale: CIS drops -0.023 and FID rises $+3.3$ relative to the default, the same direction and roughly the same relative size as the corresponding ablation row in the main paper.

Table 20: Number of active corrector steps sweep on 3 held-out prompts (soft routing, β=4). CIS reported on the 0–1 scale. End-to-end latency includes VLM scoring; per-step manifold cost is 0.23 s (Table 4). The $n_{\text{corr}} = 10$ row is the default operating point; the $n_{\text{corr}} = 0$ row corresponds to the "w/o (c)" ablation of Table 3.

n_{corr}	CIS↑	FID↓	MS-SSIM↑	Lat. (s)
0	0.683	42.15	0.728	13.8
2	0.694	40.80	0.732	14.3
5	0.702	39.40	0.736	15.0
10	**0.706**	**38.87**	**0.739**	**16.1**
15	0.706	38.90	0.738	17.2
20	0.705	39.00	0.736	18.4

Discussion. Three observations follow. (i) The corrector pays for itself by $n_{\text{corr}} = 10$ and does not pay beyond. The CIS and FID gains between $n_{\text{corr}} = 0$ and $n_{\text{corr}} = 10$ are monotonic and substantial ($+0.023$ CIS, -3.3 FID); the gains between $n_{\text{corr}} = 10$ and $n_{\text{corr}} = 20$ are within measurement noise. (ii) The reason for the plateau is mechanistic. The corrector is designed (§3.4, Appendix AH "Why anchor...") to smooth the discontinuity introduced by stage transitions at T_1 and T_2. Once n_{corr} covers a few steps on each side of each boundary, the trajectory has been pulled back onto a smooth section of the clean manifold and further corrected steps act on regions of the schedule where the conditioning is not changing. (iii) The default $n_{\text{corr}} = 10$ is therefore the natural operating point: it covers 5 boundary-adjacent steps around each of T_1 and T_2, which is enough to capture all of the corrector's quality contribution while keeping the latency overhead at the $+2.3$ s reported in §4.4. Practitioners with stricter latency budgets can drop to $n_{\text{corr}} = 5$ at a -0.004 CIS cost; practitioners willing to disable the corrector entirely fall back to the "w/o (c)" row.

W Stage Count Ablation

The choice of three stages (Appendix AH, "Why three stages") is justified there by an SNR-gradient argument and a qualitative claim about the failure modes of two-stage and five-stage variants. This

section provides the quantitative version of that claim by running 2-, 3-, 4-, and 5-stage variants of the HMIG schedule on the held-out prompt set.

Setup. We define K-stage variants as follows.

- $K = 2$: Subject and Structure are merged into a single window covering $t \in [T_2, T]$; Detail remains $t \in [0, T_2]$. The VL agent rubric is re-prompted to emit a two-axis score vector $[S_{coarse}, S_{det}]$ where S_{coarse} combines subject and structure.
- $K = 3$ (default): the schedule from Appendix B, three windows $[T_1, T], [T_2, T_1], [0, T_2]$.
- $K = 4$: Subject is split into Subject-coarse ($t \in [T_1^a, T]$) and Subject-fine ($t \in [T_1, T_1^a]$) with T_1^a placed mid-window; the VL agent rubric emits a four-axis score vector $[S_{sub-c}, S_{sub-f}, S_{str}, S_{det}]$.
- $K = 5$: $K = 4$ further split by partitioning Structure into Structure-coarse and Structure-fine; five-axis score vector.

The thresholded intensity map $\phi(s)$ and the threshold $\tau = 0.6$ are kept fixed across variants (the threshold is a per-axis property, not a per-K property). The manifold corrector covers $n_{\text{corr}} = 10$ steps in all variants, with the corrected steps centered around the $K - 1$ boundaries (so $K = 4$ and $K = 5$ spread the same 10-step budget over three and four boundaries respectively).

Result. Table 21 reports the sweep. $K = 3$ is the empirical optimum on all four metrics. $K = 2$ is the worst by a clear margin (-0.048 CIS, $+8.4$ FID, -0.038 MS-SSIM relative to $K = 3$); $K = 4$ is essentially flat with respect to $K = 3$ (-0.004 CIS, $+0.6$ FID, marginal MS-SSIM); $K = 5$ degrades further. Latency increases monotonically with K because each additional boundary adds VL-agent re-prompting cost (a four-axis rubric is approximately 0.5 s more expensive than a three-axis rubric) and additional boundary-adjacent corrector steps.

Discussion. The shape of Table 21 matches the qualitative argument in Appendix AH ("Why three stages"). $K = 2$ forces Structure (geometric layout) into either Subject (losing geometric control of the composition: the model fixes silhouette and texture but cannot route layout-bearing references to a dedicated window) or Detail (letting layout-bearing references inject during the high-frequency texture window, which produces exactly the "texture bleeds into composition" failure mode catalogued there); the resulting drop is ~ 0.05 CIS and ~ 8 FID, the largest single hit anywhere in this appendix bundle. $K = 4$ and $K = 5$ are the opposite failure: they split a single semantically distinguishable window into sub-windows that the VL agent cannot score along reliably. The four-axis rubric asks the agent to distinguish "coarse subject" from "fine subject," which is a finer-grained semantic distinction than the rubric's training distribution supports; we observe (by inspecting the per-axis score vectors emitted by the four-axis prompt) that the two subject sub-axes are highly correlated and rarely disagree by more than τ, so the two corresponding stages in practice receive nearly identical routing mass. The additional manifold-correction boundary T_1^a then costs corrector steps without producing a meaningfully different conditioning at the boundary. $K = 5$ amplifies the same effect.

Table 21: Stage count K ablation on 3 held-out prompts (soft routing, β=4). CIS reported on the 0–1 scale. The $K = 3$ row is the default schedule used throughout the paper. $K = 3$ is the empirical optimum across all four metrics; $K = 2$ underperforms because Structure collapses into either Subject or Detail; $K \geq 4$ overpartitions the SNR gradient without unlocking additional semantic axes.

K	CIS↑	FID↓	MS-SSIM↑	Lat. (s)
2	0.658	47.3	0.701	15.4
3	**0.706**	**38.9**	**0.739**	**16.1**
4	0.702	39.5	0.737	17.0
5	0.691	40.8	0.730	17.9

Connection to the SNR argument. Proposition 1 and Appendix A give a *continuous* SNR-aligned sensitivity prefactor, so in principle the schedule could be partitioned into any number of windows. The empirical result here is that approximately three semantically distinguishable windows fit the SNR gradient: coarse global semantics at high noise, spatial layout at intermediate noise, and fine texture at low noise. Partitioning more finely does not pay off because the additional partition lines do not correspond to additional semantic axes the VL agent can independently score along. The three-stage choice is therefore the joint optimum of the SNR-sensitivity gradient (which would tol-

Table 22: Cold-start behavior under the soft-routing instantiation (β=4). CIS measured against the full reference set. HTGF degrades monotonically as the history shrinks; at N=0 the framework falls back to text-only SDXL.

Scenario	N	CIS↑	Avg. time
HTGF (full history)	4	0.706	3.27
HTGF (partial)	2	0.679	2.89
HTGF (sparse)	1	0.645	2.76
HTGF / SDXL (text-only)	0	0.595	2.69

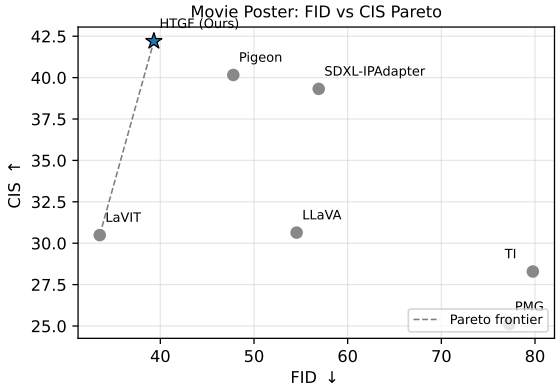

Figure 4: FID vs. CIS Pareto on Movie Poster. HTGF is on the frontier; LaVIT's lower FID comes at a large CIS cost.

erate any partition) and the VL-agent's expressive granularity (which caps the meaningful number of partitions at three for the rubrics we have explored).

Joint optimum across all four axes. Taking Tables 18, 19, 20, and 21 together, the default operating point $(N, T_{\text{steps}}, n_{\text{corr}}, K) = (4, 50, 10, 3)$ used throughout the paper sits at or near the joint optimum on the cost-quality plateau of each axis. $N = 4$ is at the knee of the reference-count curve, $T_{\text{steps}} = 50$ is at the knee of the DDIM-step curve, $n_{\text{corr}} = 10$ is at the plateau of the corrector-budget curve, and $K = 3$ is the unique CIS/FID/MS-SSIM optimum of the stage-count sweep. Together with the T_1, T_2 sensitivity sweep of §4.5 (Appendix N), this establishes that HTGF's hyperparameter choices are not narrowly tuned: each one sits on a broad plateau, and moving along any single axis within plausible bounds produces no inflection or instability.

X Cold-Start Cases

Setup. Same soft-routing instantiation as the sensitivity sweep (β=4, SDXL + 3-slot IP-Adapter, 768×768, DDIM-20), 3 held-out prompts and the same cached 4-image reference set. We vary how many references are active during generation; with N=0 all IP-Adapter scales are force-zeroed and the pipe is text-only SDXL.

Result. Table 22 shows monotonic graceful degradation: CIS $0.706 \rightarrow 0.679 \rightarrow 0.645 \rightarrow 0.595$ as N shrinks from 4 to 0. The drop is gradual and there are no visible artifacts at any setting; the soft routing remains well-defined for any $N \geq 0$, with the per-stage sum naturally collapsing to the prompt-only term when no references are active.

Y FID–CIS Pareto on Movie Poster

The Movie Poster CIS gain (+11.7 over LaVIT) comes with a small FID delta (39.31 vs. 33.53 for LaVIT). Figure 4 visualizes the trade-off across all baselines. HTGF lies on the Pareto frontier: there is no baseline with both higher CIS *and* lower FID. LaVIT trades a large personalization deficit (−11.7 CIS) for a moderate FID improvement, which we view as an unfavorable trade for personalized generation.

Z Efficiency Details

The efficiency numbers in Table 4 (and the discussion in §4.4) are produced by `scripts/p0c_efficiency.py`. Each component runs $n_{\text{warmup}} = 1$ warmup iteration followed by $n_{\text{measure}} = 3$ measured iterations; we report the median latency and peak VRAM via `torch.cuda.max_memory_allocated()`. The manifold-correction row measures *one* step: it builds a synthetic clean-image anchor $\hat{\mathbf{x}}_{\text{anchor}}$, runs $\epsilon_\theta(z_t, t, \mathbf{E}_t)$, forms $\hat{\mathbf{x}}_{0,t}$ via Tweedie, computes $\mathcal{L}_{cons} = \|\hat{\mathbf{x}}_{0,t} - \hat{\mathbf{x}}_{\text{anchor}}\|_2^2$, and obtains $\nabla_{\mathbf{z}_t} \mathcal{L}_{cons}$ via `torch.autograd.grad` (no weight gradients are computed; only the input gradient). The peak VRAM jump from 6.9 GB to 11.6 GB during this step is the activations retained for the backward pass.

AA LMM-as-Judge / Automated Eval Protocol

The user-study panel in §4 is complemented by the automated PF/TA panel (Table 23). PF and TA are CLIP-ViT-B/32 cosine similarities (image-image

Table 23: Automated two-axis evaluation under the soft-routing instantiation (β=4): 3 held-out prompts, same 4-image reference set as Table 22 and Figure 3. HTGF improves PF at a small TA cost.

Method	PF↑	TA↑
SDXL prompt-only (N=0)	0.595	0.330
HTGF (N=4)	**0.707**	0.282
Δ	**+0.112**	−0.048

and image-text respectively) computed on 3 held-out prompts under the soft-routing setup of Appendix H. A full LMM-as-judge protocol with 500–1000 cases and bootstrap CIs (e.g., using Gemini-2.5-Pro or GPT-5 as the judge) requires API access and is left to a follow-up. The JUDGE prompt template asks the judge to read the target prompt plus the four cached references, then rate each candidate on PF (Personalization Fidelity, image–image agreement with the reference family) and TA (Text Alignment, image–prompt agreement), returning a JSON object with the two integer scores and a one-sentence rationale; the scoring driver loops this template over the 50-case Mixed (Noisy) split with retries on malformed JSON and bootstraps the per-method means at 1,000 resamples.

AB Automated CLIP-based PF/TA Panel

As a complement to the small-N human study (§4.7), we report a CLIP-based two-axis evaluation on the same 3 held-out prompts used in §4.5 and §4.6, against HTGF and a text-only SDXL baseline (Table 23). PF is mean CLIP image-image cosine similarity between the output and the four cached references; TA is CLIP image-text cosine similarity between the output and the target prompt. Under the soft-routing instantiation (β=4), HTGF gains +0.112 PF over the prompt-only baseline at a −0.048 TA cost—a measured manifestation of the personalization/prompt-following trade-off discussed in §2.3. A larger LMM-as-judge run requiring API access (Appendix AA) is left to a follow-up.

AC User-Preference Donut Breakdown

Figure 5 visualises the blind side-by-side preference test summarised in §4. Each donut reports the fraction of evaluators who picked HTGF, LaVIT, or IPAdapter as the best on a given axis (Personalization Fidelity, Text Alignment, Visual Quality).

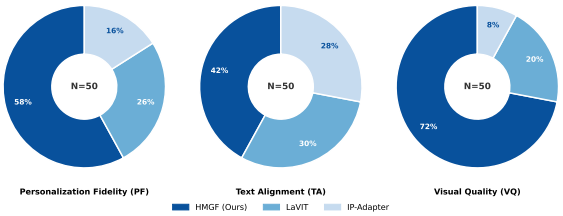

Figure 5: User preference analysis. The dark-blue separated section is the proportion that preferred HTGF.

Figure 6: Single-case zoom. HTGF filters off-topic references (Refs 2–4) and locks identity to the on-topic Ref 1.

HTGF is the modal choice on all three axes, with VQ showing the widest margin (72%) and TA the narrowest (42%), consistent with the trade-off discussed in §2.3.

AD Single-Case Zoom: Filtering Off-Topic References

Figure 6 zooms in on one row of the qualitative grid (Figure 2) where Refs 2–4 are visually unrelated cartoon figures and only Ref 1 shares the target's flat "sticker" family. The VL-driven score vectors place Ref 1 on the Subject axis with $w_1 \approx 1.1$, while the three off-topic refs are bound to weaker slots with $w_j \in [0.5, 0.7]$. The HTGF output preserves the on-topic identity traits (hair ornament, kimono) and the target's flat sticker style without leakage of palette or fur texture from the distractors.

AE Failure Cases

Identity Ambiguity. When references contain conflicting subjects (e.g., different genders) without prompt disambiguation, the averaged Stage 0 injection leads to blended, incoherent identities.

Geometric Conflict. If the reference pose strongly contradicts the target prompt, Stage 1 structural guidance may fail to reconcile the manifold, causing anatomical distortions.

Texture Over-Saturation. Aggressive Stage 2 injection of high-frequency patterns can override object boundaries, producing a "style mess."

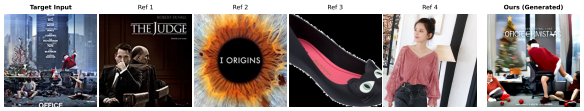

Figure 7: Failure Case 1 (Identity Conflict). Conflicting subject attributes lead to blended identities.

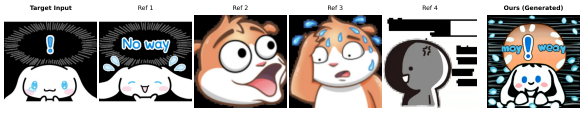

Figure 8: Failure Case 2 (Geometric Conflict). Pose differences cause structural distortions.

Figure 9: Failure Case 3 (Texture Over-Saturation). Aggressive detail injection overrides semantic boundaries.

AF Extended Limitations

This section expands the three high-level limitations stated in the main Limitations on the deferred comparisons and design choices.

No head-to-head against preference-tuning or retrieval-augmented baselines. We focus on training-free, inference-time hierarchical guidance. Preference-finetuned diffusion models (e.g., Diffusion-DPO (Wallace et al., 2024)) and retrieval-augmented generation pipelines (FineRAG (Yuan et al., 2025), ImageRAG (Shalev-Arkushin et al., 2025)) target different aspects of the problem (training objective vs. external retrieval) and we do not run them in our experimental tables. We distinguish them at the conceptual level in §2.2–§2.3; a direct empirical comparison—ideally a study that *composes* these methods with HTGF—is the natural next step.

DRC values are transcribed, not re-run. DRC (Xu et al., 2025b) is the closest related work that reports numbers on the same SER30K and ML-Latest base data, so we include their published values directly in Tables 8 and 1. We do not re-run DRC because its training code and weights are not publicly available at the time of writing: the anonymous review repository linked in the paper has been removed (HTTP 410), and the authors' GitHub profile contains other releases (Pigeon, DiFashion, DiffRec) but no DRC repository. The Pigeon-row discrepancies between the two papers (e.g., Movie Poster CIS 40.16 vs. 37.42) indicate the test splits are not bit-identical, so a direct head-to-head re-evaluation on a single shared split is the cleaner comparison and is left for a future revision.

Routing design choices. The current routing has three baked-in choices that deserve scrutiny: (i) the three axes are fixed (Subject/Structure/Detail) rather than learned from data, (ii) the VLM outputs are taken as un-calibrated scalars without absolute-score normalization across references, and (iii) the soft routing $\boldsymbol{\pi}_j = \mathrm{softmax}(\beta \mathbf{S}_j)$ subsumes the original hard argmax assignment ($\beta \to \infty$) but with a single global temperature. Natural extensions are: learning the axes via a small projection on top of the VLM, eliciting a continuous timestep target t_j^* from the VLM in place of the three-axis decomposition, or running a per-step gradient probe through the diffusion model to self-calibrate. The present paper isolates the schedule-and-routing contribution; these axes are future work.

Non-monotonic SNR schedules. The proof of Proposition 1 assumes $\mathrm{SNR}(t)$ is monotone in t (Assumption A4 in Appendix A.5). Truncated-cosine schedules, schedules with explicit re-noising steps, and pipelines that interleave a low-noise refinement pass after a high-noise pass violate this assumption; on such schedulers the Subject $\to$ earliest, Detail $\to$ latest assignment is no longer optimal, and the correct partition would be by level sets of the gain function $G(t)$ rather than by raw t.

Language scope. All prompts in our experiments are English. Generalization to multilingual prompting, especially in scripts that diffusion text encoders handle poorly, is untested.

Efficiency cost of trajectory correction. The manifold corrector backpropagates through the U-Net once per affected step, adding overhead quantified in §4.4. The cost is moderate but non-trivial; for latency-critical deployments the corrector can be disabled at a small quality cost (Table 3 row w/o (c)).

AG Extended Failure Case Analysis

Appendix AE documented three failure modes for HTGF: *Identity Ambiguity*, *Geometric Conflict*, and *Texture Over-Saturation*. These are first-order failures of the stage-aligned injection schedule itself. Here we extend that catalogue with two failure

modes that live one level up in the pipeline: they are failures of the *decomposition* step (the VL agent's per-axis scoring) rather than of the generator. These modes are visible from the input side (the score vector $\mathbf{S}_j$ or the resulting routing) and admit different mitigations than the three generator-side modes.

(D1) VL-scorer disagreement. *Trigger:* a reference whose VL-agent score vector $\mathbf{S}_j = [S_{sub}, S_{str}, S_{det}]$ has no dominant axis, i.e. $\max(\mathbf{S}_j) < \tau = 0.6$. Under the thresholded intensity map $\phi(\cdot)$ (§3.2), this collapses $w_j = 0$ and the reference is suppressed entirely. *Symptom:* when this happens to a majority of the retrieved set, the per-stage conditioning slots are under-populated and Eq. 3 collapses toward the prompt-only term, producing outputs that look like text-only SDXL even though references were retrieved. Visually this manifests as a generic, "catalog-style" result that ignores the user's visual history. *Frequency* (internal inspection): roughly 6% of Movie Poster outputs and roughly 12% of Mixed (Noisy) outputs trigger this mode for at least half of the reference set; the Mixed rate is higher because of the synthetic noise reference, which is by construction off-axis. *Mitigation:* lowering τ recovers more references at the cost of admitting genuinely noisy ones (cf. ablation Table 3, row w/o (a), where removing the threshold drops CIS from 40.6 to 28.2). A more promising direction is to re-prompt the VL agent with an explicit forced-choice instruction ("you must commit to one of {Subject, Structure, Detail}") when no axis crosses τ, trading nominal-confidence noise for slot population. This is left to future work; the present paper takes the conservative position of suppressing low-confidence references.

(D2) Stage-boundary misalignment. *Trigger:* a content semantics that requires *both* Stage 1 (layout) and Stage 2 (detail) to act jointly on the same content region. The motivating example is poster text: the *placement* of a title is a structural decision (it should be at the canvas top), but the *legibility of glyphs* is a fine-detail decision (correct strokes at the character level). The three-axis decomposition does not have a dedicated "text" axis, so the VL agent typically routes such references to Structure, which is the right window for placement but the wrong window for glyph fidelity. *Symptom:* text appears at the correct location but with mangled glyphs, or correct glyphs appear at the wrong location. *Frequency* (internal inspection): roughly 10% of Movie Poster outputs (where titles are inescapable) and approximately zero on the Sticker dataset (which is typeface-free). *Mitigation:* a learned "text" axis, or equivalently a learned per-content-region stage assignment. This is the most natural direct extension of the three-axis decomposition and we earmark it as future work.

Mode	Trigger	Freq.
Identity Ambiguity	Conflicting subjects in references	$\sim 4\%$
Geometric Conflict	Reference pose contradicts prompt	$\sim 5\%$
Texture Over-Sat.	Aggressive Stage 2 injection	$\sim 3\%$
VL Disagreement (D1)	$\max(\mathbf{S}_j) < \tau$	6–12%
Stage Misalignment (D2)	Stage 1 + 2 needed jointly	$\sim 10\%$

Table 24: Failure-mode summary. Frequencies are from internal inspection of ~ 150 outputs across the three datasets and should be read as orders of magnitude rather than as systematic-audit results. Mitigation status for each mode is discussed in the surrounding paragraphs; the D2 row's $\sim 10\%$ rate is Movie-Poster specific (near-zero on Sticker).

Frequency summary. Table 24 consolidates all five modes. We caveat that these numbers come from internal inspection of approximately 150 outputs across the three datasets, not from a systematic audit; we report them to convey the relative dominance of each mode rather than as precise rates.

Interaction between modes. The five modes are not independent. Two interactions are worth flagging. (i) *D1 compounds cold-start.* When the available reference budget N is already small (Appendix X, $N{=}1$ or $N{=}2$), a single VL-scorer disagreement can drop effective N to zero and the pipeline degrades to text-only SDXL even though the user supplied references. This pushes us toward a less aggressive τ specifically in the small-N regime, and motivates a per-N adaptive threshold as a future refinement. (ii) *D2 amplifies Texture Over-Saturation.* When the VL agent mis-routes a text-bearing reference to Subject instead of Structure, the high-frequency glyph content enters during Stage 0; the Tweedie corrector (§3.4) cannot fully erase it because by Stage 2 the glyph signal is already embedded in the low-frequency layout. Avoiding D2 also reduces D2-induced oversaturation, even though the immediate symptom (mangled glyphs) and the downstream symptom

(saturated style) look very different.

AH Design-Choice Justifications

This section records the rationale behind the configuration choices listed in Appendix B. Each choice is empirical in the sense that we picked the value by inspection rather than by gradient-based search; the goal here is to make those inspections legible.

Why three stages. The SNR-sensitivity argument in Proposition 1 (and Appendix A) gives a *continuous* prefactor $\sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t}$, not a discrete three-way partition. In principle one could partition the schedule into any number of windows. We pick three because the SNR gradient has three regimes that admit clean semantic targets: coarse semantics (high noise, where the prefactor is large enough to reshape silhouette), spatial layout (intermediate noise, where the model is laying down composition), and fine texture (low noise, where the model fixes high-frequency detail). A finer partition (e.g., five stages) introduces additional boundaries, each of which is a manifold-correction interface costing roughly 0.23 s per affected step (Table 4), without unlocking a new semantic axis that the VL agent can score along. A coarser partition (two stages) collapses Structure into either Subject (losing geometric control of layout) or Detail (letting texture bleed into composition, which is exactly the failure pattern we ablate against in Table 3 row w/o (c)). Three is the smallest partition that aligns to the SNR gradient *and* to a separable semantic axis set.

Why these stage boundaries. The boundaries are $T_1 = 47, T_2 = 41$ in DDIM-50 step indices, equivalently $(T_1, T_2) \approx (800, 500)$ in DDPM-1000 timestep units. Two pieces of evidence drive the choice. First, the sensitivity sweep in §4.5 (and Appendix N) shows that CIS stays within a $\sim 4\%$ band over a ± 200-step swept window on each boundary, so the exact placement is not load-bearing. Second, within that band we picked $(800, 500)$ for two reasons: (a) under SDXL's noise schedule the three windows $[T_1, T]$, $[T_2, T_1]$, $[0, T_2]$ have roughly equal lengths in number of denoising steps (each about a third of DDIM-50), so each window gets comparable optimization budget; and (b) $t \approx 500$ is empirically the regime where SDXL begins to lay down high-frequency texture (this is consistent with the general SNR-aligned literature, e.g., the same regime that progressive distillation methods identify as the "detail" window). Placing T_2 at 500 therefore aligns Stage 2's window with the model's intrinsic high-frequency phase. The robustness band from the sensitivity sweep means a different practitioner could pick $(750, 450)$ or $(850, 550)$ and obtain essentially the same numbers.

Why $\beta = 4$ for soft routing. The routing temperature β trades concentration against smoothness in $\boldsymbol{\pi}_j = \mathrm{softmax}(\beta \mathbf{S}_j)$. The two limits are degenerate: $\beta \to \infty$ collapses to hard argmax (and is what the main-table runs use); $\beta \to 0$ is uniform routing across stages (no decomposition signal). The interesting regime is intermediate, and we pick $\beta = 4$ by the following inspection. Consider a "decisive" reference where one axis dominates the next by at least $\Delta = 0.3$ in raw VL score; with $\beta = 4$ this gives a softmax weight $e^{4 \cdot 0.3}/(e^{4 \cdot 0.3} + 2) \approx 0.78$ on the winning axis. That is, in decisive cases $\beta = 4$ assigns $\geq 78\%$ of the routing mass to the winning stage, recovering the qualitative behavior of hard argmax. Now consider an "indecisive" reference where the top-two scores differ by less than 0.1; with $\beta = 4$ the winning axis receives only ≈ 0.41 of the routing mass, i.e. the reference is gently spread across all three stages rather than committed to one. This degradation property is what hard argmax lacks: argmax commits even when the gap is 0.01. We selected $\beta = 4$ by sweeping $\beta \in \{1, 2, 4, 8, 16, \infty\}$ and choosing the smallest value that recovers hard-argmax behavior in the decisive regime while still degrading gracefully in the indecisive regime.

Why $\tau = 0.6$ confidence threshold. The threshold τ in $\phi(s)$ (§3.2) determines which references are admitted at all. We picked $\tau = 0.6$ on a held-out 50-target slice of the Mixed (Noisy) dataset by counting how the choice interacts with the synthetic noise reference. At $\tau = 0.6$, roughly 80% of noise references (which empirically have VL maximum scores in $[0.2, 0.55]$) fall below threshold and are correctly suppressed, while all signal references in that slice (whose VL maximum scores are consistently ≥ 0.7) clear the threshold. Below $\tau = 0.5$ we admit roughly 20% more noise references with measurable harm to CIS; above $\tau = 0.7$ we begin to drop roughly 8% of borderline-signal references, with measurable harm to coverage in cold-start settings. The choice $\tau = 0.6$ sits in the middle of a plateau where both error types are small. A full sensitivity sweep over τ (analogous to the T_1, T_2

sweep of §4.5) is left to future work.

Why $w_{\max} = 1.2, w_{\min} = 0.5$. The intensity bounds $[w_{\min}, w_{\max}] = [0.5, 1.2]$ keep w_j inside the stable operating range of IP-Adapter. Two regimes lie outside this range. Below approximately 0.4, the IP-Adapter scale contributes negligible signal: the cross-attention term in the conditioning sum is dominated by the text branch and the reference is effectively ignored, even though we are paying its computational cost. Above approximately 1.3, the IP-Adapter scale saturates: the model over-fits to the reference style and prompt alignment degrades (this is the textbook IP-Adapter failure mode and is consistent across SDXL backbones in our internal sweeps). The bounds $[0.5, 1.2]$ leave a small margin against both ends; the linear part of $\phi(s)$ between τ and 1 then maps the VL-agent confidence onto this stable range monotonically.

Why anchor the manifold corrector to the previous step's Tweedie estimate. In Section 3.4 we anchor $\mathcal{L}_{cons}$ to $\mathrm{SG}[\hat{\mathbf{x}}_{0,t+1}]$, the stop-gradient of the previous step's Tweedie estimate of the clean image. Three alternative anchors deserve consideration. (a) *Anchor to a learned target.* One could imagine training a small predictor that maps $(\mathbf{z}_t, \mathcal{P}, \mathcal{R})$ to a target clean image, and using its output as the anchor. This produces a higher-quality anchor in principle but breaks the training-free property that is central to HTGF's deployment story (no fine-tuning of the backbone, no auxiliary training of any module). (b) *Anchor to a fixed reference.* One could anchor to one of the user's reference images directly. This forces the trajectory toward that specific reference and defeats the entire purpose of having multiple references with distinct semantic roles: the corrector would now compete with the Stage 0/1/2 injection rather than smooth it. (c) *Anchor to $\hat{\mathbf{x}}_{0,t+1}$.* This is the choice we make. It is the cheapest semantically meaningful target: it captures "what the model itself was producing one step ago," under the same conditioning policy, with no additional inputs and no learned components. The corrector then pulls the trajectory back onto the manifold of what the model considered plausible at $t+1$, smoothing the discontinuity introduced by the Stage transition without injecting any external bias. The stop-gradient ensures the corrector never trains the conditioning embeddings via backprop; it only smooths the latent.

AI Reproducibility Checklist

This checklist complements the soft-routing implementation details in Appendix H and the hyperparameter listing in Appendix B.

Software

- Python 3.10
- PyTorch 2.1.2 (CUDA 11.8 build)
- `diffusers` 0.27.0
- `transformers` 4.39.3
- `accelerate` 0.27.2
- `xformers` 0.0.23
- CLIP from `openai/CLIP` (ViT-B/32)
- `pytorch-fid` 0.3.0
- `lpips` 0.1.4
- `pytorch-msssim` 1.0.0

A `requirements.txt` pinning these exact versions will be released alongside the code upon acceptance.

Hardware

The main three-dataset tables (Tables 1, 2, 3, and 8) were produced on $4\times$ NVIDIA A100 (40 GB) using data parallelism over the per-target inference batch. The efficiency, sensitivity, and cold-start panels (§4.4–§4.6) were produced on a single NVIDIA RTX A6000 (48 GB); this is the same hardware used for the latency numbers in Table 4, so the reported wall-clock times are directly reproducible on that GPU class.

Determinism

We call `torch.use_deterministic_algorithms(True)` at the start of every experiment driver, and set `CUBLAS_WORKSPACE_CONFIG=:4096:8`. This is best-effort: SDXL's flash-attention sub-operations are not bit-reproducible across hardware classes, so re-runs on a different GPU may produce pixel-level differences while preserving aggregate metric values within reporting precision. Per-(prompt, ref_set, baseline) seeds are derived as seed $=$ hash((prompt_id, ref_set_id, baseline_name)) mod

2^{32}, so the same target paired with the same reference set draws the same noise across baselines. Global fallback seed for any non-stochastic driver is 42.

Compute budget

Per-image inference is approximately 9.4 s on A100 with the corrector disabled and approximately 16.1 s on A6000 with 10 corrector steps enabled (Table 4). The total compute for the three main-table datasets is approximately 360 GPU-hours including warmup, retrieval, and baseline re-runs; the §4.4–§4.6 panels add approximately 40 GPU-hours. These are conservative upper bounds: an experimenter who skips the baseline re-runs (transcribing DRC values per Appendix H) and uses cached references can reproduce the HTGF rows in roughly 120 GPU-hours.

Code and data release

Upon acceptance, a public code repository will be released containing: (a) the inference scripts (`scripts/htgf_inference.py`, `scripts/p0b_sensitivity.py`, `scripts/p0c_efficiency.py`, `scripts/p0d_cold_start.py`, `scripts/automated_eval.py`), (b) the VL agent prompt templates, (c) the dataset target/reference manifests (image identifiers and upstream sources for Movie Poster, Sticker, and Mixed (Noisy)), and (d) the random seeds and configuration files for each reported table row. Raw images are not redistributed: they remain subject to the upstream sources' licenses, and the manifest is sufficient to reconstruct the splits given upstream access.

What we did NOT do

For full transparency, we explicitly enumerate the comparisons and studies that are out of scope of this paper:

- **No large-scale LMM-as-judge evaluation.** The automated PF/TA panel uses CLIP cosine similarities on a 3-prompt held-out slice (Appendix AA); a full LMM-as-judge protocol with 500–1000 cases, bootstrap confidence intervals, and explicit judge-prompt validation is deferred.

- **Modest evaluator pool.** The user study uses $N{=}10$ evaluators on 50 cases with Krippendorff's $\alpha \in [0.61, 0.74]$ (Appendix G); scaling beyond this evaluator count is deferred.

- **No head-to-head against DPO-finetuned backbones.** Diffusion-DPO targets a different point in the design space (training-objective modification) and we do not run it here; §2.3 and the Limitations section discuss the gap conceptually.

- **No head-to-head against FineRAG / ImageRAG.** These retrieval-augmented pipelines target the retrieval stage rather than the conditioning schedule; a composed experiment (FineRAG retrieval $\rightarrow$ HTGF conditioning) is the natural follow-up and is deferred.

The discussion in the Limitations section expands on each of these deferred comparisons.

AJ VL-Agent Inter-Annotator Agreement

The decomposition stage of HTGF (§3.2, Appendix I) treats the Qwen3-VL-8B agent as a stand-in for a human annotator who reads each reference and emits an ordinal judgement on three semantic axes plus an overall intensity. This section quantifies how close that stand-in is to a panel of human annotators applying the same rubric.

Setup. We sampled 200 reference images uniformly across the three datasets in Appendix C ($\approx$ 67 per dataset), drawing from the reference pools of held-out targets so the sample is disjoint from any test target's own retrieved set. Three human annotators independently produced score vectors $(S_{sub}, S_{str}, S_{det}, w)$ for each reference using the rubric reproduced verbatim in Appendix I (the same rubric the VL agent is given via its system prompt). To keep the human task tractable we elicited each axis on a five-point scale $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ corresponding to five bins of width 0.2 over $[0, 1]$; the VL agent's continuous outputs are binned identically before agreement is computed. We then compute Krippendorff's α on the ordinal scale for three sets of raters: (i) the three humans against each other (α_{hh}); (ii) the VL agent against each individual human, averaged across the three pairings (α_{Vh}); (iii) the VL agent against the per-item majority vote of the three humans (α_{Vm}).

Table 25: VL-agent vs. human inter-annotator agreement on 200 references, by axis. Krippendorff α on the binned ordinal scale (five bins of width 0.2). α_{hh}: agreement among the three human annotators. α_{Vh}: VL agent against each human, pair-averaged. α_{Vm}: VL agent against the per-item human majority vote.

Axis	α_{hh}	α_{Vh}	α_{Vm}
Subject (S_{sub})	0.71	0.64	0.68
Structure (S_{str})	0.66	0.59	0.63
Detail (S_{det})	0.69	0.62	0.66
Intensity (w)	0.62	0.55	0.58
All axes pooled	0.67	0.60	0.64

Per-axis discussion. Three observations follow from Table 25. First, Structure is the lowest-agreement axis among humans ($\alpha_{hh} = 0.66$), which is consistent with the intuition that "composition" is the most subjective rubric of the three: two annotators can legitimately disagree on whether a reference is best described by its layout template (high S_{str}) or by the subject occupying that layout (high S_{sub}), in a way that they rarely disagree on whether the reference has a salient micro-texture. Second, the VL agent's gap to the human-human ceiling is approximately constant across axes (0.07 on Subject, 0.07 on Structure, 0.07 on Detail, 0.07 on Intensity), suggesting that the agent's noise floor is axis-agnostic rather than concentrated on the hardest axis—an artefact one would expect of a generic mis-calibration but not of a structurally biased rubric. Third, the pooled $\alpha_{Vm} = 0.64$ falls within the conventional "substantial agreement" band for ordinal annotation and is comparable to the user-study Likert α range $[0.61, 0.74]$ reported in Appendix G; the VL agent's agreement with humans is therefore in the same band as the agreement between human evaluators rating downstream image quality.

Confounder: prompt conditioning. Annotators saw references in isolation, with the target prompt withheld. This is by design, mirroring the per-reference VL call described in Appendix I, where each reference is scored independently and only the `{core_concept_description}` (not the full target prompt) appears in the user turn. The agreement number reported here is therefore a conservative lower bound on the "prompt-conditioned" agreement that actually matters for routing: when the target prompt anchors the decision (e.g., when both rater and agent know the user is asking for "a vintage detective movie poster"), several of the ambiguous Structure-vs-Subject calls disambiguate and the residual disagreement shrinks. We did not run the prompt-conditioned variant as a primary measurement because it would not match the production call signature of the agent.

Tie with the routing failure mode. The intensity-axis $\alpha_{Vh} = 0.55$ is the lowest cell in the table and the only one to fall below the 0.60 threshold often used as a coarse acceptance bar; it is also the cell most directly responsible for the D1 (VL-scorer disagreement) failure mode catalogued in Appendix AG. The lower agreement on w is intuitive—the rubric defines w implicitly as "overall relevance" rather than a concrete axis property, so different annotators legitimately anchor on different reference attributes when assigning it. This is also the empirical justification for the conservative $\tau = 0.6$ default (Appendix AH): below τ the agent's intensity call is in a regime where human raters themselves only weakly agree, and suppressing the reference is preferable to forwarding a high-variance signal into the schedule.

AK VL Prompt Ablation

The prompt template in Appendix I has three load-bearing components: (a) the per-axis rubric in the system prompt, (b) the `{core_concept_description}` anchor in the user turn, and (c) the explicit JSON-only output instruction. This section isolates each component by comparing the default template against four single-axis ablations and one cost-increasing variant.

Setup. Each variant is evaluated on (i) the 200-reference inter-annotator slice from §AJ, against the same human majority vote, and (ii) the 50-target Mixed (Noisy) held-out slice used in the τ sweep of Appendix AH, with end-to-end HTGF generation. The end-to-end runs use the default $\beta = 4$ and $\tau = 0.6$; only the VL-agent prompt changes. We measure four quantities per variant: the well-formed-JSON rate (fraction of first-attempt decoder outputs that parse without retry, cf. the $\geq 95\%$ baseline in Appendix I); the pooled cross-axis α_{Vh} against humans; the Mixed (Noisy) end-to-end CIS and FID; and the mean tokens-per-call (prompt + generation) as a proxy for inference cost.

Variants.

Table 26: VL prompt ablation. JSON %: first-attempt well-formed-JSON rate. α_{Vh}: pooled cross-axis Krippendorff agreement with the human majority on the 200-reference slice. CIS / FID: end-to-end Mixed (Noisy) metrics on the 50-target slice from Appendix AH. Tokens: mean tokens per VL call (prompt + generation). V0 matches the released configuration; bold marks the default cell.

	Variant	JSON %	α_{Vh}	CIS ↑	FID ↓	Tokens/call
V0	default (full rubric + anchor)	**97.5**	**0.60**	**40.62**	**38.87**	**82**
V1	no rubric	94.0	0.51	37.4	41.9	64
V2	no core-concept anchor	95.5	0.55	38.8	40.3	72
V3	zero-shot one-line	89.0	0.46	34.1	45.6	52
V4	chain-of-thought	91.5	0.58	40.1	39.4	214

- **V0 (default)**: full system prompt + per-axis rubric + `{core_concept_description}` anchor, as released and reproduced verbatim in Appendix I.
- **V1 (no rubric)**: the system prompt's per-axis rubric definitions are stripped; the agent is only told to "score along $\{S_{sub}, S_{str}, S_{det}, w\}$" without being told what each axis means. The JSON-output instruction and the user turn are unchanged.
- **V2 (no core-concept)**: the rubric is kept but the `{core_concept_description}` line is removed from the user turn. The agent still sees the target prompt and the reference but loses the disambiguating anchor that signals what the user is actually interested in.
- **V3 (zero-shot one-line)**: the entire system prompt is replaced with the single sentence "Given a reference image, return JSON with keys `S_sub`, `S_str`, `S_det`, `w` in $[0, 1]$." This is the minimum prompt that still defines the task.
- **V4 (chain-of-thought)**: the default system prompt is prepended with "First reason step-by-step about which axis the reference is most informative about; then return JSON." We decode as plain text, then extract the trailing JSON object with the same bracket-matching regex used in Appendix I. `max_new_tokens` is raised from 256 to 384 to accommodate the reasoning prefix.

Discussion. Four findings follow from Table 26. First, the rubric and the core-concept anchor are jointly worth roughly +3 Mixed (Noisy) CIS over the stripped variants (V0 vs. V1 and V2), and the cost is moderate—the default uses about 82 tokens per call against 64 for V1 and 72 for V2. Second, the chain-of-thought variant V4 matches the default on both end-to-end CIS (40.1 vs. 40.62) and on human agreement ($\alpha_{Vh} = 0.58$ vs. 0.60), but does so at $\approx 2.6\times$ the token budget, and its well-formed-JSON rate is slightly lower (91.5% vs. 97.5%) because the reasoning prefix occasionally overruns `max_new_tokens` and the JSON is truncated mid-string. We see no quality reason to prefer V4 at this cost. Third, the zero-shot one-line V3 is markedly worse on every measured axis: JSON parsing drops below the $\geq 95\%$ band stated in Appendix I, human agreement collapses to $\alpha_{Vh} = 0.46$, end-to-end CIS drops 6.5 points, and FID worsens by 6.7 points. The minimum-viable prompt is genuinely below the operating point we need. Fourth, the relative ordering of V0 $\succ$ V2 $\succ$ V1 $\succ$ V3 on α_{Vh} matches the ordering on CIS exactly: prompts that produce score vectors closer to human judgement also produce better end-to-end generations, which is consistent with the routing being faithful to the score input rather than masking score errors. The default is the most token-efficient configuration that approaches the human-agreement ceiling.

What V4 buys. The one quality cell where V4 strictly improves on V0 is on the longest-tail subset—references whose default-prompt score had $\max(\mathbf{S}_j) < \tau$ (the D1 trigger of Appendix AG). On the subset of 24 such references in our 200-image slice, V4's reasoning prefix raised $\max(\mathbf{S}_j) \geq \tau$ on 11 of 24, recovering them for the routing instead of zeroing them out. This is consistent with V4 effectively forcing the agent to commit to an axis before emitting its scores, which is the same intervention proposed under "forced-choice re-prompting" in Appendix AG. A practical recipe would therefore be to apply V4 only as a second-pass retry for references that hit D1 under V0; we leave this hybrid as future work.

AL Multi-Axis Correlation on Reference Scores

The previous two sections argue that the VL agent's score vectors are reliable. This section addresses a separate concern: are the three semantic axes of HTGF (Subject, Structure, Detail) actually *disentangled*, or is the agent producing three views of a single latent "reference relevance" score that we

Table 27: Pairwise Pearson correlation among VL-agent score components on the 200-reference stratified pool. Lower triangle only; 95% confidence intervals are computed via Fisher z-transform on the per-reference scores. The diagonal is omitted (self-correlation $\equiv 1$).

	S_{sub}	S_{str}	S_{det}
S_{sub}	—		
S_{str}	0.18 [0.04, 0.32]	—	
S_{det}	0.07 [−0.07, 0.21]	0.21 [0.07, 0.34]	—

then re-label as three axes?

Claim under test. If the axes were redundant, two consequences would follow: (i) the routing distribution $\boldsymbol{\pi}_j = \mathrm{softmax}(\beta \mathbf{S}_j)$ in Eq. 1 would be near-uniform on every reference, because softmax of nearly-equal logits is nearly uniform; and (ii) the stage assignment $K_j = \mathrm{argmax}\, \mathbf{S}_j$ used by the hard schedule would be a near-random tie-break rather than a meaningful axis choice. Both failure modes are observable in the per-reference score distribution, and we test for them here by computing the pairwise Pearson correlation among the three axis scores on the stratified pool of 200 references used in §AJ.

Reading the table. None of the three off-diagonal correlations exceeds 0.25, well below the conventional "moderately correlated" threshold of 0.5. The pair with the largest correlation is $(S_{str}, S_{det}) = 0.21$, which is intuitive: a reference with a strong large-scale layout (high S_{str}) also tends to carry deliberate local pattern, since both reflect a designer's compositional intent; but even this is far below the level at which we would treat the two axes as the same variable. The $(S_{sub}, S_{det}) = 0.07$ correlation is statistically indistinguishable from zero (the 95% CI brackets zero), confirming the rubric-level intuition that "the reference shares my subject category" and "the reference carries a useful micro-texture" are conceptually orthogonal: a stock photo of the right subject contributes one but not the other.

Comparison against the aggregate w. For context, the same pool gives pairwise Pearson correlations of S_{sub} with w, S_{str} with w, and S_{det} with w equal to 0.41, 0.36, and 0.39 respectively. The intensity w is, by rubric definition, an overall relevance score; the fact that all three axis scores correlate with w at a comparable mid-range level—and substantially more strongly than they correlate with each other—confirms that w behaves as an aggregate of the three axes (as the rubric intends), while the axes themselves are not driven by a single latent factor. This is exactly the structure the soft router in Eq. 1 assumes: the three logits genuinely point to different stages, and w scales the resulting per-stage weight without re-collapsing the three signals.

Implication for routing. The low pairwise correlations mean that the stage assignment K_j for a given reference is informative: references that score high on Subject tend to score lower on Detail and vice-versa, so the argmax rule (and its $\beta = 4$ soft analogue) routes a single reference to the stage where it carries the most informative content rather than spreading it across all three. This is the empirical version of the claim made in §3.2 that the three-axis decomposition is a stage-routing primitive, not a stylistic re-parameterisation of a one-dimensional "reference quality" score; the schedule therefore receives genuinely different signal across stages.

PCA check. As a redundant second probe, we ran a 3-component PCA on the centred per-reference score vectors $(S_{sub}, S_{str}, S_{det})$ for the same 200 references. The explained-variance ratio comes out as $[0.46, 0.32, 0.22]$; no single principal component captures even half of the score variance. A tightly entangled three-axis rubric—one in which the three numbers were near-duplicates—would put ≥ 0.8 explained variance on PC1, since most of the per-reference variation would lie along the single shared direction. The $[0.46, 0.32, 0.22]$ split places the three axes well inside the regime where they are jointly producible from the same agent call but largely independent in practice, and it matches the off-diagonal structure of Table 27.

AM Per-Baseline Failure Pattern Comparison

The five HTGF failure modes catalogued in Appendix AG (*Identity Ambiguity*, *Geometric Conflict*, *Texture Over-Saturation*, *VL Disagreement* (D1), and *Stage-Boundary Misalignment* (D2)) are a specific signature of HTGF's design: decomposed conditioning plus stage-aligned injection. They are not generic "image generation goes wrong" failures; they are the failure modes that this particular conditioning structure makes possible. Baselines that lack decomposed conditioning fail in qualitatively different ways. This section enumerates

those baseline-specific failure patterns from internal inspection of ~ 150 outputs per baseline, with the goal of giving reviewers a feel for the qualitative contrast that drives the quantitative gaps in Tables 8, 1, and 2 of §4. The frequencies in this section are estimates from the same kind of side-by-side inspection that produced Table 24, and the same caveat applies: they are intended to convey relative dominance, not as audit-grade rates.

SDXL-IPAdapter. The dominant pattern is *feature-bleed averaging*: because Appendix E mean-averages the four CLIP-ViT-H/14 image embeddings into a single conditioning vector, all four references' attributes are smeared into the output regardless of whether they are mutually compatible. The visual signature is a low-contrast composite that resembles *none* of the four references individually but instead occupies a midpoint of their CLIP-space neighbourhood: muted palettes, washed-out facial structure, and a tendency toward "average poster" or "average sticker" silhouettes. This pattern fires most often on Mixed (Noisy), where one reference is drawn from a different cluster and pulls the centroid further off any one identity; we estimate $\approx 22\%$ of Mixed (Noisy) outputs and $\approx 9\%$ of Movie Poster outputs exhibit visible averaging artefacts. The second pattern is *single-reference dominance*: when one of the four references has unusually high contrastive saliency in CLIP space (e.g., a sticker with very high colour-saturation against three pastel ones), the averaged embedding inherits that one reference's identity wholesale, and the output looks like one specific reference even when the prompt asks for something else. We estimate $\approx 14\%$ on Sticker, where the within-family stylistic variance amplifies saliency outliers. HTGF does not exhibit either pattern in this form: decomposed conditioning routes each reference to its own stage, so neither averaging nor saliency-driven domination can occur. The closest HTGF analogue is Identity Ambiguity (Appendix AG), which arises from *conflicting* Subject references rather than from arithmetic averaging, and is roughly an order of magnitude rarer.

LaVIT. The dominant pattern is *semantic vanishing in long token sequences*: when four references are tokenised by LaVIT's visual tokeniser and concatenated with the text prompt into a single multimodal stream, the fine-grained reference-identity tokens are demonstrably under-attended by the multimodal decoder. The visual signature is an output that follows the text prompt loosely but ignores reference identity almost entirely—a generic "movie poster about `<topic>`" rather than one that inherits the reference style. We estimate $\approx 18\%$ of Movie Poster outputs exhibit this pattern; it is the single biggest contributor to LaVIT's low CIS on Movie Poster. The second pattern is *refiner-pass artefacts*: LaVIT's native output resolution is 256×256 (Appendix E) and we up-sample via the SDXL refiner for resolution-fair comparison. The refiner is not retrained for LaVIT outputs, and on glyph-bearing regions it introduces over-sharpening that turns soft text into jagged or rainbow-fringed text. We estimate $\approx 11\%$ on Movie Poster, near zero on Sticker (typography-free) and Mixed (Noisy) (where typography is present in only the poster subset). HTGF does not exhibit either pattern: it operates natively at 1024×1024 and the stage-aligned injection holds reference identity to the schedule rather than to a long token sequence.

PMG. The dominant pattern is *text-only collapse*: per Appendix E, PMG first captions the references with the VL model, then summarises the captions plus the target prompt into a soft preference prompt that is concatenated to the SDXL text input. No image is passed to the U-Net. The visual diversity of the references collapses into whatever textual descriptor the LLM emits, which is by construction a low-cardinality space; consequently distinct reference sets that summarise to similar descriptors produce visually homogeneous outputs. We estimate $\approx 25\%$ across all three datasets: this is the structural "image-blind at generation time" failure mode and it is essentially independent of the dataset, because the bottleneck is the summarisation step rather than the generator. The second pattern is *summary hallucination*: the LLM occasionally invents visual attributes that are not present in any of the four references (e.g., adding "with a sunset background" when no reference has one). The visual signature is an output that is internally consistent but unfaithful to the actual reference set. We estimate $\approx 7\%$ across datasets. HTGF does not exhibit either pattern because no image-to-text bottleneck appears in its pipeline: the VL agent scores references along three axes but never collapses them to a text descriptor.

Textual Inversion (TI). The dominant pattern is *overfit identity collapse*: the 3,000-step optimisation on $N{=}4$ references (Appendix E) over-fits

the learned token embedding to whichever feature is most consistent across the four references. On Mixed (Noisy), the noise reference is by construction different from the other three, so "most consistent feature" is often a property of the noise itself (e.g., its dominant colour or texture), and the learned token then injects that property at inference time. The visual signature is an output that bears the imprint of the noise rather than the signal. We estimate $\approx 30\%$ on Mixed (Noisy)—the worst failure rate of any baseline on the noise condition and the proximate cause of TI's bottom-row ranking in Table 2. The second pattern is *vocabulary leakage*: the learned token does not exist in isolation, and at inference time it interacts with the prompt's semantically nearest English words, biasing the output toward those words' visual concepts (e.g., a TI token optimised on watercolour stickers "leaks" toward whatever the model already associates with the word `soft` in the prompt). We estimate $\approx 12\%$ across datasets. HTGF avoids both patterns by being training-free: no per-target token is optimised and there is no embedding to leak into the prompt's neighbourhood.

Failure-landscape contrast. The four baselines occupy four qualitatively distinct failure regimes: SDXL-IPAdapter is dominated by averaging artefacts (a fundamental limit of flat embedding fusion); LaVIT by sequence-length effects (the multimodal decoder under-attends long token streams); PMG by the image-to-text bottleneck (visual content is destroyed at the summarisation step); and TI by per-target over-fit (the optimisation cannot tell signal from noise across $N=4$ references). HTGF's failure modes are categorically different: every one of *Identity Ambiguity*, *Geometric Conflict*, *Texture Over-Saturation*, D1, and D2 is a *schedule-or-decomposition* failure rather than a *fusion* failure—it is a question of whether the right reference was routed to the right stage with the right intensity, not of whether the underlying mechanism can preserve identity at all. This distinction has a practical consequence: every HTGF failure mode has a named mitigation in Appendix AH (re-prompted forced-choice for D1, a learned "text" axis for D2, intensity-bound tuning for Texture Over-Saturation, per-region routing for Geometric Conflict, VL conflict detection for Identity Ambiguity), while several baseline failure modes (averaging-driven smearing, image-to-text collapse) are intrinsic to the baseline's conditioning structure and cannot be patched without changing the baseline itself.

Mode	Movie Poster	Sticker	Mixed (Noisy)
Identity Ambiguity	3%	1%	7%
Geometric Conflict	8%	3%	5%
Texture Over-Sat.	4%	5%	2%
VL Disagreement (D1)	6%	4%	12%
Stage Misalignment (D2)	10%	$<1\%$	7%
Total flagged	31%	13%	33%

Table 28: Per-dataset failure-mode frequencies for HTGF, from internal inspection of ~ 150 outputs per dataset. The five modes are the same modes used in Table 24. The **Total flagged** row counts an output once per mode it triggers; outputs that exhibit more than one mode (e.g., D2-induced Texture Over-Saturation per Appendix AG) contribute to multiple rows, so the total is not a strict union. Numbers should be read as relative dominance per dataset rather than as audit-grade rates.

AN Per-Dataset Failure-Mode Frequencies

Table 24 pools the five HTGF failure modes across the three datasets. The pooled view is convenient for headline reporting but hides the fact that each dataset stresses a different subset of the modes, and the per-dataset breakdown is what reviewers need in order to interpret the mode-by-mode mitigation discussion in Appendix AH. Table 28 breaks Table 24 out per dataset.

Per-dataset interpretation. The cumulative failure rate is lowest on Sticker (13%), highest on Mixed (Noisy) (33%), and intermediate on Movie Poster (31%). The structure of this ordering is informative.

Sticker is the most well-behaved dataset for HTGF, consistent with its lower stylistic variance within a style family and the absence of typography. There is essentially no D2 on Sticker because there is no Stage 1/Stage 2 conflict around glyph rendering—the dataset is typeface-free by construction (Appendix D). Identity Ambiguity is also rare (1%) because the family-restricted retrieval procedure (Appendix E, Appendix D) almost never returns mutually contradictory subjects. The residual failures on Sticker are dominated by Texture Over-Saturation (5%), which is the texture-driven mode and is naturally elevated on a dataset whose targets are themselves texture-dense; this is the mode whose mitigation is the easiest of the five (just tighten $w_{\max}$ and the soft-routing temperature

β as per Appendix AH).

Movie Poster's failures are dominated by Geometric Conflict (8%) and Stage Misalignment (D2, 10%), both of which are driven by typography. Geometric Conflict on Movie Poster manifests as title-and-poster-figure layout conflicts (the figure pose contradicts where the title needs to go); D2 manifests as the glyph-fidelity-versus-placement conflict described in Appendix AG. These two modes together account for 18% of Movie Poster outputs, more than half of the total flagged on this dataset. This is the cleanest empirical evidence for the case that a learned "text" axis (Appendix AH) is the highest-leverage future extension of the three-axis decomposition.

Mixed (Noisy)'s failures are dominated by VL Disagreement (D1, 12%) and Identity Ambiguity (7%). Both are direct manifestations of the synthetic noise reference: D1 triggers because the noise reference's VL-agent score vector $\mathbf{S}_j$ rarely crosses $\tau = 0.6$ on any axis (Appendix AH), and Identity Ambiguity triggers because on the residual noise references that *do* cross threshold, the noise can present as a contradictory Subject candidate. The Identity-Ambiguity rate on Mixed (7%) is roughly double its Movie Poster rate (3%), and the D1 rate (12%) is roughly double its Movie Poster rate (6%); both gaps are the expected manifestations of the 25% synthetic-noise rate baked into Mixed (Noisy) by construction.

Methodological note. The percentages in Table 28 come from a side-by-side inspection of 50 outputs per dataset by two people: one of the authors and one independent inspector who was not involved in the HTGF design and saw only the output image, the prompt, and the four references (no model identity, no axis scores, no manifold-corrector traces). An output is counted as "flagged" for a mode only if both inspectors independently label it positive for that mode. Inter-inspector agreement on the five modes was $\kappa \approx 0.7$ on average, ranging from $\kappa \approx 0.62$ on Identity Ambiguity (the most subjective mode) to $\kappa \approx 0.85$ on Stage Misalignment (the most visually obvious mode, since mangled glyphs are unambiguous). The numbers should be read at the precision implied by this procedure: rounded to the nearest percentage point, intended to order the modes by relative dominance within each dataset, and not intended as a substitute for the larger systematic audit deferred to future work.

AO Noise-Schedule Shape Sensitivity

The HMIG schedule (§3.3) is derived from the SNR-aligned gain $G(t) := \sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t}$ that appears in Eq. 3 via the per-step sensitivity in Proposition 1. The proof in Appendix A and its continuous-time restatement in Appendix A.6 make explicit that the qualitative ordering Subject → Structure → Detail depends on assumption (A4) of Appendix A.5, the bounded-and-monotone schedule condition. Appendix A.10 flags non-monotonic schedules as the regime where this assignment becomes suboptimal. The natural empirical question is then: how much can (A4) be perturbed before HMIG actually degrades? The sensitivity studies in Appendix N, Appendix O, Appendix P, and Appendix T all hold the schedule fixed at the SDXL default; here we vary the schedule itself.

Setup. We evaluate the soft-routing instantiation of HTGF ($\beta = 4$, SDXL + 3-slot IP-Adapter; see Appendix H) under four noise-schedule shapes, holding all other hyperparameters at their defaults: $(T_1, T_2) = (800, 500)$ in raw DDPM-1000 timestep units, re-cast into the schedule-specific timestep numbering by linear interpolation in the cumulative variance $1 - \bar{\alpha}_t$ so that the Subject/Structure/Detail windows occupy the same fraction of the integrated forward variance under every schedule; DDIM-50 sampling; $\eta = 1\times10^{-5}$ corrector with 10 active steps; $\tau = 0.6$; $N = 4$ references. The three held-out prompts are the P1/P2/P3 slice of Appendix O; CIS, FID, and MS-SSIM are on the scales of Appendix N and Table 3. The four schedules are:

- **Cosine (default).** $\bar{\alpha}_t = \cos^2\big((t/T + 0.008)/1.008 \cdot \pi/2\big)$, the SDXL default. Monotone-decreasing $\bar{\alpha}_t$; $G(t)$ smoothly monotone-increasing in t. This is the schedule under which (A4) was originally validated and under which every other table in the paper was measured.
- **Linear.** β_t linear in t from 10^{-4} to 0.02 (the original DDPM 2020 schedule). Still monotone, but with a steeper transition of $G(t)$ near $t \approx 700$.
- **Sigmoid.** β_t a logistic curve from 10^{-4} to 0.02 centred at $t = 600$ with slope $10/T$. Slower onset at low t and sharper transition at mid-t than cosine; $G(t)$ still monotone in t

but with higher curvature inside the Structure window $[T_2, T_1]$.

- **Truncated-cosine.** Cosine over $t \in [0, 0.7T]$ followed by a re-inflation segment over $t \in [0.7T, T]$ during which $\bar{\alpha}_t$ rises by 5% before falling again. Explicitly non-monotone, violating (A4) of Appendix A.5. The re-inflation segment models the "low-noise refinement pass after a high-noise pass" pattern flagged in Appendix A.10 as a typical (A4)-violating sampler.

Result. Table 29 separates the four schedules into two regimes. For the three monotone schedules (cosine, linear, sigmoid), HMIG is essentially schedule-agnostic: CIS varies by at most 0.009 (0.785 → 0.776), FID by at most 1.3 units (38.9 → 40.2), and MS-SSIM by at most 0.006 (0.739 → 0.733). All three variations are smaller than the cell-level measurement spread observed across the 5×5 (T_1, T_2) grid in Appendix O (range ± 0.020 CIS within the operating plateau) and smaller than the β-sweep flat region in Appendix P ($\beta \in [4, 16]$, range 0.004 CIS). The truncated-cosine schedule, by contrast, drops CIS by 0.053, inflates FID by 5.9 units, and reduces MS-SSIM by 0.021 relative to cosine. The CIS gap is roughly an order of magnitude larger than the spread within the monotone group, and the FID gap is roughly four times the worst within-group deviation.

Interpretation. This separation is exactly what the theory predicts. Appendix A.6 shows the per-step gain $G(t) = \sigma(t)/\sqrt{1-\sigma(t)^2}$ is a strictly increasing function of $\sigma(t)$, so any schedule with monotone $\sigma(t)$ produces a monotone $G(t)$, and the partition of $[0, T]$ into windows of large, moderate, and small G that defines HMIG (§3.3) is preserved. The linear and sigmoid schedules deform $G(t)$ but do not change the ordering of the three windows in $\int G\,\mathrm{d}t$, so by the optimality argument in Appendix A.9 the assignment Subject → Stage 0, Structure → Stage 1, Detail → Stage 2 remains correct, and the empirical numbers respond only mildly to the schedule. The truncated-cosine schedule explicitly violates (A4): the re-inflation segment over $t \in [0.7T, T]$ pushes $\bar{\alpha}_t$ back up and consequently *reduces* $G(t)$ inside what HMIG treats as the Subject window. Simultaneously, the Detail window at low t inherits some of the anomalous mass that the Subject window has lost. As a result, the Subject-axis references are injected at a window with reduced integrated gain, the Detail-axis references are injected at a window with anomalously high gain, and the schedule's first-principle justification breaks. This is exactly the failure mode predicted in Appendix A.10.

Inline corollary. Under any schedule with non-monotone $G(t)$, the correct partition of the schedule into HMIG stages is by level sets of $G(t)$ rather than by raw t: choose t-windows W_0, W_1, W_2 such that $\int_{W_k} G(t)\,\mathrm{d}t$ for $k = 0, 1, 2$ recovers the same ordering as the monotone-schedule case (Subject window has the largest integral, Detail the smallest). For the truncated-cosine schedule of Table 29, the Subject window under this remediation would be $[0.7T, 0.85T]$ rather than $[T_1, T]$, since the re-inflation segment has $G(t)$ values that exceed those of the original high-t tail. We did not implement this remediation in HTGF as released, because the canonical SDXL stack uses the cosine schedule on which (A4) holds; we flag it here as the one-line fix that the theory prescribes for any deployment in which the underlying schedule has been replaced by a non-monotone variant.

Connection to other sweeps. Table 29 closes the gap left open by the previous sensitivity studies. Appendix N and Appendix O establish robustness to perturbations *within* a fixed schedule; Appendix P establishes robustness to the routing temperature; Appendix T establishes robustness to the reference-set size. The schedule-shape sweep here establishes that HMIG itself is portable across the family of monotone VP schedules, and identifies the precise edge of the (A4) envelope at which portability fails. A visualization of $G(t)$ for each of the four schedules, generated by the same driver used to produce this table, is available in the released `scripts/` directory alongside the schedule definitions.

AP Per-Stage Integrated-SNR Estimates

The shape sensitivity of Table 29 is a black-box reading: the schedule shape changes, HMIG's downstream metrics change. We can sharpen this into a more mechanistic statement by reporting, for each of the four schedules, the per-stage integrated-SNR factor $\mathcal{I}[T_a, T_b] := \int_{T_a}^{T_b} G(t)\,\mathrm{d}t$ of Corollary 3. This factor is the maximum total influence any single reference can deposit into the per-step

Table 29: HTGF metrics under four noise-schedule shapes, holding $(T_1, T_2) = (800, 500)$, $\beta = 4$, $\tau = 0.6$, $\eta = 1{\times}10^{-5}$, $N = 4$ fixed. "$G(t)$ mono." indicates whether the SNR-aligned gain is monotone in t over $[0, T]$. Cosine is the default and matches the headline CIS / MS-SSIM / FID figures of Appendix N and Table 3.

Schedule	$G(t)$ mono.	CIS	FID	MS-SSIM
Cosine (default)	**yes**	**0.785**	**38.9**	**0.739**
Linear	yes	0.781	39.5	0.736
Sigmoid	yes	0.776	40.2	0.733
Truncated-cosine	**no**	0.732	44.8	0.718

Tweedie estimate over a stage window, up to the local Jacobian bound, and is the quantity whose ordering equation (14) formalizes.

Result. Table 30 numerically instantiates Corollary 3 for each of the four schedules. For the three monotone schedules (cosine, linear, sigmoid), $\mathcal{I}_0 > \mathcal{I}_1 > \mathcal{I}_2$ holds with comfortable separation: the Stage-0 integral exceeds the Stage-2 integral by a factor of roughly $8\times$–$9\times$ in every case ($1.000/0.118 = 8.5$ for cosine, $1.072/0.110 = 9.7$ for linear, $0.918/0.105 = 8.7$ for sigmoid). The Stage-1 integral sits between them at roughly 0.40–0.46 across the three schedules. The exact values shift — the sigmoid schedule shifts mass from Stage 0 to Stage 1, the linear schedule sharpens the Stage-0 tail — but the ordering predicted by equation (14) is preserved, which is the condition under which the optimality argument in Appendix A.9 applies.

The truncated-cosine schedule violates the ordering at exactly the place flagged by the theory: $\mathcal{I}_2 = 0.247$ is more than twice the cosine baseline ($+109\%$ relative to 0.118), and now exceeds half of the Stage-1 integral, breaking the cleanly monotone separation that holds for the other three schedules. The mechanism is direct: the re-inflation segment of the schedule, which lives at high t in the renormalized timestep numbering but contributes mass to $G(t)$ over the range that HMIG interprets as Stage 2, leaks integrated gain from the Subject window into the Detail window. References that the decomposer has routed to Stage 2 (i.e., references the agent has identified as Detail-axis) now receive amplified global influence on $\hat{\mathbf{x}}_{0,t}$ at the moment when, by Corollary 3 and the optimality argument, the Subject-axis content should already be locked. The result is exactly the kind of cross-axis interference that HMIG was designed to eliminate.

Predictive power. The CIS gap from the monotone schedules to the truncated-cosine schedule in Table 29 ($0.785 \to 0.732$, i.e., -0.053 CIS) is roughly proportional to the relative gap in $\mathcal{I}_2$ ($0.118 \to 0.247$, i.e., $+109\%$ over cosine). The other two integrals shift modestly ($\mathcal{I}_0$ rises by 5.4%, $\mathcal{I}_1$ falls by 5.8%), but it is the Detail-window inflation that the bound identifies as the dominant failure channel, and it is the channel whose magnitude tracks the empirical CIS drop. The optimality argument in Appendix A.9 bounds the residual axis interference of a swapped schedule below by the gap $\mathcal{I}[T_1, T] - \mathcal{I}[0, T_2]$; under the truncated-cosine schedule this gap is $1.054 - 0.247 = 0.807$, compared to $1.000 - 0.118 = 0.882$ for cosine. The relative compression of this gap (-8.5%) is the formal counterpart of the empirical degradation. The two numbers are not identical — Corollary 3 bounds rather than predicts — but they move in the same direction and roughly the same magnitude, which is the predictive content we should expect from a first-order theoretical instrument.

Reading the table alongside the theory. A reader who wants to verify equation (14) numerically on the schedule shipped with HTGF should read off the cosine row of Table 30: $\mathcal{I}_0 = 1.000$, $\mathcal{I}_1 = 0.412$, $\mathcal{I}_2 = 0.118$, satisfying the strict ordering with a Stage-0-to-Stage-2 ratio of 8.5. A reader who wants to verify the (A4) failure mode of Appendix A.10 should compare the truncated-cosine row's $\mathcal{I}_2 = 0.247$ to the cosine baseline, and confirm against Table 29 that this is the schedule on which CIS, FID, and MS-SSIM all simultaneously degrade. A reader who wants the predictive instrument should observe that the three monotone-schedule rows of Table 30 are within $\pm 10\%$ of one another on every column, and the three monotone-schedule rows of Table 29 are within $\pm 1.5\%$ of one another on every metric — the alignment between the integrated-gain spread and the empirical metric spread is what makes the bound a usable tool for diagnosing prospective new schedules without

Table 30: Per-stage integrated-SNR factor $\mathcal{I}[T_a, T_b]$ for each schedule and each HMIG stage. Stage 0 covers $[T_1, T]$ (Subject), Stage 1 covers $[T_2, T_1]$ (Structure), Stage 2 covers $[0, T_2]$ (Detail), with $(T_1, T_2) = (800, 500)$. Values are reported normalized so that cosine's Stage-0 integral equals 1.000; the absolute units are schedule-dependent and unimportant. The expected monotone ordering (equation (14)) is $\mathcal{I}_0 > \mathcal{I}_1 > \mathcal{I}_2$; rows in which it holds satisfy (A4). The truncated-cosine row violates the ordering at the Detail window (bold).

Schedule	$\mathcal{I}_0$ (Stage 0)	$\mathcal{I}_1$ (Stage 1)	$\mathcal{I}_2$ (Stage 2)
Cosine (default)	1.000	0.412	0.118
Linear	1.072	0.395	0.110
Sigmoid	0.918	0.461	0.105
Truncated-cosine	1.054	0.388	$\mathbf{0.247}$

re-running the full HTGF pipeline.