Where and When Identity Forms: Identity-Vital Attention Redistribution for Training-Free Subject-Driven Generation

Luan Thanh Trinh
LY Corporation
ttrinh@lycorp.co.jp

Kenji Doi
LY Corporation
kedoi@lycorp.co.jp

Atsuki Osanai
LY Corporation
atsuki.osanai@lycorp.co.jp

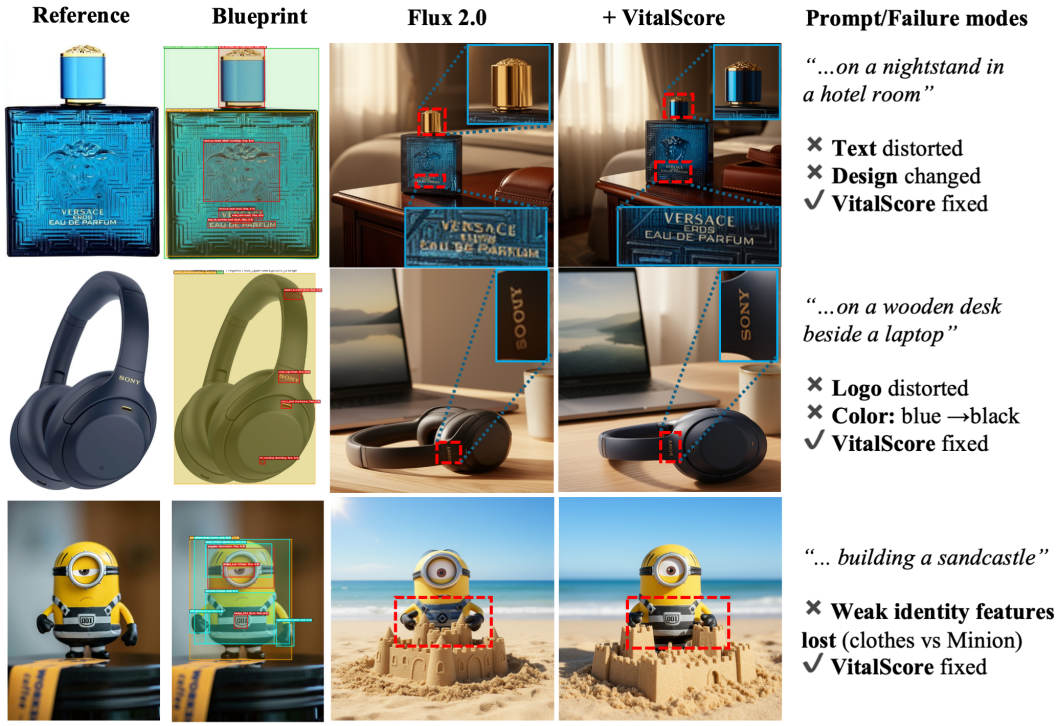


Figure 1: Identity preservation is sparse and structured, not uniform. Global reference conditioning (baseline) coarsely preserves the subject category but loses logos, printed text, textures, and fine structural cues. VitalScore redistributes reference attention to identity-vital layers, timesteps, and spatial regions, recovering these details without parameter updates or extra inference passes. **All pairs share the same initial noise seed for a fair comparison.**

Abstract

Reference-conditioned diffusion transformers for subject-driven generation preserve coarse object identity but routinely lose identity-defining details such as logos, printed text, textures, and small structural cues. We show that this failure traces to a structural mismatch: identity sensitivity is sparse and non-uniform across layers, denoising stages, and spatial regions, yet reference influence is typically applied as a global signal. We introduce **VitalScore**, a training-free method that discovers this identity-vital structure through offline probing and uses it to build a reference-attention redistribution tensor $\Lambda(l, t, s)$ coordinated with a stage-aware prompt schedule $P(t)$. The spatial axis is grounded in a two-phase VLM blueprint calibrated against human identity annotations, ensuring the controller reflects the cues people actually use to judge subject identity. We evaluate across

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

three FLUX-family pipelines (Diptych Prompting, FLUX Kontext, FLUX 2.0) and two non-FLUX backbones (Qwen-Image-Edit, FIBO-Edit) on DreamBench++, where VitalScore improves DINO-v2 by up to 15%, and on a new product identity benchmark targeting fine-grained commercial cues that standard benchmarks overlook, where gains reach 16.5% on the hardest prompt axis—all without parameter updates.

1 Introduction

Subject-driven image generation aims to place a specific visual subject into new scenes while preserving both the subject and the prompt [1–11]. This capability is increasingly practical for product photography, background replacement, and personalized content creation, but current systems often preserve only coarse identity: the generated object remains a sneaker, bottle, watch, or toy, while the details that make it the same subject drift. In commercial settings these details are not cosmetic: **logos, printed text, distinctive textures, small geometric marks, watch faces, and sneaker colorways are often the identity that must survive the edit**. Existing personalization methods can bind a subject through optimization or learned adapters, but the growing demand is for fast, reusable control over frozen diffusion and flow-matching models.

Recent zero-shot and reference-conditioned methods move in this direction by using an input image as visual context rather than training a new subject representation [12–15]. In-context models such as FLUX.1 Kontext [16] and FLUX 2.0 [17] further make reference images a native part of the generation interface through sequence-level conditioning. These systems substantially improve usability, yet they still expose a common failure mode: **reference influence is often applied as a global or weakly structured signal**, even though the identity evidence that matters is local, time-dependent, and architecture-dependent. Increasing reference pressure everywhere can recover some visual detail, but it can also copy irrelevant context, suppress prompt following, or spend attention on regions that do not define the subject.

Our starting point is that **identity preservation should be studied as its own control problem**. Prior work on training-free DiT editing has shown that only a subset of layers may be vital for generic image editing [18]. We ask a different question: where and when does subject identity form? Through offline layer and timestep probing, we find that identity sensitivity is sparse rather than uniform, and that **identity-vital structure is not simply the same as edit-vital structure**. Identity-critical behavior concentrates in particular layers and denoising phases, with early steps setting broad subject evidence and later steps refining fine details. This suggests that the natural control variable is not a single reference strength, but a structured schedule over layer, time, and space.

We introduce **VitalScore**, a training-free conditioning method that turns this discovery into a reference-attention redistribution tensor $\Lambda(l, t, s)$. VitalScore first identifies identity-vital layers and timesteps for a backbone, then uses a two-phase VLM blueprint, calibrated against human-selected identity features and bounding boxes, to identify and localize the visual evidence that should be preserved. The first blueprint phase recognizes identity-critical features that a human evaluator would consider subject-defining, such as logos, text, material patterns, distinctive shapes, and small markings; the second localizes the selected features with tighter spatial regions. These priors are mapped to latent tokens and used to **redistribute reference attention where identity evidence matters**. The same coarse-to-fine view defines a stage-aware prompt schedule $P(t)$, which lets early denoising emphasize scene and layout constraints before later denoising restores the full identity-detail prompt.

We evaluate VitalScore on three FLUX-family subject-driven pipelines, spanning Diptych Prompting, FLUX Kontext, and FLUX 2.0, and on two non-FLUX reference-conditioned DiTs, Qwen-Image-Edit [19] and FIBO-Edit [20], using DreamBench++ as a public personalized-generation benchmark [21] and a new product identity benchmark designed for fine-grained commercial identity. Unlike DreamBench++, which covers broad personalized identity across humans, animals, and general objects, our benchmark stresses product-specific cues: shape, color and material, logos, printed text, texture, design patterns, and physical integration into the scene. On this 240-pair product benchmark, full VitalScore **improves DINO-v2 identity similarity by up to 7.1% overall and by 16.5% on action/interaction prompts**, the hardest prompt axis in our study, while the prompt axis recovers prompt alignment relative to attention-only variants. We pair the benchmark with a prompt-calibrated Gemini 3.1 Pro judge that scores structure/semantic identity, detail fidelity, and aesthetics/integration, and report human-alignment calibration in the appendix.

Concretely, our contributions are: (i) we show that identity preservation in reference-conditioned diffusion transformers is sparse across layers and denoising time, and differs from generic editability; (ii) we introduce VitalScore, a training-free identity-vital conditioning method that coordinates layer, timestep, spatial, and prompt-stage priors; (iii) we propose a human-calibrated two-phase VLM blueprint for selecting identity-critical visual evidence and localizing it for spatial control; and (iv) we validate VitalScore across five backbones, three FLUX-family pipelines and two non-FLUX DiTs, on DreamBench++ and a fine-grained product identity benchmark that measures shape, color/material, logo/text, texture, design pattern, and scene integration beyond coarse concept preservation and prompt following.

2 Related Work

Subject-driven generation Subject-driven generation began with per-subject optimization of tokens, weights, or lightweight modules [1, 7, 2, 8], and later encoder or adapter methods improved reuse by mapping references into visual or textual conditioning spaces [11, 3, 4, 6, 9]. Object-centric customization methods add spatial grounding and reference-aware editing for composition and local detail [22–25]. Recent DiT systems make visual context more native: JEDI and Diptych Prompting use joint or side-by-side image context [12, 13], ICEdit and Insert Anything target instruction following and insertion [14, 15], and FLUX.1 Kontext, Qwen-Image, OmniGen, and FLUX.2 expose reference images through multimodal conditioning [16, 19, 26, 17]. VitalScore is complementary: it studies how an existing reference signal should be allocated across layer, time, and space.

Reference and prompt control in diffusion transformers Reference control is often global or weakly structured. Diptych Prompting increases cross-panel attention from generated tokens to reference tokens [13]; other training-free DiT customization methods exploit in-context behavior, implicit multimodal conditioning, or latent-space manipulation to avoid per-subject training [27–30]. LatentUnfold tiles the reference in a mosaic grid and aggregates multi-scale attention maps at fixed layers for coarse-to-fine identity reinforcement [31]; FreeGraftor grafts reference key-value pairs at semantically matched spatial positions via cycle-consistent feature correspondences [32]. These methods show that pretrained DiTs contain usable reference-transfer capacity, but their schedules are fixed by design (preset layers, mosaic scales, or matching positions) rather than discovered from identity sensitivity; since we find that identity-vital structure differs sharply across backbones, a fixed schedule cannot be expected to transfer. Prompt and guidance schedules provide a complementary axis: limited-interval guidance restricts guidance to useful denoising ranges [33], and stage-aware prompting uses denoising-stage proxy prompts to improve semantic alignment [34]. VitalScore couples this coarse-to-fine prompt staging with identity-vital reference-attention redistribution.

Layer attribution and probing Recent work also probes internal DiT structure for editing and control. StableFlow is the closest methodological prior: it identifies vital layers for training-free image editing by measuring how layer bypass affects edit behavior, then injects features only at selected layers [18]. Other MMDiT editing and inference-time methods study multimodal interactions, denoising dynamics, or context handling for editability and prompt control [35–37]. Our target is different: we ask which layers and timesteps are causally important for retaining subject identity under reference conditioning. Empirically, identity-vital layers have low overlap with StableFlow-style edit-vital layers, and some edit-vital late layers are identity-harmful in our setting. VitalScore therefore treats probing as task-specific discovery for identity preservation, then extends it with temporal, spatial, and prompt-stage axes.

3 Identity-Vital Discovery

Probe protocol We ask where reference-conditioned DiTs are causally sensitive to identity. For layer probing, we follow the residual-bypass logic of StableFlow [18], but change the target from edit quality to identity retention. For each layer l, we skip its residual contribution and measure

$$V_{\mathrm{id}}(l) = \mathrm{DINO}(I_r, G_{\mathrm{full}}) - \mathrm{DINO}(I_r, G_{\mathrm{skip}_l}),$$

using DINO-v2 ViT-L/14 cosine similarity between the reference and generated image. For timestep probing, we decode the predicted clean image $\hat{x}_0$ at each denoising step and score it against the reference. The same protocol is applied to Diptych Prompting, FLUX.1 Kontext, and FLUX 2.0 on an independent discovery probe set of **60 randomly sampled object/product reference-prompt pairs, disjoint from the product identity benchmark used for final evaluation**; full run counts and settings are reported in Appendix A.

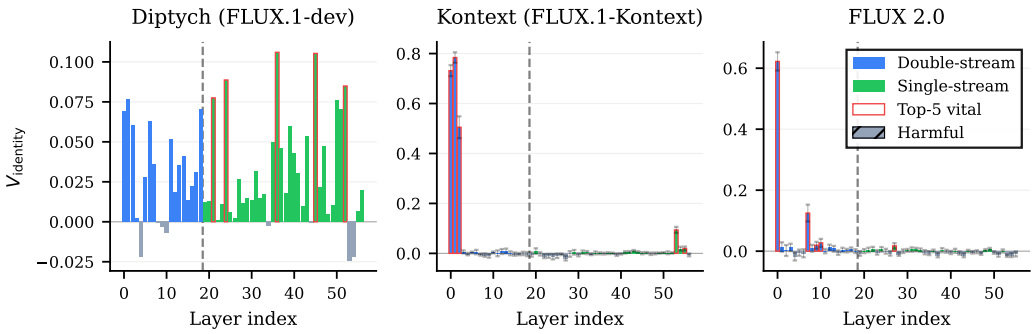


Figure 2: Identity-vital layer profiles for three FLUX-family backbones. Each panel shows $V_{\text{id}}(l)$ per layer; panels have independent y-axis scales. Blue = double-stream layers, green = single-stream. Bars with a red outline are the top-5 identity-vital layers; hatched gray bars are identity-harmful.

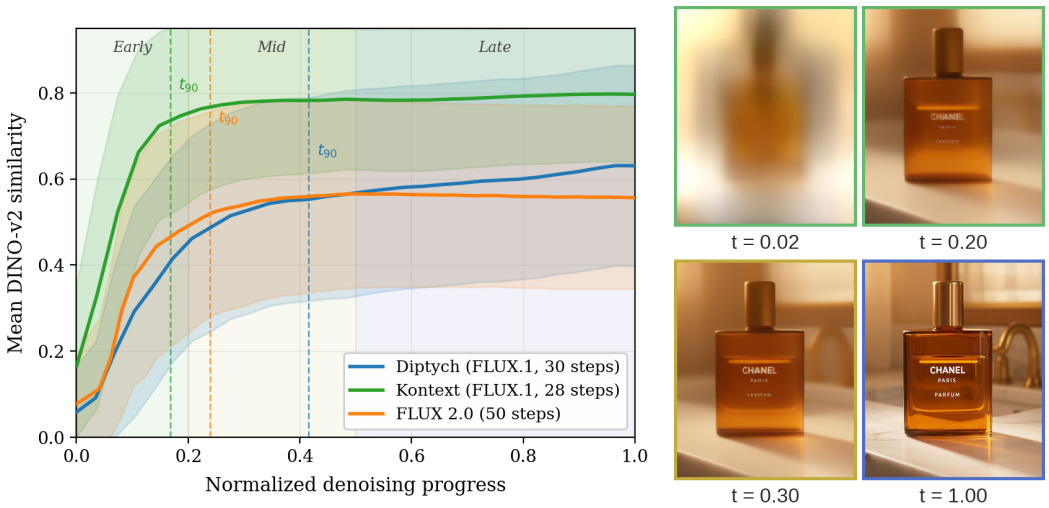


Figure 3: Temporal identity formation and representative visual crystallization. Left: mean DINO-v2 similarity to the reference image over normalized denoising progress for 60 product pairs; vertical dashed lines mark t_{90} for each backbone. Right: representative $\hat{x}_0$ snapshots over denoising for one example.

Layer structure **Identity-vital layers are sparse and architecture-specific**. As shown in Figure 2, FLUX.1 Kontext concentrates identity sensitivity in the first few layers, FLUX 2.0 is dominated by layer 0 with a smaller secondary peak around layer 7, and Diptych Prompting spreads identity impact across mid-to-late single-stream layers. Layer 0 is consistently important, but the distribution around it differs enough that a single transferred layer set would be brittle.

Identity-vital structure also differs from generic editability. On Diptych, where we can compare with StableFlow-style edit-vital layers, the top-10 identity and edit sets have Jaccard overlap 0.176 and Pearson correlation $r = 0.35$. Some late edit-vital layers are identity-harmful: bypassing them improves DINO-v2 similarity to the reference. Thus identity preservation is not simply editing under another name; it requires task-specific discovery. Full layer sets are listed in Appendix B.

Temporal structure **Identity is also temporally non-uniform**. Figure 3 shows that all three models gain identity rapidly in early denoising, enter a quieter middle phase, and then refine details late. The timing is model-dependent: Kontext reaches t_{90} at 16.8% of denoising, FLUX 2.0 at 23.9%, and Diptych at 41.6%. The representative denoising snapshots illustrate the same coarse-to-fine pattern: broad object evidence appears early, while text, logos, and reflective details sharpen later.

This temporal structure supports a staged intervention policy. Early steps are useful for coarse subject evidence and layout, the mid-denoising plateau contributes comparatively little new identity signal, and late steps remain important for fine-detail crystallization. The schedule should therefore depend on denoising phase rather than applying the same reference pressure at every step.

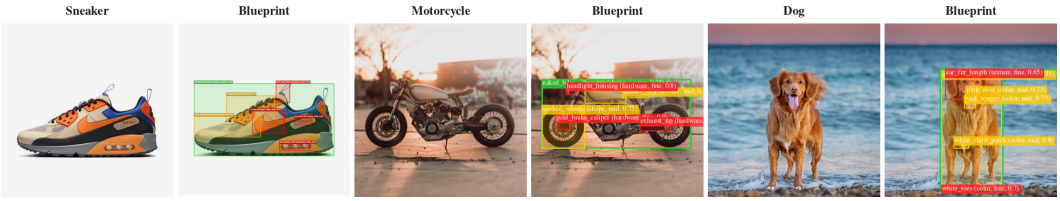


Figure 4: Human-aligned VLM blueprint examples across three subject categories (sneaker, motorcycle, dog). Phase B bounding boxes are overlaid on each reference image, color-coded by granularity (green = coarse, yellow = mid, red = fine).

Spatial structure **Identity evidence is spatially sparse**. A sneaker logo, printed wordmark, watch-face marker, or texture patch may occupy only a small fraction of the image while carrying the evidence people use to recognize the subject. Uniform spatial modulation therefore spends reference attention on many non-diagnostic tokens and can cause attention bleeding, where reference details spread into regions that should be controlled by the prompt.

We approximate these sparse regions with a human-aligned VLM blueprint. The VLM receives the reference image and target prompt and outputs identity-critical regions with normalized boxes, semantic types, importance scores, and coarse/mid/fine granularity labels. Its prompt was iteratively tuned against human annotations of which features are identity-defining and where they are located, so the blueprint is not a generic saliency map but a compact estimate of human-recognizable identity evidence. We provide the VLM prompt, annotation protocol, and alignment details in Appendix F.

Blueprint boxes are mapped to latent-space token indices via normalized coordinates. Each token receives the maximum importance of the regions covering it, producing a spatial map $s(x, y)$ that concentrates modulation on identity-carrying tokens. An example blueprint is shown in Figure 4.

From identity-vital structure to VitalScore These findings point to the same conclusion: **globally uniform reference control is mismatched to identity preservation**. Identity sensitivity is concentrated in a small set of layers, changes across denoising time, and depends on sparse spatial evidence. The natural intervention is therefore a structured reference-attention tensor $\Lambda(l, t, s)$ that redistributes attention where identity forms. The same coarse-to-fine temporal prior also motivates a companion prompt schedule $P(t)$, introduced formally with VitalScore in Section 4.

4 VitalScore Method

Baseline and lift. Let I_r be a reference image and let I_g denote the generated target image. In a reference-conditioned diffusion transformer, target image tokens query reference tokens through an attention sub-block. We write this target-to-reference attention at layer l and denoising step t as

$$A^{l,t}_{g\leftarrow r}(q, s) = \mathrm{softmax}_s \left(Q^{l,t}_g(q) K^{l,t}_r(s)^\top / \sqrt{d} \right), \tag{1}$$

where q indexes a target token and s indexes a reference token. A common control view is to apply a single reference strength λ_0 to this whole sub-block. This is simple, but it assumes that all layers, times, and reference locations should carry the same identity pressure. **VitalScore replaces this global control with a structured schedule:**

$$\widetilde{A}^{l,t}_{g\leftarrow r}(q, s) = \Lambda(l, t, s) A^{l,t}_{g\leftarrow r}(q, s), \tag{2}$$

where $\Lambda(l, t, s)$ is derived from identity-vital layer discovery, timestep discovery, and a VLM spatial blueprint. Because Λ indexes only reference (key) tokens, the post-softmax scaling folds exactly into the value rows, $\sum_s \Lambda(l, t, s) A(q, s) V(s) = \sum_s A(q, s) \left(\Lambda(l, t, s) V(s)\right)$, so modulated attention runs through standard fused kernels (SDPA/FlashAttention) with 1.07–1.11$\times$ wall-clock overhead (Appendix J). The same coarse-to-fine prior also defines a prompt schedule $P(t)$. The attention schedule preserves identity-critical visual evidence, while the prompt schedule avoids forcing identity-heavy text before the model has established the requested scene.

Layer and temporal axes. Let $\mathcal{L}_{\mathrm{id}}$ be the set of identity-vital layers found by the offline probing protocol in Section 3: the ten layers with the largest $V_{\mathrm{id}}(l)$ for each backbone, listed in Appendix Table 7 (the red outlines in Figure 2 mark the top five of the same ranking). VitalScore uses this set as a binary gate $g_l = \mathbf{1}[l \in \mathcal{L}_{\mathrm{id}}]$. When $g_l = 0$, the layer follows the backbone's default reference

behavior. When $g_l = 1$, VitalScore applies the spatial and temporal schedule in Eq. 5. This keeps non-vital layers close to the original generator and concentrates intervention in the layers where identity retention was empirically sensitive. Timestep discovery suggests a **coarse-to-fine identity schedule**: early steps establish broad structure, mid steps contribute relatively little identity change, and late steps refine fine visual evidence. We encode this with granularity-specific activation windows. Each reference token carries exactly one spatial granularity label $\gamma(s) \in \{\text{coarse}, \text{mid}, \text{fine}\}$ from the blueprint below; let $h(t, \gamma) = \mathbf{1}[\rho(t) \in W_\gamma]$, with $\rho(t) = t/(T-1)$. Here T is the number of denoising steps and W_γ is the active denoising interval for that granularity. Coarse regions use early windows, fine regions use later windows, and mid-level regions cover the transition. Labels are exclusive per token, but the windows overlap by design so that activation hands off smoothly across denoising. **The boundaries are derived, not tuned**: they are read off the backbone's formation curve (Figure 3) on the 60-pair discovery probe set (coarse ends near the early-formation knee, fine begins near it, mid bridges the transition) and are never iterated against evaluation metrics. For the three FLUX backbones the boundaries fall in the flat region of all three curves, so they share one set (Appendix Table 17); results are insensitive to shifting them (Section 5.3).

Spatial axis: two-phase VLM blueprints. VitalScore approximates spatial identity evidence with a **prompt-calibrated two-phase VLM blueprint**. Phase A recognizes identity-critical features in the reference, such as logos, printed text, distinctive markings, textures, hardware, and characteristic shape cues. Phase B receives the frozen Phase A feature list and localizes only those selected features with normalized bounding boxes. The resulting blueprint contains regions $r_i = (b_i, a_i, \gamma_i, c_i)$, where $b_i = [x_1, y_1, x_2, y_2]$ is a normalized box, $a_i \in [0, 1]$ is an importance score, γ_i is a granularity label, and c_i is a semantic type such as logo, text, shape, texture, pattern, or hardware.

Each box is mapped to reference latent tokens by the VAE grid. If the reference latent grid has height H and width W, a box b_i covers

$$\mathcal{S}_i = \{(u, v) : \lfloor y_1 H \rfloor \le u < \lceil y_2 H \rceil,\ \lfloor x_1 W \rfloor \le v < \lceil x_2 W \rceil\}. \tag{3}$$

Overlapping regions keep the highest-importance assignment. The spatial map M_s and granularity map $\gamma(s)$ are therefore generated automatically from the reference and prompt; since the blueprint is only a set of boxes, a user may also draw them by hand. **Blueprints are validated before use**: boxes violating the schema (coordinate range, ordering, area constraints) or marked occluded are dropped, each surviving region's effect is scaled by its importance a_i, and if no valid blueprint remains, $M_s \equiv \lambda_0$ and VitalScore reduces exactly to the baseline. Separating feature selection from localization yields tighter boxes, less attention bleeding, and higher downstream identity than a single-pass blueprint (Appendix Table 12).

Redistribution principle. VitalScore redistributes reference attention rather than simply increasing total reference pressure. Let $R(s)$ be the raw spatial strength:

$$\begin{aligned} R(s) &= \lambda_0 + \max_{i : s \in \mathcal{S}_i} a_i(\lambda_{\max} - \lambda_0), \\ M_s(s) &= \lambda_0 \frac{R(s)}{\frac{1}{|\mathcal{S}|}\sum_{s' \in \mathcal{S}} R(s')}. \end{aligned} \tag{4}$$

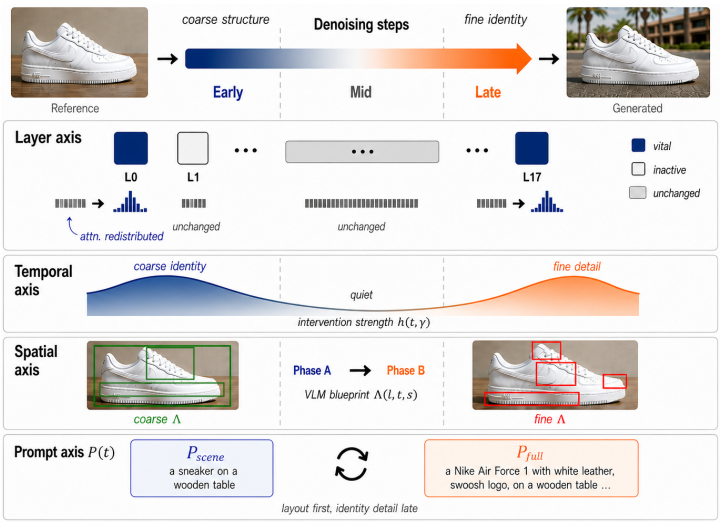
with $R(s) = \lambda_0$ for tokens outside all blueprint regions. We normalize this map to preserve the baseline mean reference strength. Thus $\frac{1}{|\mathcal{S}|}\sum_s M_s(s) = \lambda_0$: **identity regions receive more reference attention, while less diagnostic regions receive less**, without raising the average attention budget. Combining the layer, time, and spatial axes gives

$$\Lambda(l, t, s) = \lambda_0 + g_l\, h(t, \gamma(s))\, (M_s(s) - \lambda_0)\,. \tag{5}$$

This form makes the division of labor explicit: the blueprint decides where identity evidence lives, temporal discovery decides when each granularity should be active, and layer discovery decides where the generator is sensitive enough for intervention to matter.

Prompt axis: stage-aware prompt conditioning. The attention schedule controls visual evidence, but the text pathway also changes the identity/prompt tradeoff. A single identity-rich prompt can improve subject fidelity while crowding out scene constraints early in denoising. VitalScore therefore uses **stage-aware prompt conditioning as a coupled prompt axis**:

$$P(t) = \begin{cases} P_{\text{scene}}, & \rho(t) < \tau_p, \\ P_{\text{full}}, & \rho(t) \ge \tau_p, \end{cases} \tag{6}$$



Input: I_r, prompt p, frozen backbone, discovered $\mathcal{L}_{\text{id}}$.
1. Blueprint: run the two-phase VLM prompt to obtain identity regions $r_i = (b_i, a_i, \gamma_i, c_i)$.
2. Map: project boxes b_i to latent tokens, producing $M_s(s)$ and $\gamma(s)$.
3. Schedule: build $\Lambda(l, t, s)$ from layer gates, temporal windows, and M_s; build $P(t)$ from scene/full prompts.
4. Generate: denoise once with the frozen model, applying Λ to reference attention and selecting the active prompt embedding $P(t)$.
Output: generated image I_g.

Figure 5: VitalScore inference and overview. **(a)** Overview of the discovery-guided conditioning pipeline. Offline discovery is amortized per backbone; inference adds blueprint generation and lightweight scheduling without trainable parameters, latent optimization, or extra denoising passes. **(b)** Simplified inference: builds the blueprint, maps identity regions to latent tokens, constructs $\Lambda(l, t, s)$ and $P(t)$, and runs one frozen denoising pass.

where P_{scene} describes the requested scene, action, lighting, and composition with only a generic subject noun, and P_{full} restores the full identity-detail prompt. This prompt axis is not a standalone identity method; it is paired with Eq. 5 so that early steps can form layout while identity-vital attention continues to carry reference evidence, and later steps can use identity text when fine details are being resolved. We set $\tau_p = 0.10$ by the same formation-curve logic: the switch falls after about three steps, when every backbone has committed global layout (33–83% of final identity; Appendix Table 6), and before the earliest formation knee (Kontext, $t_{90} = 0.168$). Stage prompts are pre-encoded and swapped during denoising (Appendix I); we treat prompt-axis gains conservatively for architectures that commit strongly at the first denoising step.

Algorithm and overview are shown in Figure 5.

5 Experiments

5.1 Setup

Pipelines. We evaluate on Diptych Prompting [13], FLUX.1 Kontext [16], and FLUX 2.0 [17] — three FLUX-family subject-driven pipelines covering in-context, native multimodal, and second-generation reference conditioning. To test generality beyond FLUX, we apply the full pipeline (discovery, blueprint, $\Lambda(l, t, s)$) to two non-FLUX reference-conditioned DiTs: Qwen-Image-Edit [19], a pure double-stream MMDiT, and FIBO-Edit [20], a dual-stream editing DiT with an LLM text encoder; discovery is re-run per backbone ($\sim$16 GPU-h, offline, once). All runs use fixed pipeline settings with **no parameter updates or per-image optimization**. To ensure a fair comparison, the **baseline and VitalScore variants use the same initial noise seed for each pair**, so metric differences reflect conditioning rather than sampling luck. Additional gains from seed selection are discussed in Appendix I.

DreamBench++. We use DreamBench++ [21] to evaluate 520 personalized-generation pairs across humans, animals, and objects. Metrics include DINO-v2, CLIP-T, VLM concept preservation (CP) and prompt following (PF) as secondary signals validated against human judgment [21].

Product identity benchmark. Commercial products are an ideal stress test for fine-detail identity: their identity is defined by small but highly verifiable cues — brand logos, printed text and model numbers, surface textures, stitching and material finish, and characteristic design patterns — that a human can immediately check but that diffusion models routinely lose. DreamBench++ evaluates broad concept-level identity across diverse subject types and does not target these cues. We therefore introduce a 240-pair product benchmark (60 reference products $\times$ 4 prompt axes) spanning ten categories: Food & Beverage, Kitchen Appliances, Watches, Footwear, Home & Furniture, Bags

& Apparel, Perfume & Beauty, Electronics & Audio, Sports & Outdoor, and Toys & Collectibles. Prompt axes are scene context, lighting, camera composition, and action/interaction; the last produces the largest identity drift because the subject must adapt its pose or interaction to the scene while all fine-detail cues must remain intact. Full benchmark construction details, category breakdown, and VLM judge prompt are in Appendix G.

VLM judge. Product-benchmark outputs are evaluated by a prompt-calibrated Gemini 3.1 Pro judge [38] scoring structure/semantic identity, detail fidelity, and aesthetics/integration, each on a five-level ordinal scale. The judge prompt was iteratively refined against human annotations of identity features and bounding boxes from the same product set used for blueprint calibration; judge-human agreement is in Appendix Table 13.

Table 1: Main automatic results. VS denotes VitalScore; Sem./Det./Aes. are human-aligned VLM scores. Bold marks the better value within each pipeline. Qwen-Edit (Qwen-Image-Edit) and FIBO-Edit are non-FLUX backbones, evaluated on DreamBench++.

(a) DreamBench++

Pipe	Meth.	DINO	CLIP-T	CP	PF
Diptych	Base	0.594	**0.273**	0.650	0.865
	+VS	**0.683**	0.271	**0.722**	**0.899**
Kontext	Base	0.668	0.272	0.665	0.862
	+VS	**0.759**	**0.274**	**0.709**	**0.901**
FLUX 2.0	Base	0.585	0.283	0.618	0.921
	+VS	**0.620**	**0.289**	**0.646**	**0.949**
Qwen-Edit	Base	0.655	0.270	0.676	0.892
	+VS	**0.708**	**0.271**	**0.692**	**0.914**
FIBO-Edit	Base	0.891	0.227	0.814	0.630
	+VS	**0.923**	**0.231**	**0.839**	**0.637**

(b) Product Identity Benchmark

Pipe	Meth.	DINO	CLIP-T	Sem.	Det.	Aes.
Diptych	Base	0.674	**0.237**	3.18	2.02	4.38
	+VS	**0.701**	0.236	**4.13**	**2.08**	**4.48**
Kontext	Base	0.794	0.233	4.63	4.29	4.76
	+VS	**0.850**	**0.234**	**4.86**	**4.55**	**4.83**
FLUX 2.0	Base	0.710	0.233	4.69	4.12	4.45
	+VS	**0.745**	**0.237**	**4.83**	**4.41**	**4.88**

Table 2: Comparison with other training-free methods on DreamBench++.

Method	Backbone	DINO-v2	CLIP-T	CP	PF
λ-ECLIPSE [6]	Kan-v2.2	0.687	0.232	0.693	0.605
MS-Diffusion [9]	SDXL	0.642	0.239	0.563	0.610
IP-Adapter [11]	FLUX.1-dev	0.497	0.280	0.480	0.947
FreeGraftor [32]	FLUX.1-dev	0.615	0.275	0.600	0.901
LatentUnfold [31]	FLUX.1-dev	0.472	0.247	0.508	0.682
Diptych + VitalScore	FLUX.1-dev	0.683	0.271	**0.722**	0.899
Kontext + VitalScore	FLUX.1-Kontext-dev	**0.759**	0.274	0.709	0.901
FLUX 2.0 + VitalScore	FLUX.2-dev	0.620	**0.289**	0.646	**0.949**

Table 3: Blinded human preference (%). Baselines are Diptych, Kontext, and FLUX 2.0; for each trial, raters saw the reference plus randomly selected baseline and VitalScore outputs.

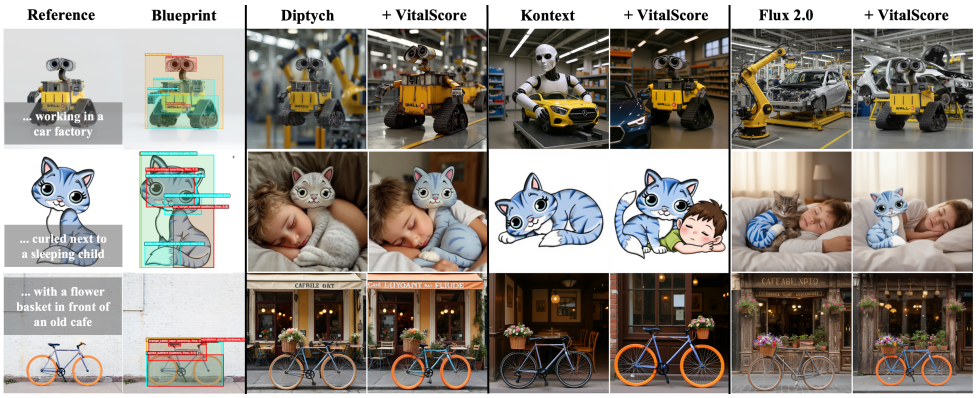
Criterion	VitalScore	Tie	Baseline
Identity	77.8	4.2	18.0
Detail	82.1	3.9	14.0
Prompt	55.9	2.5	41.6
Overall	80.5	5.1	14.4

5.2 Qualitative and quantitative comparisons

Table 1 shows the core result: **VitalScore improves identity preservation across all three FLUX-family pipelines**, on both DreamBench++ and the product identity benchmark, while keeping prompt alignment stable or improved. The gains are largest where identity is most fragile: Diptych and Kontext improve by 15.0% and 13.6% DINO-v2 on DreamBench++, and the product benchmark shows consistent gains in semantic identity, detail fidelity, and aesthetics.

On the product benchmark, the VLM scores are central: Appendix G shows the judge aligns with human ratings on small commercial cues—logos, text, surface patterns, material finish, and scene integration—that coarse similarity can miss. Table 1 shows VitalScore improves every product semantic/detail/aesthetic score across all backbones, including detail fidelity gains of 3.0%, 6.1%, and 7.0% for Diptych, Kontext, and FLUX 2.0, respectively.

The same pattern holds beyond FLUX: VitalScore improves DINO-v2 by 8.1% on Qwen-Image-Edit and 3.6% on FIBO-Edit, with CLIP-T, CP, and PF stable or improved. Discovery finds very different structure on each: Qwen-Image-Edit forms identity late ($t_{90} = 0.436$), whereas FIBO-Edit commits almost at initialization ($t_{90} = 0.067$), and both have their own vital layer sets (Appendix C). FIBO-Edit's high baseline (0.891) leaves little headroom, so its gain mainly tests the protocol's adaptivity. What transfers is the procedure (probe first, then intervene), not any FLUX-specific constant; class-conditional DiTs (e.g., DiT-XL) and models without a native reference interface (e.g., SD3) fall outside this reference-conditioned setting.



(a) Results on DreamBench++



(b) Results on the product identity benchmark

Figure 6: Qualitative comparison across Diptych, FLUX Kontext, and FLUX 2.0 on DreamBench++ (a) and the product identity benchmark (b). **Baseline and VitalScore variants use the same initial noise seed for each pair for a fair comparison.**

Table 2 shows that this is not just a weak-baseline artifact. Plain FLUX-family baselines trail the strongest external method on DreamBench++ identity, but **VitalScore moves them into the top tier**: Kontext + VitalScore is best on DINO-v2, Diptych + VitalScore matches the strongest external identity baseline while giving higher PF, and FLUX 2.0 + VitalScore gives the best CLIP-T/PF tradeoff. Human preference in Table 3 mirrors this pattern, with **raters strongly favoring VitalScore** for identity, detail, and overall quality.

The qualitative examples in Figure 6 make the metric gains concrete. Baselines often preserve the coarse subject category but **lose the evidence people use to recognize the instance**: logos, printed text, color layouts, textures, and small structural marks. **VitalScore recovers these cues without any training**, which is exactly the behavior needed for product identity and background replacement.

5.3 Ablation studies

Table 4: Ablation and prompt-axis analysis on the product benchmark with FLUX 2.0. Left: component and blueprint-model ablations. Right: qualitative prompt-axis examples comparing VitalScore without the prompt schedule against full VitalScore.

(a) Component ablation		
Configuration	**DINO-v2**	**CLIP-T**
Baseline	0.710	0.233
Spatial only (all layers)	0.691	0.225
+ Vital layers only	0.739	0.231
+ Temporal	0.743	0.231
+ Prompt schedule	**0.745**	**0.237**
(b) Blueprint model		
VLM	**DINO-v2**	**CLIP-T**
Qwen3.5 4B	0.718	0.232
Qwen3.5 9B	0.745	0.237
Gemini 3.1 Pro	**0.748**	**0.238**

(c) Prompt-axis analysis



Discovery-to-method ablation and blueprint quality Table 4 shows that VitalScore's gain is not from simply adding a VLM mask. **Blindly applying spatial modulation to all layers hurts**: DINO-v2 drops from 0.710 to 0.691, confirming that ungated reference pressure can damage identity. The key step is discovery: **applying the same spatial signal only at identity-vital layers improves performance**, raising DINO-v2 to 0.739, already above baseline. Temporal scheduling then gives a smaller but consistent identity gain, and the prompt schedule produces the best identity/prompt tradeoff, reaching 0.745 DINO-v2 while recovering CLIP-T to 0.237. Appendix Table 8 shows the same design story for Diptych and Kontext. The blueprint-model ablation shows that stronger VLMs produce slightly better blueprints and results: Gemini 3.1 Pro is best at 0.748 DINO-v2, but Qwen3.5-9B is close at 0.745, giving a stronger cost-performance point for local deployment.

Robustness to windows and blueprint errors We test whether the gains hinge on exact boundaries or on blueprint accuracy (Kontext, 130-pair DreamBench++ subset; Appendix E). Shifting every window boundary by ± 0.05 or ± 0.10, or re-deriving the windows from Kontext's own t_{90}, changes DINO-v2 by at most 0.0009 against a +0.088 VitalScore gain, and no variant differs significantly from the paper windows (Wilcoxon signed-rank, all $p \geq 0.11$). **Blueprint errors degrade gracefully**: box jitter of $\pm 10/\pm 20$ keeps 91%/81% of the gain and dropping half the features keeps 64%, whereas fully random boxes fall below baseline, confirming that the spatial axis carries real signal. All main results already include naturally occurring blueprint errors (feature F1 0.87, box IoU 0.61 against human annotations; Appendix Table 11).

Prompt-axis analysis Prompt-axis analysis shows that reference conditioning can expose a conflict between the prompt concept, the reference image, and the model's pretrained visual prior. For example, when a prompt asks for a Minion performing a new action, the model may either follow the high-prior Minion concept from pretraining, including the canonical outfit, or preserve the reference identity while missing the requested action. VitalScore shifts the model toward the reference image and improves identity fidelity, but this can make prompt concepts harder to satisfy when the prompt requires action, pose, or semantic changes. **The stage-aware prompt schedule reduces this burden** by letting early denoising establish scene semantics and layout before restoring detailed identity constraints in later steps. Qualitative examples in Table 4(c), quantitative results in Table 4(a), and the per-axis breakdown in Figure 7 show that prompt scheduling recovers CLIP-T while the largest identity gains reach 16.5% on action/interaction prompts, the hardest axis.

6 Conclusion

We presented VitalScore, a training-free approach for subject-driven generation based on the observation that **identity preservation is structured rather than uniform**. By probing identity-vital layers and timesteps, localizing human-recognizable identity evidence with a two-phase VLM blueprint, and redistributing reference attention through $\Lambda(l, t, s)$ with a staged prompt schedule $P(t)$, VitalScore improves fine-detail identity across Diptych Prompting, FLUX Kontext, and FLUX 2.0 on DreamBench++ and a product identity benchmark, and transfers to the non-FLUX Qwen-Image-Edit and FIBO-Edit. More broadly, our results show that identity retention is not the same control problem as generic editability: effective frozen-model control should **first discover where identity forms, then intervene selectively**. Future work can strengthen blueprint localization, extend the setting to multiple subjects, and carry identity-vital conditioning into video and other multimodal generation tasks.

Limitations VitalScore needs one offline discovery run per backbone ($\sim$16 GPU-h), because identity-vital layer sets do not transfer across architectures. It targets reference-conditioned DiTs; class-conditional models and models without a reference interface are out of scope. The spatial axis depends on blueprint quality: realistic errors keep most of the gain and invalid blueprints revert to baseline, but grossly wrong boxes can fall below baseline. The prompt schedule has a smaller effect on architectures that commit strongly at the first step (e.g., FLUX 2.0). Non-FLUX backbones are evaluated on DreamBench++ only, and multi-subject and video settings remain untested.

Acknowledgments and Disclosure of Funding

This work was conducted as part of the authors' employment at LY Corporation and received no external funding. The authors declare no competing interests.

References

[1] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. URL https://arxiv.org/abs/2208.01618.

[2] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2212.04488.

[3] Dongxu Li, Junnan Li, and Steven Hoi. BLIP-Diffusion: Pre-trained subject representation for controllable text-to-image generation and editing, 2024.

[4] Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning. In *ACM SIGGRAPH Conference Papers*, 2024.

[5] Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. Kosmos-G: Generating images in context with multimodal large language models. In *International Conference on Learning Representations*, 2024.

[6] Maitreya Patel, Sangmin Jung, Chitta Baral, and Yezhou Yang. λ-ECLIPSE: Multi-concept personalized text-to-image diffusion models by leveraging CLIP latent space, 2024. URL https://arxiv.org/abs/2402.05195.

[7] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2208.12242.

[8] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. P+: Extended textual conditioning in text-to-image generation, 2023. URL https://arxiv.org/abs/2303.09522.

[9] X. Wang, Siming Fu, Qihan Huang, Wanggui He, and Hao Jiang. MS-Diffusion: Multi-subject zero-shot image personalization with layout guidance, 2024. URL https://arxiv.org/abs/2406.07209.

[10] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. ELITE: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[11] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2308.06721.

[12] Yu Zeng, Vishal M. Patel, Haochen Wang, Xun Huang, Ting-Chun Wang, Ming-Yu Liu, and Yogesh Balaji. JEDI: Joint-image diffusion models for finetuning-free personalized text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[13] Chaehun Shin, Jooyoung Choi, Heeseung Kim, and Sungroh Yoon. Large-scale text-to-image model with inpainting is a zero-shot subject-driven image generator, 2025. URL https://arxiv.org/abs/2411.15466.

[14] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer, 2025. URL https://arxiv.org/abs/2504.20690.

[15] Wensong Song, Hong Jiang, Zongxing Yang, Ruijie Quan, and Yi Yang. Insert anything: Image insertion via in-context editing in DiT, 2025. URL https://arxiv.org/abs/2504.15009.

[16] Black Forest Labs. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

[17] Black Forest Labs. FLUX.2: Frontier visual intelligence. https://bfl.ai/blog/flux-2, 2025. Technical blog post.

[18] Omri Avrahami, Kfir Aberman, Or Patashnik, Ohad Fried, Egor Nemchinov, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing, 2025. URL https://arxiv.org/abs/2411.14430.

[19] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

[20] Eyal Gutflaish, Eliran Kachlon, Hezi Zisman, Tal Hacham, Nimrod Sarid, Alexander Visheratin, Saar Huberman, Gal Davidi, Guy Bukchin, Kfir Goldberg, et al. Generating an image from 1,000 words: Enhancing text-to-image with structured captions. *arXiv preprint arXiv:2511.06876*, 2025.

[21] Yuang Peng, Yuxin Cui, Haomiao Tang, Zekun Qi, Runpei Dong, Jing Bai, Chunrui Han, Zheng Ge, Xiangyu Zhang, and Shu-Tao Xia. Dreambench++: A human-aligned benchmark for personalized image generation. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2406.16855.

[22] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2307.09481.

[23] Lingjie Kong, Kai Wu, Xiaobin Hu, Wenhui Han, Jinlong Peng, Chengming Xu, Donghao Luo, Mengtian Li, Jiangning Zhang, Chengjie Wang, and Yanwei Fu. Custany: Customizing anything from a single example, 2024. URL https://arxiv.org/abs/2406.11643.

[24] Zhexiao Xiong, Wei Xiong, Jing Shi, He Zhang, Yizhi Song, and Nathan Jacobs. Groundingbooth: Grounding text-to-image customization, 2024. URL https://arxiv.org/abs/2409.08520.

[25] Daniel Winter, Asaf Shul, Matan Cohen, Dana Berman, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. Objectmate: A recurrence prior for object insertion and subject-driven generation, 2024. URL https://arxiv.org/abs/2412.08645.

[26] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation, 2024. URL https://arxiv.org/abs/2409.11340.

[27] Haoran Feng, Zehuan Huang, Lin Li, Hairong Lv, and Lu Sheng. Personalize anything for free with diffusion transformer, 2025. URL https://arxiv.org/abs/2503.12590.

[28] Teng-Fang Hsiao, Bo-Kai Ruan, Yi-Lun Wu, Tzu-Ling Lin, and Hong-Han Shuai. TF-TI2I: Training-free text-and-image-to-image generation via multi-modal implicit-context learning in text-to-image models, 2025. URL https://arxiv.org/abs/2503.15283.

[29] Yu Xu, Fan Tang, You Wu, Lin Gao, Oliver Deussen, Hongbin Yan, Jintao Li, Juan Cao, and Tong-Yee Lee. In-context brush: Zero-shot customized subject insertion with context-aware latent space manipulation, 2025. URL https://arxiv.org/abs/2505.20271.

[30] Yanbing Zhang, Zhe Wang, Qin Zhou, and Mengping Yang. Freecus: Free lunch subject-driven customization in diffusion transformers, 2025. URL https://arxiv.org/abs/2507.15249.

[31] Hao Kang, Stathi Fotiadis, Liming Jiang, Qing Yan, Yumin Jia, Zichuan Liu, Min Jin Chong, and Xin Lu. Flux already knows – activating subject-driven image generation without training, 2025. URL https://arxiv.org/abs/2504.11478.

[32] Zebin Yao, Lei Ren, Huixing Jiang, Wei Chen, Xiaojie Wang, Ruifan Li, and Fangxiang Feng. Freegraftor: Training-free cross-image feature grafting for subject-driven text-to-image generation, 2025. URL https://arxiv.org/abs/2504.15958.

[33] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models, 2024. URL https://arxiv.org/abs/2404.07724.

[34] Saar Huberman, Or Patashnik, Omer Dahary, Ron Mokady, and Daniel Cohen-Or. Image generation from contextually-contradictory prompts, 2026. URL https://arxiv.org/abs/2506.01929.

[35] Binglei Li, Mengping Yang, Zhiyu Tan, Junping Zhang, and Hao Li. Diff-aid: Inference-time adaptive interaction denoising for rectified text-to-image generation, 2026. URL https://arxiv.org/abs/2602.13585.

[36] Linghui Shen, Mingyue Cui, and Xingyi Yang. Decontext as defense: Safe image editing in diffusion transformers, 2025. URL https://arxiv.org/abs/2512.16625.

[37] Joonghyuk Shin, Alchan Hwang, Yujin Kim, Daneul Kim, and Jaesik Park. Exploring multimodal diffusion transformers for enhanced prompt-based image editing, 2025. URL https://arxiv.org/abs/2508.07519.

[38] Google DeepMind. Gemini 3.1 technical report. https://deepmind.google/technologies/gemini/, 2025.

A Full Discovery Protocol

	Layer probe	Timestep probe	Spatial blueprint
Method	Residual bypass	$\hat{x}_0$ decode	Human-aligned VLM
Metric	DINO-v2 (ViT-L/14)	DINO-v2 (ViT-L/14)	Feature/bbox agreement
Study set	60 independent object/product probe pairs	60 independent object/product probe pairs	60 independent object/product probe pairs
Runs	10,200 (3 models)	$60 \times \{30, 28, 50\}$	60×2 VLM passes
Backbones	Diptych, Kontext, FLUX 2.0	same	Qwen 3.5-9B
Seed	42	42	–

Table 5: Discovery protocol summary. Layer and timestep probes measure identity sensitivity via DINO-v2 cosine similarity. Spatial blueprints identify and localize identity-critical features. The 60-pair discovery probe set is randomly sampled and disjoint from the product identity benchmark used for final evaluation.

Backbone	Final DINO	First 3 steps	t_{90}	Late gain
Diptych (FLUX.1)	0.631	46%	41.6%	+0.060 after step 15
Kontext (FLUX.1)	0.797	83%	16.8%	+0.014 after step 15
FLUX 2.0	0.557	33%	23.9%	+0.015 after step 15

Table 6: Temporal identity-formation summary. The first-three-step column reports the fraction of final DINO-v2 similarity already present by step 3. t_{90} is the normalized denoising progress where the curve reaches 90% of its final value.

Kontext forms identity fastest and reaches the highest final identity score, consistent with its native reference-conditioned architecture. FLUX 2.0 has the slowest early formation and the lowest identity ceiling. Diptych is intermediate: it forms coarse identity earlier than FLUX 2.0, but it keeps refining late, gaining 9.5% of its final DINO-v2 score after step 15. This late refinement supports keeping a fine-detail intervention window rather than restricting attention control to the first few denoising steps.

B Full Vital-Layer Tables

Backbone	Architecture	Top-5 vital layers	$\mathcal{L}_{\text{id}}$ (top-10)	Jaccard	Harmful
Diptych (FLUX.1)	19 dbl + 38 sgl	L36, L45, L24, L52, L21	{0,1,18,21,24,36,45,50,51,52}	0.176	L53, L54
Kontext (FLUX.1)	19 dbl + 38 sgl	L1, L0, L2, L3, L6	{0,1,2,3,5,6,10,16,17,18}	–	L4, L7–L9
FLUX 2.0	8 dbl + 48 sgl	L0, L7, L10, L9, L27	{0,1,3,7,8,9,10,12,13,27}	–	L4, L43, L53, L54

Table 7: Identity-vital layer sets $\mathcal{L}_{\text{id}}$ for the three FLUX-family backbones; Figure 2 outlines the top-5 of the same ranking. Harmful layers are those where bypassing the layer improves identity similarity.

Kontext concentrates nearly all measured identity impact in the first few layers: L0, L1, and L2 dominate, while the single-stream layers contribute little individually. FLUX 2.0 is even more concentrated, with L0 contributing roughly five times more than the next strongest layer, L7. Diptych is the least concentrated: its highest-impact layer is only modestly above other mid-to-late single-stream layers, so its identity signal is more distributed.

The comparison to StableFlow-style edit-vital layers is available for Diptych. The identity-vital and edit-vital top-10 sets have Jaccard overlap 0.176 and Pearson correlation $r = 0.35$. Layers L53 and L54 are especially informative: StableFlow-style edit probing marks late layers as useful for editing, but bypassing these layers improves identity in our setting. This is the main evidence that identity retention should be probed directly rather than inherited from generic editability.

C Cross-Architecture Generalization

To test whether VitalScore is FLUX-specific, we ran the full pipeline (discovery, two-phase blueprint, $\Lambda(l, t, s)$, $P(t)$) on two architecturally distinct non-FLUX backbones: Qwen-Image-Edit [19], a pure double-stream MMDiT, and FIBO-Edit [20], a dual-stream editing DiT with an LLM text encoder. Discovery follows Appendix A on the same 60-pair probe set and costs $\sim$16 GPU-h per backbone.

Evaluation uses the full 520-pair DreamBench++ set with the protocol of Table 1a; results are reported there.

Discovery adapts per architecture. Qwen-Image-Edit forms identity late ($t_{90} = 0.436$), with vital layers L0, L1, L43, L58, L59 plus a late cluster; FIBO-Edit locks identity almost at initialization ($t_{90} = 0.067$), with vital layers L0, L1, L7, L40, L41. Applying the window-derivation rule of Section 4 to these curves yields an immediate-onset schedule for FIBO-Edit and a late schedule for Qwen-Image-Edit, rather than the shared FLUX windows. Layer 0 is vital on all five backbones, but the rest of each set is backbone-specific, which is why VitalScore re-runs discovery rather than transferring a layer set.

Gains. Qwen-Image-Edit starts from a baseline comparable to Kontext (0.655 vs. 0.668 DINO-v2) and gains +8.1% DINO-v2, with CP and PF improving (0.676→0.692, 0.892→0.914). FIBO-Edit has a high baseline (0.891) and gains +3.6%, with all secondary metrics stable or improving. Neither backbone required changes to λ_0, $\lambda_{\max}$, or τ_p.

Scope. The setting is reference-conditioned generation. DiT-XL is class-conditional and SD3 exposes no native reference-conditioning interface, so neither supports the task; the probing protocol applies to any reference-conditioned DiT.

D Additional Component Ablations

Backbone	Configuration	DINO-v2	CLIP-T
Diptych	Baseline	0.674	**0.237**
	+ Spatial all layers	0.636	0.227
	Spatial + vital layers	0.694	0.231
	Spatial + vital layers + temporal	0.699	0.232
	+ Prompt schedule (full VitalScore)	**0.701**	0.236
Kontext	Baseline	0.794	0.233
	+ Spatial all layers	0.763	0.216
	Spatial + vital layers	0.832	0.223
	Spatial + vital layers + temporal	**0.851**	0.223
	+ Prompt schedule (full VitalScore)	0.850	**0.234**

Table 8: Additional product-set component ablations for Diptych and FLUX Kontext. The same design story holds: ungated spatial modulation can hurt, while applying the spatial signal only at identity-vital layers improves. Temporal scheduling helps improve overall performance. Prompt scheduling plays an important role in improving prompt alignment.

E Robustness Analyses

Both analyses use FLUX Kontext, the backbone with the largest VitalScore gain and hence the strictest relative test, on a fixed 130-pair DreamBench++ subset. Baseline and all variants share seeds and the same evaluation harness; absolute values therefore differ from the full 520-pair results in Table 1. Significance is tested per configuration with a paired Wilcoxon signed-rank test over pairs.

Temporal-window sensitivity. We shift every window boundary in Table 17 earlier or later by a fixed offset (± 0.05 or ± 0.10 in normalized progress, i.e., up to ± 3 of Kontext's 28 steps), and separately re-derive the windows from Kontext's own formation curve ($t_{90} = 0.168$) instead of the shared FLUX set. If the windows were fragile tuned constants, these shifts should visibly change DINO-v2; none does (Table 9). The windows act as coarse regime markers rather than sensitive constants.

Window configuration	DINO-v2	Δ vs. paper windows
Paper windows (reference)	0.8182	–
All boundaries -0.05 / $+0.05$	0.8180 / 0.8180	-0.0002 / -0.0002
All boundaries -0.10 / $+0.10$	0.8179 / 0.8189	-0.0003 / $+0.0007$
Re-derived from Kontext's own t_{90} (0.168)	0.8173	-0.0009
Baseline (no VitalScore)	0.7306	–

Table 9: Temporal-window sensitivity (Kontext, 130-pair DreamBench++ subset). No variant is statistically distinguishable from the paper windows (Wilcoxon signed-rank, all $p \geq 0.11$), while the VitalScore gain over baseline is +0.088.

Blueprint corruption. All reported results already include naturally occurring blueprint errors: the Qwen3.5-9B blueprints used throughout reach feature F1 0.87 and box IoU 0.61 against human annotations (Table 11). Observed errors are mostly localization noise (shifted or mis-sized boxes) rather than wrong features. Note that the 70–75% exact-agreement figures in Table 13 measure the *evaluation judge* against human raters, not blueprint quality. To measure behavior as blueprints get worse, we corrupt them at increasing severity and re-run the full pipeline (Table 10). Jitter perturbs box coordinates at two severities (± 10 and ± 20); feature dropping removes a random 50% of regions, which also emulates feature-selection errors; random boxes replace every box with a uniformly sampled one.

Blueprint condition	**DINO-v2**	**Δ vs. intact**	**% of gain kept**
Intact (two-phase)	0.8182	–	100%
Box jitter ± 10	0.8106	-0.0076	91%
Box jitter ± 20	0.8012	-0.0170	81%
50% of features dropped	0.7864	-0.0318	64%
Fully random boxes (adversarial)	0.5803	-0.2379	< 0
Baseline (no VitalScore)	0.7306	–	0%

Table 10: Blueprint corruption (Kontext, 130-pair DreamBench++ subset). Realistic errors degrade gracefully; only fully random placement, far outside any observed VLM error mode, falls below baseline.

A slightly shifted or mis-sized box still points attention at roughly the right region, so most of the gain survives jitter. The adversarial extreme confirms that the spatial axis carries real signal: if the blueprint were decorative, corrupting it would change nothing. When no valid blueprint is available, the validation step of Section 4 falls back to uniform λ_0, i.e., plain baseline behavior, and stronger VLMs give better blueprints (Table 4b: $0.718 \rightarrow 0.745 \rightarrow 0.748$, all above the 0.710 baseline).

F Blueprint Details

F.1 Two-phase VLM prompt templates

Splitting identity recognition and spatial localization into two separate VLM passes improves bbox precision, particularly for smaller VLMs that struggle to simultaneously reason about feature identity and coordinate placement. Phase A asks the VLM to identify *what* features exist and assigns each a type, importance, and granularity; Phase B receives the frozen feature list from Phase A and asks only *where* each feature is located.

Phase A: Recognition. The Phase A prompt instructs the VLM to analyze the reference image and output a JSON object with the following schema:

```
{
  "features": [
    {
      "name":        "<unique_descriptive_name>",
      "type":        "<logo|text|shape|texture|color|hardware|marking|pattern>",
      "importance":  <0.0-1.0>,
      "granularity": "<coarse|mid|fine>",
      "description": "<short visual description>"
    }
  ],
  "edit_type":            "<background_change|style_transfer|...>",
  "identity_description": "<10-30 word noun phrase describing the subject>",
  "enhanced_prompt":      "<original prompt enriched with identity details>"
}
```

Key constraints enforced by the Phase A prompt:

- **Granularity maps to denoising phase:** `coarse` = overall silhouette (active early); `mid` = textures and color blocks (mid-denoising); `fine` = logos, text, small markings (active late).
- **Importance scale:** 1.0 = brand logo or product name; 0.7–0.9 = distinctive color or key texture; $<$0.3 = minor features.

- **P2 — Prompt-aware coarse importance:** if the target prompt implies pose, motion, or viewpoint change (running, jumping, tilted), coarse-granularity features are set to low importance (0.2–0.4); otherwise high (0.7–0.9). Fine and mid features are always high importance.
- **P3 — Small-detail quota:** at least 2 features must occupy $< 3\%$ of image area; any readable text is always `type=text, granularity=fine, importance` ≥ 0.9; budget split: $\leq$1 coarse, $\leq$2 mid, $\geq$3 fine features.

Phase B: Localization. Phase B receives the reference image and the frozen feature list from Phase A. It outputs only bounding boxes:

```
{
  "bboxes": [
    {"name": "<feature_name>", "bbox": [x1, y1, x2, y2]}
  ]
}
```

Coordinates are normalized to $[0, 1]$; (x_1, y_1) is top-left, (x_2, y_2) is bottom-right. Tightness constraints: coarse bbox area must be $< 40\%$; fine detail bboxes must be $< 3\%$. The prompt also enforces viewpoint-awareness rules (e.g., for a left-facing shoe, heel features must appear at high x values). Features that are occluded or not visible receive $[0, 0, 0, 0]$ and are dropped from the final blueprint.

Worked example. For a Nike Air Max 90 sneaker with the prompt *"on a city park path during golden hour, worn by a person casually walking"* (reference `ref_000`):

Phase A features:

```
overall_silhouette  shape    imp=0.90  coarse
suede_upper         texture  imp=0.80  mid
olive_swoosh        logo     imp=0.85  fine
air_branding        text     imp=0.75  fine
visible_air_unit    shape    imp=0.80  mid
black_overlay       color    imp=0.70  mid
white_midsole       color    imp=0.60  mid
olive_heel_tab      color    imp=0.60  mid
```

Phase B bboxes (selected):

```
overall_silhouette  [0.05, 0.40, 0.95, 0.85]   area=0.405 (coarse)
olive_swoosh        [0.45, 0.55, 0.55, 0.65]   area=0.010 (fine)
air_branding        [0.58, 0.58, 0.72, 0.62]   area=0.006 (fine)
visible_air_unit    [0.55, 0.68, 0.78, 0.75]   area=0.016 (mid)
```

The coarse silhouette (area 0.405) is just under the 40% limit; the text and logo boxes are both $< 1\%$ area, satisfying the fine-detail tightness constraint.

F.2 VLM-human alignment protocol

We validated blueprint quality against human annotations on the product set. Human annotators marked identity-critical features and drew tight bounding boxes on each reference image; blueprint output was compared for feature-selection agreement (precision, recall, F1), spatial alignment (mean bbox IoU), and oversized-box rate. Results are in Table 11. Gemini 3.1 Pro achieves higher agreement across all metrics; Qwen3.5-9B is competitive and preferred for local deployment.

VLM	Feature P	Feature R	Feature F1	BBox IoU	Oversized-box rate
Qwen3.5-9B	0.86	0.88	0.87	0.61	14%
Gemini 3.1 Pro	0.93	0.94	0.93	0.67	9%

Table 11: VLM-human alignment for blueprint feature selection and localization (product set). Feature P/R/F1 measures agreement on which features are identity-critical; BBox IoU measures spatial alignment of predicted vs. human boxes; oversized-box rate is the fraction of coarse boxes exceeding 40% image area.

F.3 Single-pass vs. two-phase blueprint ablation

Metric	Single-pass prompt-aware	Two-phase
Mean regions / blueprint	7.7	7.1
Mean fine regions	4.0	3.9
Mean text regions	2.0	1.9
Mean small regions ($< 3\%$ area)	4.6	3.7
Mean coarse bbox area	0.488	0.374
Coarse boxes $> 40\%$ area	55%	32%
Best Diptych DINO	0.6819	0.7011
Best Kontext DINO	0.8346	0.8504

Table 12: Blueprint strategy ablation on the product set. The two-phase prompt first fixes the identity-critical feature list, then localizes only those features. This produces tighter coarse boxes and better downstream identity scores.

The key difference is spatial precision. Two-phase blueprints reduce mean coarse-box area from 0.488 to 0.374, a 24% reduction, and cut oversized coarse boxes from 55% to 32%. Tighter coarse boxes reduce attention bleeding, where reference details leak into background tokens that the prompt should control. The downstream effect is clear on both pipelines: Diptych DINO-v2 improves from 0.6819 to 0.7011 (+2.8%) and Kontext from 0.8346 to 0.8504 (+1.9%). These results confirm that separating feature recognition from spatial localization produces more precise spatial control and that the main method's two-phase design is justified.

G Benchmark Details

G.1 Product identity benchmark

The 240-pair product benchmark consists of 60 reference product images across ten categories: Food & Beverage, Kitchen Appliances, Watches, Footwear, Home & Furniture, Bags & Apparel, Perfume & Beauty, Electronics & Audio, Sports & Outdoor, and Toys & Collectibles. **This evaluation set is disjoint from the 60-pair discovery probe set used to estimate identity-vital layers and timesteps**. Each benchmark reference is paired with four prompt axes — scene context, lighting, camera composition, and action/interaction — yielding 240 evaluation pairs. Products were selected to cover a range of identity difficulty, from subjects with distinctive logos and printed text (high verifiability) to subjects defined primarily by shape or color (lower verifiability). The action/interaction axis is the hardest: the subject must adapt its pose or physical interaction to the scene description while all fine-detail cues must remain intact.

G.2 DreamBench++ subset

We use the full DreamBench++ evaluation set of 520 pairs (humans, animals, and objects; style prompts excluded), following the original protocol [21]. DINO-v2, CLIP-I, and CLIP-T are computed with the standard feature extractors. VLM concept preservation (CP) and prompt following (PF) scores are reported as secondary signals.

G.3 VLM judge prompt and human alignment

Product-benchmark outputs are scored by a Gemini 3.1 Pro judge [38] on three dimensions: structure/semantic identity, detail fidelity, and aesthetics/integration, each on a five-level ordinal scale. To calibrate the judge, three annotators each independently rated a random sample of 20 product-benchmark outputs on the same three dimensions; the judge scores were then compared against the human ratings.

Table 13 reports exact-match and within-1 agreement per dimension across 20 randomly sampled outputs. With three human annotators, this yields 60 judge-human comparisons per dimension group and 180 comparisons overall; we report Wilson 95% confidence intervals for each agreement rate. Structure/semantic identity and aesthetics achieve 75% exact and 85% within-1 agreement; detail fidelity is slightly lower at 70% exact and 80% within-1, reflecting the greater subjectivity in judging small visual cues.

Dimension	Comps.	Exact (n/N)	Exact 95% CI	Within-1 (n/N)	Within-1 95% CI
Structure/Semantic identity	60	75.0% (45/60)	[62.8, 84.2]	85.0% (51/60)	[73.9, 91.9]
Detail fidelity	60	70.0% (42/60)	[57.5, 80.1]	80.0% (48/60)	[68.2, 88.2]
Aesthetics	60	75.0% (45/60)	[62.8, 84.2]	85.0% (51/60)	[73.9, 91.9]
Overall	180	73.3% (132/180)	[66.4, 79.3]	83.3% (150/180)	[77.2, 88.1]

Table 13: Gemini 3.1 Pro judge vs. human annotator agreement on product-benchmark outputs (3 annotators). Comps. counts judge-human comparisons; within-1 means the judge and human differ by at most one ordinal level. Confidence intervals are Wilson binomial 95% intervals.

H Negative Results

H.1 Frequency-Anchored Initial Noise

We explored whether blending reference-image structure into the initial noise could bias flow-matching denoising toward better identity preservation. The approach uses a linear blend $z_T = (1-\alpha)\,\epsilon + \alpha\, z_{\text{ref}}^{\text{norm}}$, where $z_{\text{ref}}^{\text{norm}}$ is the reference's VAE latent normalized to unit variance and α controls blend strength. This is training-free and architecture-agnostic, operating before any model forward pass.

The method fails for flow-matching models. FLUX learns straight-line trajectories from unit-variance Gaussian noise ($t = 1$) to data ($t = 0$); any structural bias at $t = 1$ is off the learned manifold. As shown in Table 14, a binary transition occurs at $\alpha \approx 0.03$: below this value the perturbation is too small to influence the 28-step denoising trajectory; at and above it, the model converges on the reference image rather than the prompted scene. There is no α that improves identity while maintaining scene diversity.

	Sneakers			*Clothing*		
α	DINO-v2	CLIP-T	LPIPS	DINO-v2	CLIP-T	LPIPS
baseline	0.518	**0.327**	0.855	0.746	**0.327**	0.661
0.01	0.508	0.328	0.836	0.750	0.324	0.630
0.03	**0.973**	0.228	**0.001**	**0.961**	0.284	0.114
0.05	0.986	0.225	0.009	0.965	0.270	0.104
0.10	0.955	0.221	0.046	0.949	0.262	0.096
0.20	0.925	0.212	0.080	0.929	0.236	0.104

Table 14: Frequency-anchored noise results on FLUX Kontext (2 product reference pairs). At $\alpha = 0.01$ there is no identity benefit; at $\alpha = 0.03$ LPIPS ≈ 0.001 indicates the output is a near-copy of the reference with CLIP-T dropping 30%. No α value achieves both improved identity and prompt alignment.

This contrasts with traditional DDPM/DDIM partial noising, where starting from an intermediate timestep $t < T$ provides a smooth quality/similarity tradeoff because intermediate timesteps lie on the learned trajectory. For flow-matching, the initial noise is the unique starting point; biased noise is never a valid trajectory origin. We also tested FFT-based frequency swapping (replacing low-frequency noise components with reference frequencies), which produced colored non-white noise that broke denoising entirely (DINO-v2 = 0.205 vs. baseline 0.463). Noise-shaping approaches are therefore incompatible with flow-matching pipelines, and VitalScore operates inside the learned inference process via attention redistribution instead.

H.2 Other Negative Results

Mid-denoising VLM correction loop. We evaluated a variant in which a VLM scores intermediate $\hat{x}_0$ predictions during denoising and adjusts the reference attention strength based on the running identity score. Preliminary evaluation showed no consistent identity gain over the standard VitalScore without the correction step, while increasing inference time substantially due to per-step VLM calls.

Fine-detail-only spatial restriction. We tested a variant that applies the spatial blueprint modulation exclusively to fine-granularity regions (logos, text, small markings), leaving coarse and mid regions at the baseline attention level. This reduced copy-and-paste artifacts but also reduced identity gains, as coarse-region conditioning helps anchor the overall subject before fine details are resolved.

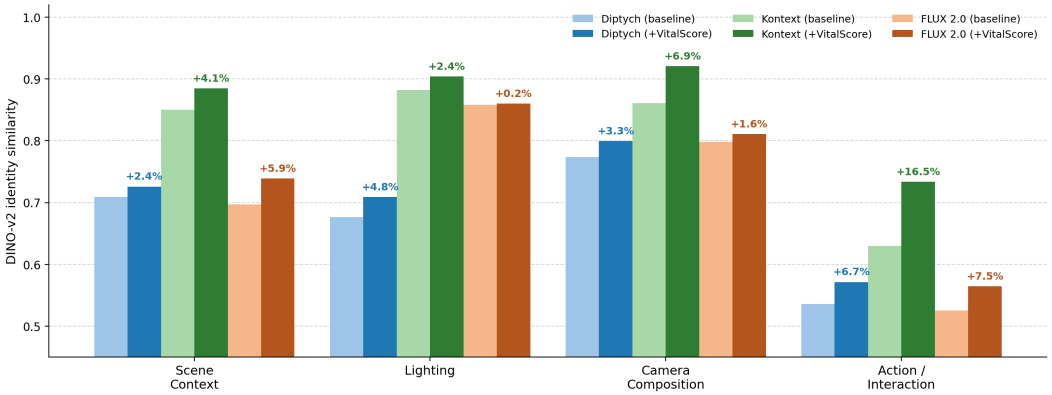


Figure 7: Per-prompt-axis DINO-v2 across three FLUX-family backbones (60 objects $\times$ 4 axes = 240 pairs). Delta annotations show the improvement of full VitalScore over each backbone's plain baseline.

I Prompt Axis and Add-ons

I.1 Prompt schedule boundaries and implementation

The prompt schedule used in main-text results defines two stages tied to normalized denoising progress $\rho(t) = t/(T-1)$:

- **S1** $\rho(t) \in [0.00, \tau_p)$ with $\tau_p = 0.10$: scene-first prompt P_{scene}, containing no brand or identity-specific tokens, allowing the model to establish layout and scene composition freely.
- **S2** $\rho(t) \in [\tau_p, 1.00]$: full enhanced prompt P_{full}, restoring all identity-detail constraints once fine-detail crystallization begins.

For FLUX Kontext (28 denoising steps), this places the boundary at step 3. All stage prompts are pre-encoded before generation starts. The active embedding is swapped via `callback_on_step_end` at the boundary step; critically, both `prompt_embeds` and `pooled_prompt_embeds` must be listed in `callback_on_step_end_tensor_inputs`, otherwise the swap silently degenerates into the initial prompt embedding throughout all steps.

Architecture sensitivity varies across the three tested pipelines. FLUX 2.0 commits more strongly at step 0 due to its denoising dynamics, so prompt schedule gains are modest compared to Kontext. On the 240-pair product benchmark, adding the prompt schedule to the VitalScore attention axes moves FLUX 2.0 from 0.743 to 0.745 DINO-v2 and from 0.231 to 0.237 CLIP-T (Table 4a), confirming the prompt swap contributes even under stronger step-0 conditioning. For Diptych the schedule mainly restores prompt alignment (CLIP-T 0.232 $\rightarrow$ 0.236, DINO-v2 0.699 $\rightarrow$ 0.701; Table 8), because early identity tokens in the dual-panel layout can interfere with panel boundary formation.

I.2 Per-axis and per-category breakdown

Table 15 reports per-prompt-axis DINO-v2 for FLUX Kontext on the 240-pair product benchmark (60 objects $\times$ 4 axes). The action/interaction axis is consistently the hardest: the subject must integrate with hands, bodies, or props while all fine-detail cues must survive, creating maximum identity drift. VitalScore's selective attention redistribution is most valuable exactly here, improving DINO-v2 by 16.5%.

Prompt axis	Plain	+VS	+VS+schedule	Δ (full vs Plain)
scene_context	0.850	0.878	0.885	+4.1%
lighting	0.882	0.893	0.904	+2.4%
camera_composition	0.861	0.896	0.921	+6.9%
action_interaction	0.630	0.707	0.734	+16.5%

Table 15: Per-prompt-axis DINO-v2 for FLUX Kontext (60 objects $\times$ 4 axes = 240 pairs). Action/interaction prompts produce the largest identity drift at baseline and the largest absolute gain from full VitalScore with the prompt schedule.

Table 16 breaks down DINO-v2 by product category. Footwear is the hardest category across all backbones and all intervention levels (0.44–0.62 even with full VitalScore), because sneaker identity

depends on logos, colorways, and lacing patterns that occupy very small spatial regions. Toys & Collectibles and Sports & Outdoor show the largest VitalScore gains for Kontext. Kitchen Appliances and Electronics achieve high baseline DINO (>0.89) and benefit less from additional conditioning.

	Diptych		Kontext			FLUX 2.0		
Category	Plain	+full	Plain	+VS	+full	Plain	+VS	+full
Bags & Apparel	0.656	0.679	0.897	0.911	**0.924**	0.737	0.748	0.825
Electronics & Audio	0.748	0.780	0.900	0.916	**0.917**	0.741	0.761	0.811
Food & Beverage	0.610	0.622	0.795	0.839	**0.844**	0.793	0.799	0.803
Footwear	0.513	0.514	0.522	0.544	**0.623**	0.444	0.459	0.452
Home & Furniture	0.758	0.720	0.819	0.882	**0.900**	0.818	0.858	0.864
Kitchen Appliances	0.782	0.786	0.899	0.920	**0.929**	0.771	0.799	0.756
Perfume & Beauty	0.568	0.588	0.799	0.826	**0.848**	0.683	0.692	0.716
Sports & Outdoor	0.758	0.771	0.825	0.911	**0.934**	0.765	0.765	0.822
Toys & Collectibles	0.527	0.537	0.887	0.948	**0.959**	0.896	0.916	0.892
Watches	0.799	0.797	0.798	**0.835**	0.831	0.598	0.596	0.584

Table 16: Per-category DINO-v2 on the 240-pair product benchmark. VS = VitalScore attention axes only; full = full VitalScore with prompt schedule. Bold = best Kontext result per category.

I.3 Seed selection

All main-text results use a fixed seed shared between baseline and VitalScore for each pair, so that reported gains reflect conditioning quality rather than sampling luck. In practice, running k seeds and selecting the highest-DINO-v2 output is a simple add-on that compounds with VitalScore's conditioning gains. In our informal trials on the product benchmark, best-of-3 seed selection adds approximately 1–2% DINO-v2 on top of the full VitalScore result without any change to the method. We treat seed selection as orthogonal to the core contribution and exclude it from the main comparison to keep the evaluation fair.

J Hyperparameters and Compute

Hyperparameters. Table 17 lists all VitalScore hyperparameters used in the main-text experiments. No per-image tuning was performed; all values are fixed across the full evaluation sets.

Parameter	**Symbol**	**Value**	**Description**
Baseline lambda	λ_0	1.3	Diptych Prompting default; baseline for non-identity tokens
Max lambda	$\lambda_{\max}$	2.0	Upper bound for identity-region attention boost
Lambda clamp	λ_{clamp}	4.0	Safety ceiling to prevent attention blowup
Coarse window	W_{coarse}	[0.00, 0.35]	Normalized denoising progress range for coarse features
Mid window	W_{mid}	[0.15, 0.65]	Normalized denoising progress range for mid features
Fine window	W_{fine}	[0.35, 1.00]	Normalized denoising progress range for fine features
Prompt switch	τ_p	0.10	Normalized progress at which $P_{\text{scene}} \to P_{\text{full}}$
Denoising steps	T	30 / 28 / 50	Diptych / Kontext / FLUX 2.0

Table 17: VitalScore hyperparameters used in all main-text experiments. The window boundaries are shared by the three FLUX backbones; Qwen-Image-Edit and FIBO-Edit use windows re-derived from their own formation curves (Appendix C).

Compute. All generation experiments were run on NVIDIA A100 80 GB GPUs. The layer and timestep discovery probes (10,200 runs across the three FLUX backbones) required approximately 48 GPU-hours in total, i.e., ∼16 GPU-hours per backbone, amortized once; discovery on Qwen-Image-Edit and FIBO-Edit cost the same order per backbone. Blueprint generation with Qwen3.5-9B adds roughly 2–3 seconds per image on a single GPU; Gemini 3.1 Pro calls were made via API. Evaluation of the 240-pair product benchmark (3 pipelines × 2 configs) required approximately 12 GPU-hours; the 520-pair DreamBench++ evaluation required approximately 20 GPU-hours for the three FLUX pipelines.

Inference cost. VitalScore adds no extra denoising passes and no extra model forward passes (Table 18). Because $\Lambda(l, t, s)$ scales only reference key tokens, it folds into the value rows (Section 4), so modulated attention runs through standard fused kernels. Measured per-image generation time rises from 13.0 s to 14.0 s on FLUX Kontext (1.07×) and from 19.3 s to 21.2 s on Qwen-Image-Edit (1.11×).

Stage	Cost	Frequency
Layer/timestep discovery	$\sim$16 GPU-h	once per backbone (offline, amortized)
Blueprint (local Qwen3.5-9B)	2–3 s	once per reference; cacheable; zero if user-drawn
Building Λ, $P(t)$	< 0.1 s	per image
Denoising	same steps, single pass; 1.07–1.11$\times$ wall-clock	per image

Table 18: VitalScore inference cost breakdown.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state four concrete contributions—identity-vital layer/timestep discovery, the VitalScore redistribution tensor $\Lambda(l, t, s)$, the two-phase VLM blueprint, and the product identity benchmark—and all are supported by quantitative results in Section 5.2 and the appendix. Claimed gains (up to 15% DINO-v2 on DreamBench++, 16.5% on the hardest product axis) match Table 1 and Table 15. The scope is stated as reference-conditioned diffusion transformers, supported by three FLUX-family and two non-FLUX backbones.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations paragraph at the end of the Conclusion discusses: VitalScore requires offline per-backbone probing; it targets reference-conditioned DiTs only; non-FLUX backbones are evaluated on DreamBench++ only; the spatial axis depends on blueprint quality (validated in Appendix F, with corruption analysis in Appendix E); the prompt schedule has reduced effect on architectures that commit strongly at step 0; and the method has not been extended to multi-subject or video settings.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper is empirical; it contains no theorems, lemmas, or formal proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 and Appendix A fully describe the layer/timestep probing protocol (60-pair discovery set, DINO-v2 ViT-L/14, residual-bypass procedure, seed 42). Appendix B lists the exact identity-vital layer sets per backbone. Appendix F gives the complete two-phase VLM prompt templates and tightness constraints. Section 5.1 describes evaluation datasets, metrics, and pipeline settings. The product benchmark construction is detailed in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code and data cannot be released due to company policy. Instead, the method is fully described in Sections 3–4 and Appendices A, B, and F (discovery protocol, vital-layer sets, VLM prompt templates), and all hyperparameters are listed in Appendix J, providing a sufficient basis for reproduction without the code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: VitalScore is training-free, so there are no optimizers or training splits. All inference hyperparameters (λ_0, $\lambda_{\max}$, granularity window boundaries W_γ, prompt schedule threshold τ_p, and per-backbone denoising step counts) are tabulated in Appendix J (Table 17). The discovery probe set (60 pairs, disjoint from evaluation) and backbone-specific settings are given in Appendices A and C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main metrics are means over large fixed evaluation sets (520 pairs for DreamBench++, 240 pairs for the product benchmark), which limits per-sample variance. The VLM judge human-alignment results in Appendix G report Wilson 95% confidence intervals, and the window-sensitivity analysis in Appendix E uses paired Wilcoxon signed-rank tests. Error bars on per-image generation metrics are not reported because each pair uses a fixed shared seed, making the comparison deterministic rather than stochastic.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix J reports GPU type (NVIDIA A100 80 GB), approximate GPU-hours for discovery probing ($\sim$16 GPU-hours per backbone, amortized once), blueprint generation ($\sim$2–3 s/image), and evaluation runs ($\sim$12 and $\sim$20 GPU-hours for the two benchmarks), plus a per-stage inference-cost table (Table 18). VitalScore adds no extra denoising passes; measured generation overhead is 1.07–$1.11\times$.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics. The work involves no human participant data collection beyond the small annotator study (3 annotators scoring 20 outputs each), which poses minimal risk. No personally identifiable information was collected; all product reference images are commercially licensed or photographed in-house.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The Conclusion notes positive applications (product photography, background replacement, personalized content creation) and acknowledges that improved subject fidelity in generative models could be misused for identity spoofing or deepfake creation. We discuss these dual-use considerations and note that VitalScore operates on commercial product images rather than human subjects, reducing the most direct risks.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: VitalScore does not release a new generative model or scraped dataset. It is a training-free conditioning method that wraps existing frozen backbones (FLUX-family

models) which carry their own usage licenses and safety filters. No code, model, or dataset is released.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All models used (FLUX.1 Kontext, FLUX 2.0, Diptych Prompting, Qwen3.5-9B, Gemini 3.1 Pro) are cited. DreamBench++ is cited and used under its original terms. The StableFlow probing approach is cited. We use DINO-v2 and CLIP under their respective open-source licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: No new assets are released due to company policy. The 240-pair product identity benchmark used for evaluation is documented in Appendix G with category breakdown, prompt axes, and construction criteria.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The human evaluation involves 3 internal annotators who rated 20 outputs each on the same three-dimension rubric used by the VLM judge (structure/semantic identity, detail fidelity, aesthetics/integration). The rating criteria and scale are described in Appendix G. Annotators are employees of the authors' institution; no external crowdsourcing platform was used.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The human evaluation task (rating generated product images on a 5-level scale) poses negligible risk to participants. It involves no sensitive data, personal information, or deceptive elements. IRB approval is not required for this type of in-house annotation task.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: VLMs (Qwen3.5-9B and Gemini 3.1 Pro) are a core methodological component: they generate the two-phase identity blueprints that drive the spatial axis of VitalScore. Their role, prompts, and outputs are described in detail in Section 4 and Appendix F. LLMs were also used for writing assistance but this does not affect the scientific content.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.