HyperTransport: Amortized Conditioning of T2I Generative Models

Valentino Maiorca **Eleonora Gualdoni** **Xavier Suau**
Marco Cuturi **Luca Zappella** **Pau Rodríguez**
Apple

Abstract

As foundation models grow in capability, the ability to efficiently and reliably control their behavior becomes critical. Fine-tuning these models can be costly, and while prompting can be practical for controllability, it offers no continuous, interpretable control over the strength of a requested attribute. This limitation has driven interest in activation steering techniques that offer more stable and predictable control over model behavior. However, existing activation steering methods require per-concept optimization, which makes them ill-suited to deployment scenarios where the concept set is large, evolving, or only specified at request time: each new concept incurs at least minutes of optimization on the target model. We propose HyperTransport, a hypernetwork framework that amortizes this cost by mapping embeddings from a pretrained encoder (CLIP in our instantiation) directly to intervention parameters, trained end-to-end using an optimal transport loss. Once trained, HyperTransport produces each new intervention in a single hypernetwork forward pass, 3600–$7000\times$ faster than per-concept fitting. On concepts unseen during training, it matches the strongest per-concept baselines at inducing the target concept. By decoupling concept representation from intervention prediction, HyperTransport combines three capabilities that no existing approach offers as a set: amortized steering for open-ended concept sets, continuous interpretable strength control, and cross-modal conditioning where reference images can directly steer text-based generation. We validate HyperTransport on DMD2, Nitro-1-PixArt, and FLUX.2 [klein] across 158 held-out test concepts. In pairwise comparisons, both human and VLM judges prefer HyperTransport over prompting $\sim 2\times$ as often.
Code: `https://github.com/apple-aiml-research/ml-hypertransport`

1 Introduction

As generative models continue to scale, methods for efficiently controlling them toward specialized behaviors are critical, both for aligning towards human preferences (*e.g.,* mitigating toxic language) and for creative applications like personalized image generation. Traditional approaches rely on fine-tuning entire models [Ouyang et al., 2022] or crafting precise prompts [Brown et al., 2020]. However, fine-tuning is computationally demanding, while prompting inherently lacks fine-grained continuous control, even with recent advances in instruction following [Cheng et al., 2026].

Activation steering takes a different approach: small, targeted modifications of internal activations express a target concept without changing the model's weights. The premise is now well supported [Zou et al., 2023]: semantic modifications like "*apply Van Gogh style*" are identifiable within latent space and reachable through simple, often affine, activation transformations. Recent methods achieve state-of-the-art results across a wide range of concepts, including toxicity mitigation [Rodriguez et al., 2025b], object removal in 1-step distilled image generators [Rodriguez et al., 2025a], and truthfulness induction [Li et al., 2023]. But every new concept still requires *ad-hoc optimization* on

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

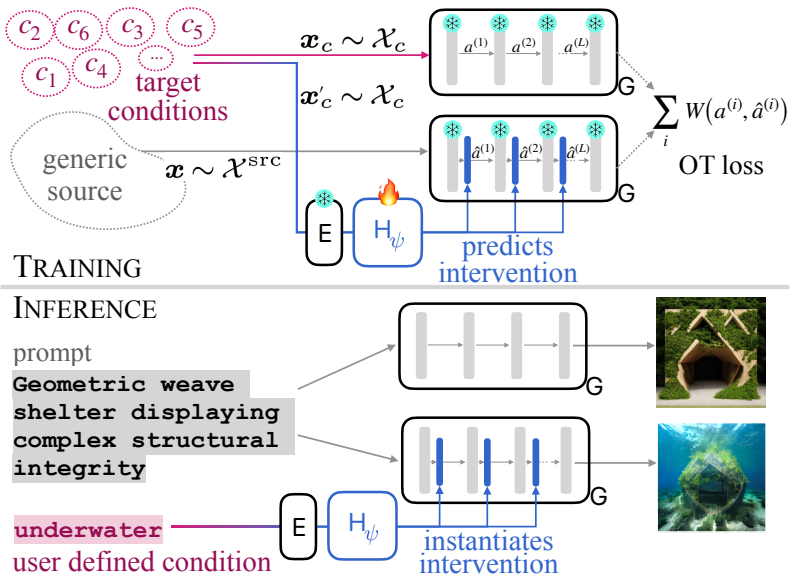

Figure 1: **Training and inference of HyperTransport**. During training (top), we sample $\{\boldsymbol{x}_c\}$ sentences corresponding to a concept c, along with $\{\boldsymbol{x}\}$ unrelated sentences. The hypernetwork H_ψ predicts an intervention on G from a target encoding given by an encoder E. H_ψ is updated using an Optimal Transport loss W until convergence. During inference (bottom) $\mathsf{H}_\psi \circ \mathsf{E}$ dynamically predicts high quality interventions for test concepts in negligible time.

the target model, costing minutes of forward (and optionally backward) passes. This per-concept cost is incompatible with the real-time inference of few-step generators [Chen et al., 2025, Zhao et al., 2023, Fan et al., 2025]. This rules out interactive interfaces where users specify concepts on demand, content pipelines whose policy set expands over time, and any system that must serve unforeseen concepts at request time.

HyperTransport (HT) *amortizes* this per-concept optimization through a hypernetwork $1000\times$ smaller than the target image generator. Conditioned on CLIP [Radford et al., 2021] representations, it maps directly to intervention parameters and serves any concept well-represented in CLIP space, even ones unseen at training, in a single forward pass. HyperTransport thus borrows the activation-space precision of per-concept steering and the open, immediate interface of prompting.

We train the hypernetwork end-to-end through the target generative model, minimizing a 1D-Wasserstein activation alignment loss, the same objective LinEAS [Rodriguez et al., 2025b] uses to achieve state-of-the-art per-concept steering (Figure 1). Our objective only needs unpaired data, which is far more realistic and cheaper to gather across hundreds of concepts than paired alternatives, and gives HT the bounded, interpretable strength control inherent of optimal-transport steering (Section I). Training on 618 concepts takes $\sim$64 GPU-hours, $2.7\times$ less than fitting LinEAS to each one individually. On image generation with DMD2 [Yin et al., 2024] and Nitro-1-PixArt (a one-step distillation of Chen et al. [2024a]), HyperTransport matches the strongest baseline (*i.e.,*, LinEAS) on 158 globally-applied visual concepts unseen by it at training time but individually fitted by each per-concept baseline, isolating amortization from reachability.

Decoupling concept representation from intervention prediction unlocks a novel capability closed to both traditional steering and prompting: **cross-modal conditioning**. The hypernetwork operates on encoder outputs, so any modality the encoder accepts can drive the intervention. With CLIP's multimodal space as input, this is zero-shot at deployment: a reference image can directly condition text-to-image generators with no retraining (Section 4.3).

Our contributions can be summarized as follows:

- **HyperTransport Framework.** A hypernetwork that maps concept representations to steering interventions, trained end-to-end with an optimal transport loss. New concepts, including ones never seen at training time, are served at deployment by a single forward pass (0.08s on Nitro-1-

PixArt, 0.26s on DMD2), eliminating the seconds-to-minutes per-concept optimization needed by standard steering methods. We show it can also also amortize multi-step generators like the 4-step FLUX.2 [klein] (Section 4.2.1) without any changes to our method.
- **Zero-shot Cross-modal Conditioning.** To our knowledge, HyperTransport is the first activation-steering method to take images as conditioning signal: at deployment, a reference image can directly steer text-to-image generation with no retraining (Section 4.3).
- **Comprehensive Evaluation.** Across 158 held-out concepts on DMD2 and Nitro-1-PixArt, HT matches the strongest per-concept baselines at applying the target concept while preserving the original generation nearly as well, at a fraction of their cost. At comparable inference cost to prompting, humans and VLMs pick HyperTransport roughly $2\times$ as often as prompting (Table 2).

2 Related Work

Controllable generation has evolved from fine-tuning to activation steering, with recent work exploring hypernetwork-based approaches. We position HyperTransport within this progression.

Activation Steering. Activation steering modifies internal model representations to control generation without changing the weights of the model. ITI [Li et al., 2023] uses classifier directions to steer toward truthfulness. CAA [Rimsky et al., 2024] applies contrastive activation addition for behavior control. Linear-AcT [Rodriguez et al., 2025a] and LinEAS [Rodriguez et al., 2025b] use optimal transport to learn affine transformations between source and target activation distributions. Other works address toxicity mitigation [Suau et al., 2024] and reasoning understanding [Venhoff et al., 2025]. While effective, all these methods require *per-concept optimization*, limiting scalability.

Hypernetwork-Based Adaptation. Hypernetworks [Ha et al., 2017] generate weights of a target network conditioned on an input. Recent work applies them to controllable generation: Text2LoRA [Charakorn et al., 2025] predicts LoRA [Hu et al., 2022] parameters from text, HyperLoRA [Li et al., 2025] targets portrait synthesis, and HyperDreamBooth [Ruiz et al., 2024] predicts DreamBooth-imitation LoRA offsets for Text-to-Image (T2I) subject personalization but still requires $\sim$20s of per-subject fine-tuning, whereas HyperTransport is trained end-to-end through the generator and operates in activation space with no further optimization at deployment. HyperSteer [Sun et al., 2025] is the closest prior work: it also trains a hypernetwork end-to-end to predict activation-steering parameters on the fly, sharing our goal of eliminating per-concept optimization. HT differs on four axes: (i) the steering mechanism is affine-elementwise optimal transport (HyperSteer: additive, CAA-style), inheriting the bounded strength parameter $\lambda \in [0, 1]$ of Linear-AcT/LinEAS; (ii) training uses an unpaired distributional loss (1D Wasserstein) that needs only examples of the concept, whereas HyperSteer is trained on paired prompt–target generations; (iii) the target is a T2I generator (HyperSteer: an LLM) and (iv) the hypernetwork is a simple MLP $\sim 1000\times$ smaller than the target model, whereas HyperSteer's is same-size (a constraint that is a practical limitation). Section A compares the two methods at a more fine-grained level.

Cross-Model Transfer. A growing body of work demonstrates that neural networks develop structurally similar latent spaces across architectures [Moschella et al., 2023, Huh et al., 2024]. This enables cross-model knowledge transfer: steering vectors can be transferred between models via linear mappings [Huang et al., 2025, Bello et al., 2025], and safety interventions can transfer across model families [Oozeer et al., 2025]. HyperTransport leverages this insight to learn a knowledge mapping from CLIP's structured embedding space to the activation space of the target model, effectively achieving complete inference-time decoupling from the expensive target model. Reference-image conditioning is usually handled by dedicated adapters such as IP-Adapter [Ye et al., 2023], which adds cross-attention modules trained on paired image-text data for each base model; InstantStyle [Wang et al., 2024] reuses these adapters without training and injects reference features only into selected style blocks. The HT image path, in contrast, adds no parameters beyond the text-conditioned hypernetwork and keeps the bounded $\lambda \in [0, 1]$ of optimal-transport steering.

3 Method

We present HyperTransport, a framework that learns a hypernetwork to predict interventions from arbitrary concept encodings. In this section, we first discuss concept representation (Section 3.1), then

review standard activation steering (Section 3.2), and finally introduce our approach for amortized intervention prediction (Section 3.3, Section 3.4).

3.1 Concepts and Data

A concept c is represented by a set of target sentences $\{\boldsymbol{x}_c\} \sim \mathcal{X}_c$ that embody the concept. For instance, for the concept "Van Gogh style", target sentences might include "*Bold, expressive brushstrokes, vibrant colors, and swirling textures*". To train a hypernetwork, we require a collection of concepts $\mathcal{C} = \{c_1, \ldots, c_K\}$, each with its associated target sentences $\mathcal{X}_{c_k}$.

We focus on globally-applied visual concepts (styles, palettes, environmental settings; full list in Section J), the regime in which the optimal-transport steering we amortize (Section 3.2) has been shown to work well. Since localized edits would require a different parameterization, they are outside the scope of this work. The HyperTransport hypernetwork itself does not depend on the choice of parameterization, so it can also amortize localized interventions once suitable parameterizations exist.

Additionally, we rely on a shared set of source sentences $\mathcal{X}^{\text{src}}$ that do not contain any of the target concepts. Model activations over $\mathcal{X}^{\text{src}}$ define the neutral distribution from which we steer away.

3.2 Background: Activation Steering

Intervention Framework. Given a target generative model G and desired concept c, an intervention $\mathcal{I}_{\theta_c}$ modifies the internal activations of G to steer generation toward the expression of c. Formally, the intervention produces steered activations $\boldsymbol{a}_c = \mathcal{I}_{\theta_c}(\boldsymbol{a})$. In practice, activations are grouped by layers in G so that $\boldsymbol{a} = \{\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(\ell)}, \ldots, \boldsymbol{a}^{(L)}\}$, which are intervened on by layer-specific parameters $\theta_c = \{\theta_c^{(1)}, \ldots, \theta_c^{(\ell)}, \ldots, \theta_c^{(L)}\}$. In our notation, L refers to the number of layers being intervened, not the total number of layers in G, which may not be the same.

Per-Concept Optimization. Common activation steering methods (re)optimize interventions independently for each concept, using a dataset to that end:

$$\theta_c^* = \arg\min_{\theta} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x} \sim \mathcal{X}^{\text{src}} \\ \boldsymbol{x}_c \sim \mathcal{X}_c}} \Big[\mathcal{L}_{\text{align}}\big(\mathsf{G}(\boldsymbol{x}; \theta), \mathsf{G}(\boldsymbol{x}_c)\big)\Big], \tag{1}$$

where $\mathcal{L}_{\text{align}}$ measures alignment between steered generations (using an intervention $\mathcal{I}_{\theta_c}$ and source prompts $\mathcal{X}^{\text{src}}$) and generation from target prompts $\mathcal{X}_c$. Here, $\mathsf{G}(\cdot)$ denotes any relevant output from G, *e.g.,* internal activations.

We focus on steering interventions of the affine elementwise form introduced by Linear-AcT [Rodriguez et al., 2025a]: $\mathcal{I}_{\theta_c}(\boldsymbol{a}) = (1-\lambda)\boldsymbol{a} + \lambda(\boldsymbol{w} \odot \boldsymbol{a} + \boldsymbol{b})$, with $\theta_c = [\boldsymbol{w}, \boldsymbol{b}]^\top$ and $\odot$ the elementwise product. The parameterization makes two scope choices. First, the same affine map is applied at every spatial location, which suits global, low-frequency properties (artistic styles, palettes, scene-wide attributes), not spatially localized edits. Second, the map is fitted once and applied unchanged at every denoising step. This form has been validated on one-step generative models and we additionally test it on a 4-step model (Section 4.2.1), while longer multi-step trajectories, where activation distributions shift across steps, remain out of scope. This is the operating regime of the per-concept methods we amortize, and our baselines (CAA, ITI, Linear-AcT, LinEAS) all live in it; HyperTransport holds the parameterization and intervention sites fixed across baselines, so amortization is the only variable under study. Within this functional form, the baselines differ only in how θ_c is estimated. CAA [Rimsky et al., 2024] uses contrastive activation addition over pairs, and ITI [Li et al., 2023] uses the normal of a logistic classifier separating source and target activations; both reduce to additive interventions ($\boldsymbol{w} = \mathbf{1}$). Linear-AcT [Rodriguez et al., 2025a] and LinEAS [Rodriguez et al., 2025b] estimate $\boldsymbol{w}$ and $\boldsymbol{b}$ in closed form by minimizing a sum of 1D optimal transport losses. While effective, all four require a *separate optimization for each new concept*, with per-concept latency of seconds to minutes that rules out interactive deployment. With HyperTransport we amortize this per-concept estimation (Equation (1)): a single hypernetwork maps a concept representation to θ_c in one forward pass, for any concept including those unseen at training time.

3.3 HyperTransport: Amortized Interventions

To amortize per-concept optimization costs, we introduce a trainable hypernetwork $\mathsf{H}_\psi : \mathcal{E} \mapsto \Theta$ that predicts intervention parameters from concept encodings $\boldsymbol{e}_c = \mathsf{E}(\boldsymbol{x}_c)$. The concept encoding space $\mathcal{E}$ is produced by a pre-trained, frozen encoder $\mathsf{E} : \mathcal{X} \mapsto \mathcal{E}$ mapping data to embeddings. Intervention prediction is achieved by composing the encoder and hypernetwork: $\theta_c = \mathsf{H}_\psi(\boldsymbol{e}_c) = \mathsf{H}_\psi \circ \mathsf{E}(\boldsymbol{x}_c)$.

End-to-End Training. We train H_ψ by directly optimizing the alignment objective, applying predicted interventions on G during training:

$$\psi^* = \arg\min_{\psi} \mathop{\mathbb{E}}_{\substack{c \sim \mathcal{C} \\ \boldsymbol{x} \sim \mathcal{X}^{\text{src}} \\ \boldsymbol{x}_c, \boldsymbol{x}'_c \sim \mathcal{X}_c}} \Big[\mathcal{L}_{\text{align}}\big(\mathsf{G}(\boldsymbol{x}; \mathsf{H}_\psi \circ \mathsf{E}(\boldsymbol{x}'_c)), \mathsf{G}(\boldsymbol{x}_c)\big) \Big]. \tag{2}$$

This end-to-end formulation requires the alignment loss $\mathcal{L}_{\text{align}}$ to be differentiable. Following LinEAS [Rodriguez et al., 2025b], we adopt the 1D Wasserstein distance as alignment objective. For each layer ℓ with d_ℓ neurons, the loss decomposes independently per neuron, noted W in Figure 1:

$$\mathcal{L}_{\text{align}} = \sum_{\ell=1}^{L} \sum_{j=1}^{d_\ell} W_p\left(\hat{a}_j^{(\ell)},\ a_{c,j}^{(\ell)}\right), \tag{3}$$

where $\hat{a}_j^{(\ell)}$ denotes the steered activation produced by applying the predicted intervention $\theta_c = \mathsf{H}_\psi \circ \mathsf{E}(\boldsymbol{x}'_c)$ at neuron j of layer ℓ, and $a_{c,j}^{(\ell)}$ denotes the corresponding target activation. For equal-sized distributions, the p-Wasserstein distance on a single activation dimension reduces to

$$W_p(\hat{\boldsymbol{a}}, \boldsymbol{a}) = \left(\frac{1}{n} \sum_{i=1}^{n} \left| \hat{a}_{(i)} - a_{(i)} \right|^p \right)^{1/p}, \tag{4}$$

where $\hat{a}_{(i)}, a_{(i)}$ are the order statistics of the steered and target activations. This requires only sorting (a permutation, so gradients flow to the underlying activations) and an element-wise computation, so no iterative optimal transport solver is needed. While LinEAS uses p=2, we found p=1 to perform better empirically (Table 11) and use it throughout. This optimal transport loss provides a principled, well-behaved objective that yields state-of-the-art steering performance. The name HyperTransport reflects hypernetwork-based prediction combined with this transport-based alignment objective.

Amortized Inference and Flexibility When a new steering intervention is required, even for a concept that was not seen during training, the user can provide one or more sentences that describe the desired concept. When more than one sentence is provided, we average their embeddings (encoded by E) and provide the result as input to the hypernetwork H_ψ that in a single inference pass produces all the steering weights for the target model. Since this avoids the per-concept optimization on G that LinEAS and similar methods require, the marginal cost per concept drops from hundreds of seconds to a fraction of a second (see Table 1 and Section G).

Crucially, HyperTransport requires $\mathcal{E}$ to be structured enough for H_ψ to generalize to unseen conditions. Arbitrary encodings such as one-hot could in principle be learned, but they would not support generalization. In practice, we use CLIP [Radford et al., 2021]: its contrastive vision-language pretraining aligns text and images at the level of image-wide attributes, matching our concept set. The shared text/image space also lets us condition on images instead of text (Section 4.3). Section F.1 compares several encoders on this task and finds CLIP-based variants generalize best to unseen concepts; HyperTransport is agnostic to this choice and can be retrained with any encoder.

3.4 HyperTransport Architecture

For practical deployment, H_ψ should be far smaller than G in terms of parameters and FLOPS required to compute the intervention weights. Additionally, H_ψ should be flexible enough to predict intervention parameters at various layers of G, which may have different dimensionalities and statistics. We opt for simplicity and implement H_ψ as an MLP that directly accepts inputs in $\mathcal{E}$ (full architecture in Section C). In practice, the inputs are the average encodings of target sentences $\mathcal{X}_c$, and the MLP

directly outputs the intervention parameters, in our case $\theta_c = [\boldsymbol{w}, \boldsymbol{b}]^\top$. We also experimented with a more expressive Perceiver IO [Jaegle et al., 2022] variant that handles per-sentence encodings without averaging; we report it in Section D along with training-curve comparisons (Section N), and find that the added complexity does not yield meaningful gains in our low-data regime (*i.e.,* 32 sentences for each concept, with the hypernetwork predicting interventions concept-wise).

4 Experiments

4.1 Experimental Setup

Dataset. We curate a dataset of 777 visually rich concepts through systematic LLM prompting across 10 conceptual categories, with a minimum per-category of 44 concepts ("digital methods") and maximum of 124 ("environmental settings"). For each concept, we prompt a frontier model to generate 32 representative image prompts with the explicit instruction of being suitable for text-to-image generation. We apply CLIPScore [Hessel et al., 2021] filtering to ensure semantic consistency and distinctiveness. Details and examples of the generated sentences are provided in Section B, and an analysis of the train/test concept distribution in Section M.

Models and Baselines. As target generative model G we experiment with **DMD2** and **Nitro-1-PixArt**. Both are single-step (distilled) diffusion models, a setting we inherit from LinEAS which fits activation steering research, as they do not require timestep conditioning and enable efficient forward passes during optimization. To test the effectiveness of HyperTransport, we compare against four per-concept activation steering methods: **CAA**, **ITI**, **Linear-AcT** and **LinEAS**. All steering baselines share identical experimental conditions: the same intervention positions, element-wise affine parameterization, training data, and intervention strength (λ=1); only the estimation procedure differs across methods. We intervene after all GroupNorm and LayerNorm layers of the DMD2 U-Net and after the two LayerNorms of each transformer block of Nitro-1-PixArt. Per-concept baselines use an incremental estimation variant that is strictly stronger than their original formulations (Section E). HT is trained with the p=1 alignment loss of Equation (4) (ablation in Section F.4). To test the framework beyond single-step models, we also evaluate **FLUX.2 [klein]-4B** [Black Forest Labs, 2026], a 4-step distilled flow transformer. Because prior optimal-transport steering was validated only on single-step generators, FLUX.2 [klein] tests whether HT can steer multi-step sampling trajectories without step-specific engineering, in a regime where instruction-tuned prompting is already strong. As on the other generators, we fit LinEAS on each held-out test concept. We also report **Prompting** as a reference: a mechanistically different conditioning, included for its wide adoption, its practical utility, and the increasing controllability it offers as models scale [Cheng et al., 2026].

Metrics. We assess interventions along two axes: how effectively they induce the desired concept in the generated image, and how well they preserve the content expressed in the original prompt. CLIPScore [Hessel et al., 2021] is the standard tool for both: it follows the evaluation protocol of related per-concept steering work [Rodriguez et al., 2025a,b] and of the broader text-to-image generation and concept-control literature [Gandikota et al., 2023, Betker et al., 2023]. We adopt it here because (i) it is cheap enough to run across our full evaluation grid (158 test concepts $\times$ 100 prompts $\times$ multiple methods), and (ii) it gives a strong relative signal when compared across methods that share the same intervention form and prompts, which is the setting throughout Section 4.2.1 Concretely, let $\boldsymbol{y} = \mathsf{G}(\boldsymbol{x}; \theta_c)$ be the image generated from source prompt $\boldsymbol{x}$ under intervention $\mathcal{I}_{\theta_c}$, and t_c the textual representation of concept c. We then define *Input Fidelity* $= \mathrm{CLIPScore}(\boldsymbol{y}, \boldsymbol{x})$, measuring preservation of the original prompt, and *Concept Fidelity* $= \mathrm{CLIPScore}(\boldsymbol{y}, t_c)$, measuring induction of the target concept. The two metrics are in tension: *Input Fidelity* is maximized by leaving the generation unchanged, whereas *Concept Fidelity* can be increased by over-steering at the expense of the prompt content (*e.g.,* a house steered toward *lush* until vegetation obscures the structure). Since LinEAS has been shown to achieve a favorable balance between the two [Rodriguez et al., 2025b], we choose it as the target for amortization, with the consequence that HyperTransport aims to match its performance on both metrics. Both metrics are computed with CLIP-ViT-bigG[1] [Ilharco et al., 2021] and reported as mean $\pm$ standard deviation across concepts. We additionally measure the time required to obtain a new intervention. For traditional per-concept steering methods, this is the time required to *train* the intervention. For HyperTransport, the training time is amortized since it

[1] `laion/CLIP-ViT-bigG-14-laion2B-39B-b160k` on Hugging Face.

Table 1: **HyperTransport matches LinEAS on 158 held-out concepts** (best *Concept Fidelity* on Nitro-1-PixArt; within 0.02 on *Input Fidelity* for both models), **at the cost of a single hypernetwork forward pass per new concept.** Held-out here means unseen by HyperTransport, while CAA, ITI, Linear-AcT, and LinEAS are trained from scratch on each test concept, and *Time* is their per-concept training cost. For HyperTransport, *Time* (*) is the inference cost of producing intervention weights from a concept encoding. End-to-end training of HyperTransport on the 618 training concepts takes $\sim$64 GPU-hours (8 GPUs $\times$ 8h), against $\sim$170 GPU-hours to fit LinEAS individually on the same set. Shaded rows (Unsteered, Prompting) are references and are not ranked; Prompting is a cross-paradigm reference (Section 4.2.2); training-concept results in Table 10; per-concept inference gap in Section G.

	DMD2			Nitro-1-PixArt		
Method	Time (s) ↓	*Input Fid.* ↑	*Concept Fid.* ↑	Time (s) ↓	*Input Fid.* ↑	*Concept Fid.* ↑
Unsteered	–	0.467	0.204 ± 0.025	–	0.440	0.213 ± 0.029
Prompting	–	0.446 ± 0.016	0.262 ± 0.036	–	0.416 ± 0.014	0.261 ± 0.036
CAA	641	0.285 ± 0.048	0.275 ± 0.041	138	0.334 ± 0.056	0.254 ± 0.038
ITI	1290	0.305 ± 0.043	0.275 ± 0.040	320	0.279 ± 0.060	0.254 ± 0.040
LinAcT	775	0.355 ± 0.045	$\mathbf{0.322 \pm 0.046}$	167	$\underline{0.391 \pm 0.024}$	$\underline{0.265 \pm 0.040}$
LinEAS	936	$\mathbf{0.431 \pm 0.027}$	0.278 ± 0.044	543	$\mathbf{0.398 \pm 0.023}$	0.264 ± 0.039
HT (ours)	$0.258^\star$	$\underline{0.416 \pm 0.026}$	$\underline{0.280 \pm 0.036}$	$0.078^\star$	0.390 ± 0.020	$\mathbf{0.266 \pm 0.038}$

Table 2: **HyperTransport vs. Prompting: judge evaluations.** Two-stage protocol on 158 held-out test concepts. Stage 1 evaluates individual *Input Fidelity*; only pairs where both methods succeed get to Stage 2. Stage 2 reports head-to-head *Concept Fidelity* preference shares among decisive ratings (BOTH/NONE excluded). In the human study, 19 participants give 41 decisive preferences over 96 prompt-faithful pairs (18 on DMD2, 23 on Nitro-1-PixArt). Bold for column winner. See Table 12 for unfiltered results, Table 13 for neutral breakdowns, and Section L for VLM details.

		Stage 1: Input Fidelity (% faithful)		*Stage 2: Concept Fidelity* (% preferred, head-to-head)	
Model		*VLM*	*Human*	*VLM*	*Human*
DMD2	Prompting	**83.1**	**57.7**	30.5	33.3
	HyperTransport	70.1	43.7	**69.5**	**66.7**
Nitro-1-PixArt	Prompting	**82.7**	**51.7**	37.0	21.7
	HyperTransport	71.1	39.9	**63.0**	**78.3**

allows to generate interventions for new concepts without retraining. The cost of generating such an intervention is the *inference* time on $\mathsf{H}_\psi \circ \mathsf{E}(\boldsymbol{x}_c)$.

4.2 Comparison Against Existing Conditioning Paradigms

We train HyperTransport on 618 of the 777 concepts in our dataset (an $\sim$80/20 split; Section 4.1), each represented by 32 sentences. The 618 include a validation split of 32 concepts, disjoint from those trained on, used for early stopping and for the strength sweep of Table 3. At inference, for fair comparison with the other methods, interventions are produced by conditioning on all available target sentences; we analyze sensitivity to this choice in Section F.2. We evaluate all methods on the 158 test concepts never seen during HT training; per-concept baselines are optimized from scratch on each test concept. Table 1 reports the results, analyzed below against per-concept steering (Section 4.2.1) and against prompting (Section 4.2.2).

4.2.1 HyperTransport vs. Per-Concept Baselines

Among steering methods, HyperTransport operates as intended as a high-speed approximation of LinEAS: it trades a small *Input Fidelity* decrement (within 0.02 on both models: 0.416 vs. 0.431 on DMD2; 0.390 vs. 0.398 on Nitro-1-PixArt) for matched *Concept Fidelity* (0.280 vs. 0.278 on DMD2; 0.266 vs. 0.264 on Nitro-1-PixArt) at 3600–7000$\times$ lower per-concept latency. At deployment, a



Figure 2: **Test concept conditioned generations using HyperTransport** on DMD2. On the left we show images generated without steering. Other columns show generations obtained via HT steering, for concepts (left to right) {*Caravaggio, Vector Graphics, Underwater, Neon alley, Ukiyo-e*}.

new intervention is produced in 0.258s on DMD2 and 0.078s on Nitro-1-PixArt through a single hypernetwork pass, against the 936s and 543s respectively that LinEAS requires to fit each concept from scratch. Among the other methods in Table 1, those that exceed HyperTransport on one metric underperform on the other, while others lag behind on both. The total-cost comparison over the full training set is reported in Table 1; Section G contextualizes the per-concept inference gap.

In Figure 2 we show qualitative examples of HyperTransport steerings with test concepts, showing the effectiveness of our approach at predicting steerings *never* seen during training. The leftmost image is unconditional (no steering applied to the model G), while the remaining columns show steerings for the five test concepts named in Figure 2. Rows relate to different prompts (top to bottom): *"Light bulbs of different shapes piled together with some broken", "Rubber duck floating in a city fountain among the coins.", "Asparagus spears pushing up through the soil in a garden row.", "A pomegranate split open with seeds spilling onto the surface.", "An octopus stretching its tentacles across tide pool rocks"*. More qualitative examples in Figure 15 and in the appendix (Section K). To show that this generalization is not driven by near-duplicate concepts in the random split, we correlate *Concept Fidelity* of each test concept with its cosine distance to the nearest training concept. The correlation is negligible on DMD2 (Spearman $\rho = -0.024$) and weakly negative on Nitro-1-PixArt ($\rho = -0.208$), with bin-to-bin variation within the per-concept standard deviation (Table 14, Section M.1).

Extension to new generators. Since HyperTransport amortizes LinEAS without modifying it, extending HyperTransport to a new generator only requires that LinEAS itself transfers; our hypernetwork then follows with no change to its architecture or training. We test this on the 4-step FLUX.2 [klein]-4B, where a single LinEAS affine map applied at every step steers the full trajectory. On held-out concepts, HT raises *Concept Fidelity* from 0.203 to 0.267 while *Input Fidelity* goes from 0.459 to 0.396, the same trade-off as on single-step models. Amortization again costs no concept induction. LinEAS fitted on each held-out concept reaches 0.269 *Concept Fidelity* and 0.417 *Input Fidelity*, so HT matches it on the first at a cost of 0.02 on the second. A paired comparison on 32 training concepts, which isolates amortization from generalization, agrees (Section F.3), with HT predicting the weights in 36 ms instead of 688 s of per-concept fitting on the same hardware ($\sim$19 000$\times$ faster). We observe prompting to be stronger on this generator than on single-step models, reaching 0.271 *Concept Fidelity* on held-out concepts against 0.267 for HT; on this generator, the advantage of HT is continuous strength control (Table 3).

Table 3: λ **versus prompting.** Mean CLIPScore between the generated image and a text embedding of the style over the 32 validation styles $\times$ 5 content prompts, for two generators. Slope: change in CLIPScore from 0% to 100%, from a least-squares line fit to each style and averaged over styles. ρ: Spearman correlation between requested amount and CLIPScore, averaged over styles. Bold marks the best route per generator. Qwen-Image-Edit edits the unstyled DMD2 image. Full protocol in Section I.1.

Route	0%	25%	50%	75%	100%	Slope ↑	ρ ↑
Plain DMD2 (no steering)	0.196	0.196	0.196	0.196	0.196	—	—
Prompt DMD2: "N% ⟨style⟩ style"	0.281	0.289	0.289	0.286	0.294	0.009	0.27
Qwen-Image-Edit	0.266	0.272	0.277	0.275	0.280	0.013	0.30
HyperTransport on DMD2, $\lambda{=}N/100$	0.196	0.202	0.225	0.263	0.307	**0.116**	**0.98**
Plain FLUX.2 [klein] (no steering)	0.178	0.178	0.178	0.178	0.178	—	—
Prompt FLUX.2 [klein]: "N% ⟨style⟩ style"	0.239	0.279	0.280	0.279	0.275	0.029	0.37
HyperTransport on FLUX.2 [klein], $\lambda{=}N/100$	0.178	0.181	0.187	0.214	0.248	**0.070**	**0.89**

4.2.2 HyperTransport vs. Prompting

Within the steering paradigm (Section 4.2.1), all baselines share the same intervention form and are evaluated on matched images, so CLIP-based metrics serve as a fair relative ranking. Here, prompting and steering produce qualitatively different outputs, and CLIP-based metrics are known to misalign with human perception in T2I [Otani et al., 2023]. We therefore complement the CLIP metrics with two judge protocols (human and VLM) that decouple *Input Fidelity* from *Concept Fidelity*.

User Study. We conducted a comparative user study with human judges to assess preference between images generated using prompting alone, and those steered with HyperTransport. To decouple *Input Fidelity* from *Concept Fidelity*, the study is organized in two intentionally separated stages. **Stage 1 (Prompt Faithfulness):** each judge rates 30 individual images for faithfulness to the original prompt (YES / NO), ignoring the applied artistic style. **Stage 2 (Concept Comparison):** the judge sees 15 side-by-side image pairs generated from the same prompt by the two methods, together with the concept name, definition, and up to three indicative reference images, and selects which image better displays the concept (IMAGE 1 / IMAGE 2 / BOTH / NONE). The study involved 19 human judges, yielding $n{=}285$ Stage-2 responses; in 135 of these, the judge expressed a preference between the two images (Image 1 or Image 2), and in the remaining 150 they selected BOTH or NONE. The exact instructions and a screenshot of the interface used in the study are provided in Section H.

When human judges expressed a preference between the two prompt-faithful images, they chose HyperTransport 30 out of 41 times (**73.2%**), a 2:1-or-better preference that holds for both generators (Table 2); without the Stage-1 filter, they chose HyperTransport 95 out of 135 times (70.4%, Table 12). Un-normalized breakdown including BOTH/NONE responses in Table 13, and analysis of ties in Section H.

VLM-as-a-Judge. To complement the user study at scale, we conduct a VLM-as-a-judge evaluation using InternVL2.5-26B [Chen et al., 2024b] on test image pairs, following the same two-stage protocol as above (full prompts in Section L). Among pairs where both methods produce prompt-faithful outputs and the VLM judge expressed a preference, HyperTransport is preferred head-to-head 69.5% on DMD2 and 63.0% on Nitro-1-PixArt (Table 2), corroborating the user-study findings.

Graded strength. Unlike prompting, which relies on text modifications and lacks continuous control [Cheng et al., 2026], the optimal-transport interventions we amortize offer bounded steering via $\lambda \in [0, 1]$ [Rodriguez et al., 2025a,b], which HyperTransport inherits (Section I). We evaluate fine-grained control across intensities $N \in \{0, 25, 50, 75, 100\}\%$ by (i) appending "N% ⟨style⟩ style" to the prompt, (ii) image editing with Qwen-Image-Edit [Wu et al., 2025] given "Apply N% ⟨style⟩ style to this image", and (iii) steering with $\lambda = N/100$ (Table 3; qualitative examples in Figure 14).

On both generators, prompt-based methods largely ignore the requested percentage. In fact, naming the style accounts for most of prompting's total effect (87% on DMD2, 63% on FLUX.2 [klein]). In contrast, concept strength tracks λ (Spearman $\rho = 0.98$ on DMD2 and 0.89 on FLUX.2 [klein], versus at most 0.37 for prompting and Qwen-Image-Edit), with a slope 9–13$\times$ steeper on DMD2

and $2.4\times$ on FLUX.2 [klein]. Amortized steering therefore provides graded control over concept strength, whereas prompting mostly determines only whether the style is present.

4.3 Cross-Modal Conditioning

Since HyperTransport conditions on CLIP embeddings, and CLIP by design maps images and text into the same embedding space, HyperTransport can be steered by images as well as by text without any retraining. This is the second capability HyperTransport offers that prompting cannot, because neither natural-language prompts nor traditional steering methods can take an image as a conditioning signal. We evaluate this setting by conditioning HyperTransport on the CLIP embeddings of reference images that depict the concept, even though it was trained only on text embeddings. On DMD2, steering from reference images preserves the prompt as well as steering from text does (*Input Fidelity* 0.418 vs. 0.416), and it still induces the target concept, although less strongly (*Concept Fidelity* 0.257 vs. 0.280, Figure 3 and Table 7). Under the VLM-as-judge protocol used above, image-conditioned HyperTransport is preferred 39.6% vs. 23.5% for prompting on prompt-faithful pairs (Section L), closely tracking the text-conditioned result in Table 2 and confirming that cross-modal transfer preserves concept articulation quality.

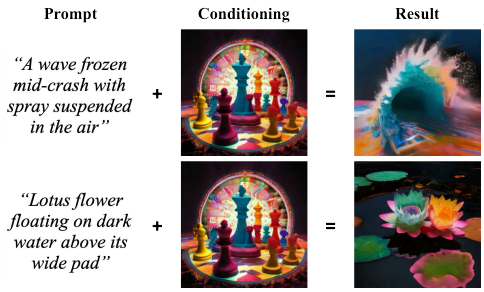

Figure 3: **Zero-shot cross-modal steering.** Left to right: prompt, conditioning image, steered output.

Qualitative Examples. Figure 3 shows steering results across conditioning modalities. Image-based conditioning successfully captures stylistic attributes from reference images while being faithful to the original prompt. We remark that, even when conditioning on images, the time required to instantiate a steering with HyperTransport is similar to that of text conditioning.

5 Conclusion

HyperTransport maps a frozen encoder embedding to the affine-elementwise intervention parameters that per-concept methods optimize from scratch, at prompting-like deployment cost. Empirically, it matches LinEAS at applying the target concept at 3600–$7000\times$ lower per-concept latency, and is preferred over prompting by both human and VLM judges at roughly 2:1. Decoupling concept representation from intervention prediction yields three properties no prior approach combines: cost-amortized steering for unseen concepts, bounded strength control $\lambda \in [0, 1]$ from optimal-transport steering, and cross-modal conditioning. HyperTransport is most useful when the concept set is large, evolving, or specified at request time; per-concept methods remain competitive on small fixed sets, and prompting suffices when text alone elicits the target. We discuss potential positive and negative societal impacts of HT in Section O.

Limitations. HyperTransport amortizes the affine elementwise operator of prior optimal-transport steering [Rodriguez et al., 2025a,b] and inherits its scope, which all per-concept baselines share. The operator applies a single map that is uniform across spatial locations and fixed across sampling steps; this suits global attributes such as style in few-step generators, but not localized edits or long sampling trajectories. This limitation is a property of the operator; the amortization framework itself can in principle predict the parameters of other operators, such as the per-location strengths of DSAS [Ferrando et al., 2025], through spatial queries in our Perceiver variant (Section D) or a dense encoder, or the nonlinear, input-gated maps of CAT [Chrabąszcz et al., 2026]. Second, training backpropagates through the generator, which is computationally prohibitive for very large models such as the 20B-parameter Qwen-Image-Edit. Third, the encoder determines which concepts can be represented and should match the target domain; since the formulation is encoder-agnostic, a domain-specific or stronger encoder than CLIP only requires retraining the hypernetwork. Finally, a single strength λ is shared across all targeted layers, with per-layer strengths left to future work.

Acknowledgments and Disclosure of Funding

We thank Louis Béthune for the internal review of this manuscript. We are also grateful to Vimal Thilak, Federico Danieli, Joao Monteiro, Amitis Shidani, Jie Chi, Nick Henderson, Tiancheng Hu, David Koski, Lucas Monteiro Paes, Anastasiia Sedova, Rita Ramos, Alice Bizeul, Alex Ferrando de las Morenas, Timur Carstensen, and Nick Apostoloff.

References

F. Bello, A. Das, F. Zeng, F. Yin, and L. Leqi. Linear representation transferability hypothesis: Leveraging small models to steer large models, 2025. URL https://arxiv.org/abs/2506.00653.

J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo, et al. Improving image generation with better captions. *Computer Science.*, 2(3):8, 2023.

Black Forest Labs. FLUX.2 [klein] 4B model card. https://huggingface.co/black-forest-labs/FLUX.2-klein-4B, 2026.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

R. Charakorn, E. Cetin, Y. Tang, and R. T. Lange. Text-to-loRA: Instant transformer adaption. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=zWskCdu3QA.

J. Chen, C. Ge, E. Xie, Y. Wu, L. Yao, X. Ren, Z. Wang, P. Luo, H. Lu, and Z. Li. Pixart-sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024a.

J. Chen, S. Xue, Y. Zhao, J. Yu, S. Paul, J. Chen, H. Cai, S. Han, and E. Xie. SANA-Sprint: One-step diffusion with continuous-time consistency distillation. In *International Conference on Computer Vision (ICCV)*, 2025.

Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024b.

E. Cheng, C. Amo Alonso, F. Danieli, A. Blaas, L. Zappella, P. Rodriguez, and X. Suau. Genctrl – a formal controllability toolkit for generative models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2601.05637.

M. Chrabąszcz, A. Szymczyk, J. Dubiński, T. Trzciński, F. Boenisch, and A. Dziedzic. Conditioned activation transport for T2I safety steering. *arXiv preprint arXiv:2603.03163*, 2026.

X. Fan, Z. Qiu, F. Wang, P. Gao, and L. Yang. Qwen-Image-Lightning. https://github.com/ModelTC/LightX2V-Qwen-Image-Lightning, 2025.

A. Ferrando, X. Suau, J. Gonzàlez, and P. Rodriguez. Dynamically scaled activation steering. *arXiv preprint arXiv:2512.03661*, 2025.

R. Gandikota, J. Materzynska, T. Zhou, A. Torralba, and D. Bau. Concept sliders: Lora adaptors for precise control in diffusion models. *arXiv preprint arXiv:2311.12092*, 2023.

D. Ha, A. M. Dai, and Q. V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017.

J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, 2021.

E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Y. Huang, C. Huang, D. Feng, W. Lei, and J. Lv. Cross-model transferability among large language models on the platonic representations of concepts. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3686–3704, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.185. URL https://aclanthology.org/2025.acl-long.185/.

M. Huh, B. Cheung, T. Wang, and P. Isola. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt. OpenCLIP. Zenodo, version 0.1. https://doi.org/10.5281/zenodo.5143773, July 2021.

A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, O. J. Henaff, M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=fILj7WpI-g.

K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2023.

M. Li, J. Chen, W. Feng, B. Li, F. Dai, S. Zhao, and Q. He. HyperLoRA: Parameter-efficient adaptive generation for portrait synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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

L. Moschella, V. Maiorca, M. Fumero, A. Norelli, F. Locatello, and E. Rodolà. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SrC-nwieGJ.

N. Oozeer, D. Nathawani, N. Prakash, M. Lan, A. Harrasse, and A. Abdullah. Activation space interventions can be transferred between large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2503.04429.

M. Otani, R. Togashi, Y. Sawai, R. Ishigami, Y. Nakashima, E. Rahtu, J. Heikkilä, and S. Satoh. Toward verifiable and reproducible human evaluation for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

N. Rimsky, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. M. Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

P. Rodriguez, A. Blaas, M. Klein, L. Zappella, N. Apostoloff, marco cuturi, and X. Suau. Controlling language and diffusion models by transporting activations. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=l2zFn6TIQi.

P. Rodriguez, M. Klein, E. Gualdoni, V. Maiorca, A. Blaas, L. Zappella, M. Cuturi, and X. Suau. LinEAS: End-to-end learning of activation steering with a distributional loss. In *Advances in Neural Information Processing Systems*, 2025b. URL https://arxiv.org/abs/2503.10679.

N. Ruiz, Y. Li, V. Jampani, W. Wei, T. Hou, Y. Pritch, N. Wadhwa, M. Rubinstein, and K. Aberman. Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models. In *CVPR*, 2024.

X. Suau, P. Delobelle, K. Metcalf, A. Joulin, N. Apostoloff, L. Zappella, and P. Rodriguez. Whispering experts: Neural interventions for toxicity mitigation in language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=2P6GVfSrfZ.

J. Sun, S. Baskaran, Z. Wu, M. Sklar, C. Potts, and A. Geiger. Hypersteer: Activation steering at scale with hypernetworks, 2025. URL https://arxiv.org/abs/2506.03292.

C. Venhoff, I. Arcuschin, P. Torr, A. Conmy, and N. Nanda. Understanding reasoning in thinking language models via steering vectors, 2025. URL https://arxiv.org/abs/2506.18167.

H. Wang, M. Spinelli, Q. Wang, X. Bai, Z. Qin, and A. Chen. Instantstyle: Free lunch towards style-preserving in text-to-image generation. *arXiv preprint arXiv:2404.02733*, 2024. URL https://arxiv.org/abs/2404.02733.

B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, N. Cooper, G. Adams, J. Howard, and I. Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024. URL https://arxiv.org/abs/2412.13663.

C. Wu, J. Li, J. Zhou, J. Lin, K. Gao, K. Yan, S. ming Yin, S. Bai, X. Xu, Y. Chen, Y. Chen, Z. Tang, Z. Zhang, Z. Wang, A. Yang, B. Yu, C. Cheng, D. Liu, D. Li, H. Zhang, H. Meng, H. Wei, J. Ni, K. Chen, K. Cao, L. Peng, L. Qu, M. Wu, P. Wang, S. Yu, T. Wen, W. Feng, X. Xu, Y. Wang, Y. Zhang, Y. Zhu, Y. Wu, Y. Cai, and Z. Liu. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. URL https://arxiv.org/abs/2508.02324.

S. Xiao, Z. Liu, P. Zhang, and N. Muennighoff. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649, 2024.

H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. URL https://arxiv.org/abs/2308.06721.

T. Yin, M. Gharbi, T. Park, R. Zhang, E. Shechtman, F. Durand, and W. T. Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024.

Y. Zhao, Y. Xu, Z. Xiao, H. Jia, and T. Hou. MobileDiffusion: Instant text-to-image generation on mobile devices. *arXiv preprint arXiv:2311.16567*, 2023.

A. Zou, L. Phan, S. Chen, J. Campbell, P. Guo, R. Ren, A. Pan, X. Yin, M. Mazeika, A.-K. Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

Appendix Contents

A Comparison with HyperSteer **15**

B Dataset Details **16**

C Implementation Details: MLP Architecture **18**

D Perceiver IO Architecture. **19**

E Baseline Implementation **20**

F Additional Results **20**

F.1 Robustness to Encoding Space 20

F.2 One-Shot Capability 21

F.3 Amortization on FLUX.2 [klein]: HyperTransport versus per-concept LinEAS . . . 22

F.4 Loss Norm Ablation: p=1 vs. p=2 23

G Per-concept inference cost **24**

H User Study **25**

I Controllability of HyperTransport **30**

I.1 λ versus prompted strength: protocol 30

J Target Concepts **33**

K Qualitative examples **35**

L VLM-as-a-Judge Evaluation **36**

M Concept Distribution Analysis **38**

M.1 Distance-Stratified Generalization 39

N Training Curves: MLP vs. PerceiverIO **40**

O Broader Impact **41**

A Comparison with HyperSteer

HyperSteer [Sun et al., 2025] shares our goal of eliminating per-concept optimization. We differ in two design choices where we use affine-elementwise optimal-transport steering instead of additive CAA-style steering, and we condition on a frozen encoder embedding rather than the base model's internal representations. Together, these choices yield four key advantages: (i) a bounded, interpretable steering strength $\lambda \in [0, 1]$ as in Linear-AcT/LinEAS, (ii) an unpaired 1D-Wasserstein loss requiring only positive exemplars, (iii) an MLP hypernetwork two to three orders of magnitude smaller than the target model rather than an equally sized copy, and (iv) support for zero-shot image conditioning in T2I models. The second point also explains why a direct empirical comparison is out of reach in our setting. HyperSteer's objective requires paired target generations, which our unpaired concept data does not provide. This is a difference in supervision regime rather than a shortcoming of HyperSteer, with HyperSteer being designed for settings where those targets exist. Table 4 presents all differences systematically.

Table 4: **HyperTransport vs. HyperSteer** along eight axes.

Axis	*HyperSteer*	*HyperTransport*
Steering form	Additive vector (CAA-style)	Affine element-wise optimal-transport map (Linear-AcT/LinEAS)
Strength knob	Unbounded scalar, set empirically	Bounded, interpretable $\lambda \in [0, 1]$
Hypernetwork	$\sim$Same size as the base model, initialized from it	MLP trained from scratch, two to three orders of magnitude smaller
Training loss	Cross-entropy against target generations from AxBench	Unpaired 1D-Wasserstein; only positive exemplars required
Conditioning	Steering prompt + base-LM activations	Frozen encoder embedding only
Coupling at inference	Coupled: cross-attention into the residual stream	Fully decoupled
Cross-modal (image) input	No	Yes, zero-shot
Domain	LLMs (text)	Few-step T2I diffusion: one-step (DMD2, Nitro-1-PixArt) and 4-step (FLUX.2 [klein])

B Dataset Details

Category	Number of Concepts
Environmental Settings	124
Art Techniques	100
Historical Periods	99
Color Treatments	80
Photography & Cinema	80
Fantasy Genres	77
Illustration Styles	64
Individual Artists	61
Artistic Movements	47
Digital Methods	44
Generic ($\mathcal{X}^{\text{src}}$)	1
Total	**777**

Table 5: Distribution of concepts across categories. The singleton "Generic" category serves as the general source one; min/max counts cited in the main body refer to the ten thematic categories.

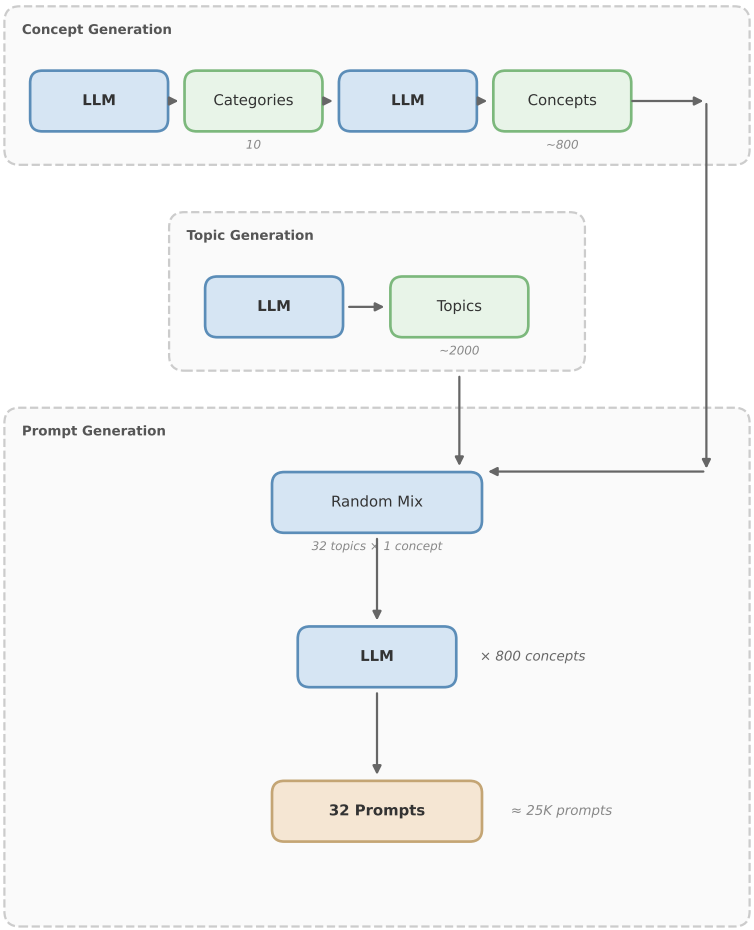

Figure 4: **Prompt generation pipeline.** Starting from 10 thematic categories elicited from a frontier LLM, we expand each into a set of visually-rich concepts (777 concepts in total) and, in a parallel pass, build a pool of $\sim$2000 generic topics. For every concept we then sample 32 topics uniformly and prompt the LLM to recombine each (concept, topic) pair into a single image-generation prompt, yielding $\sim$25K prompts.

C Implementation Details: MLP Architecture

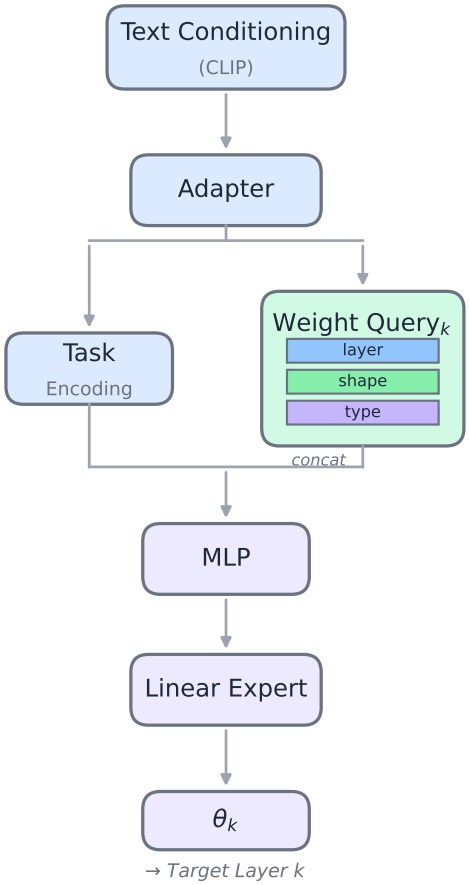

Figure 5: Architecture of the MLP-based hypernetwork. Our MLP architecture processes CLIP embeddings through multiple layers to predict intervention parameters for each layer of the target model.

Training. We train H_ψ end-to-end with AdamW [Loshchilov and Hutter, 2019] at learning rate 10^{-4} (default β_1=0.9, β_2=0.999, ϵ=10^{-8}) under a cosine schedule that dampens the learning rate by a factor of 10^{-3} at the end of training, maintaining an exponential moving average (EMA) of the hypernetwork weights with decay 0.99 and using the EMA copy at evaluation. The hypernetwork is kept in $\mathrm{float32}$; the encoder E and the target generator G are kept frozen (the target generator runs in $\mathrm{bfloat16}$). One epoch corresponds to a single pass over the 618 training concepts in random order; for each concept we sample a fresh subset of its sentences as conditioning input $\boldsymbol{x}'_c$ and as target-distribution samples $\boldsymbol{x}_c$, so the specific sentences seen for a given concept vary across epochs. Training is data-parallel: concepts are sharded across 8 H100 GPUs (per-rank batch of 1 concept), with gradients synchronized via standard all-reduce; the full 300-epoch run takes $\sim$8h ($\sim$64 GPU-hours total, see Table 1 caption).

Table 6: Hypernetwork architecture breakdown by target model. Rows above the parameter breakdown are configuration; rows below are parameter counts. Each weight query concatenates learned embeddings of the target layer's identity (dimension 128), its activation shape (64) and its type (64), as in Figure 5, followed by an output projection; these settings are shared across both targets.

	DMD2	Nitro-1-PixArt
Adapter output dim.	256	256
Weight query dim.	256	256
Task encoding dim. (adapter + query)	512	512
Intervened modules	256 (LN+GN, U-Net)	56 (LN, Transformer)
Number of weight queries	512	112
Max. parameters per intervened layer	2,560	1,152
Parameter breakdown		
Input adapters	197,120	197,120
Layer embeddings	131,584	80,064
Decoding projection (MLP)	3,674,112	3,674,112
Linear up-projection	1,313,792	591,488
Total	5,904,768	4,671,808

D Perceiver IO Architecture.

Although our main results use a simpler MLP-based architecture, we also explored implementing H_ψ with a Perceiver IO [Jaegle et al., 2022] based one, which naturally accommodates heterogeneous inputs and outputs. The added complexity yielded no meaningful gains (-1 *Concept Fidelity*, +1 *Input Fidelity*), consistent with the low-data regime of our setting (32 samples per concept/class), where the more data-hungry Perceiver IO is at a disadvantage. We therefore default to the MLP, but describe the Perceiver-based variant below for completeness, as it may prove beneficial in settings with richer or more heterogeneous data.

The architecture comprises three stages:

Cross-attention encoding. Processes an arbitrary number of concept encodings $\boldsymbol{e}_c$, enabling flexible conditioning on single or multiple examples.

Self-attention processing. Refines latent representations while conditioning on the concept encodings. The number and dimensionality of latent representations is flexible.

Specialized decoders. Generate layer-specific interventions $\{\theta_c^{(\ell)}\}$ conditioned on refined latent representations and learnable intervention queries for each target layer.

This architecture cleanly separates concept understanding (encoding) from intervention generation (decoding), allowing the hypernetwork to generalize across concepts and target layers.

The Perceiver-based hypernetwork uses:

- 256 latent tokens with dimension 512
- 4 cross-attention layers for encoding
- 6 self-attention layers for processing
- Separate decoder heads for each target layer

Table 7: **Zero-shot Cross-modal Steering** performance on test concepts using DMD2 with 32 conditionings. We combine different training modalities with conditioning ones, obtaining zero-shot settings when they are mismatched; the last column trains and conditions on both modalities. *Degenerate: the prompt is lost in favor of the concept, over-steering.

Train Modality	Text		Image		Text+Image
Condition Modality	Text	Image	Image	Text*	Text+Image
Input Fidelity ↑	0.416 ± 0.026	0.418 ± 0.030	0.414 ± 0.031	0.205 ± 0.033	0.416 ± 0.029
Concept Fidelity ↑	0.280 ± 0.036	0.257 ± 0.036	0.274 ± 0.038	0.348 ± 0.047	0.280 ± 0.038

E Baseline Implementation

Our implementation of **CAA**, **ITI**, and **Linear-AcT** is strictly stronger than their original layer-independent formulations: we estimate interventions *incrementally*, layer by layer in forward order, with each layer's objective solved on *already-intervened* activations from prior layers. This accounts for inter-layer dependencies the original formulations ignore: a correction applied at layer $\ell-1$ shifts the activation distribution observed at layer ℓ, so fitting layer ℓ to the perturbed distribution yields a tighter solution than fitting it to the unperturbed one. All baselines share the same intervention positions (Section 4.1) and the same strength $\lambda=1$.

Prompting. For each (prompt, concept) pair, images for baseline prompting and HyperTransport are generated from identical source prompts using the fixed template "{prompt} {concept}", *i.e.,* the concept name (with hyphens and underscores replaced by spaces) appended to the source prompt. Across all concepts, both methods share the same seed policy and generation settings, with no negative prompts or per-concept tuning. DMD2 uses one step at timestep 399 with the LCM scheduler, and Nitro-1-PixArt one step at timestep 400 with the default PixArt-Σ scheduler, both with guidance scale 0; FLUX.2 [klein] uses four steps with its default scheduler, guidance scale 1.0, at 512×512. Every image is generated with the same fixed seed (42), so all methods and strengths start from the same initial noise; we generate one image per prompt and resize it to 224×224 for scoring. We intentionally avoid manual prompt engineering, which fails to scale to open-ended concepts, as well as runtime adaptations, such as LLM rewriting, per-concept search, or dynamic exemplar selection, that would re-introduce the request-time overhead HyperTransport eliminates.

F Additional Results

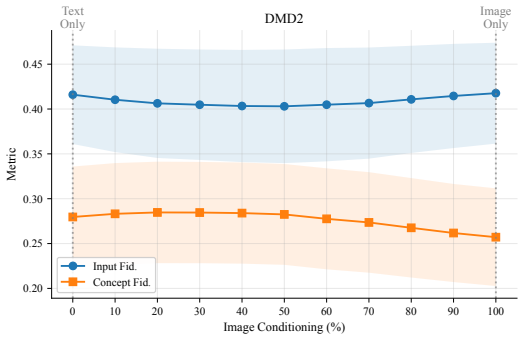

Figure 6: Varying percentage of image conditioning over 32 total for HyperTransport

F.1 Robustness to Encoding Space

We compare HyperTransport trained with several pretrained encoders to identify which embedding space is best suited for predicting interventions. We test multimodal encoders (CLIP variants and the SDXL [Podell et al., 2024] input encoder) against text-only sentence encoders: BGE [Xiao et al.,

Table 8: **Robustness to encoder choice.** Performance of HyperTransport with different pretrained encoders, same evaluation setting as Table 1. Each hypernetwork is trained for 300 epochs to steer DMD2. CLIP-based encoders achieve the best *Concept Fidelity*, with stronger generalization on test concepts.

	Train		Test	
Encoder	*Input Fid.*	*Concept Fid.*	*Input Fid.*	*Concept Fid.*
Multimodal (CLIP-based)				
SDXL Input	0.426 ± 0.032	0.288 ± 0.047	0.435 ± 0.021	0.271 ± 0.037
CLIP ViT-bigG	0.426 ± 0.032	0.288 ± 0.047	0.436 ± 0.020	0.270 ± 0.036
CLIP ViT-L	0.427 ± 0.032	0.287 ± 0.047	0.438 ± 0.020	0.269 ± 0.035
CLIP ViT-B	0.427 ± 0.032	0.287 ± 0.047	0.438 ± 0.019	0.268 ± 0.036
Text-only				
BGE Large	0.429 ± 0.032	0.283 ± 0.046	0.442 ± 0.017	0.258 ± 0.034
MPNet	0.429 ± 0.031	0.282 ± 0.046	0.443 ± 0.017	0.255 ± 0.035
MiniLM	0.430 ± 0.031	0.282 ± 0.046	0.444 ± 0.016	0.251 ± 0.035
ModernBERT Large	0.449 ± 0.017	0.243 ± 0.037	0.454 ± 0.009	0.230 ± 0.029
ModernBERT Base	0.449 ± 0.019	0.243 ± 0.035	0.454 ± 0.010	0.231 ± 0.028

2024], ModernBERT [Warner et al., 2024], MPNet[2], and MiniLM[3]. Each hypernetwork is trained for 300 epochs under identical conditions.

Table 8 reports the comparison. On training concepts, every encoder except ModernBERT performs comparably (*Concept Fidelity* from 0.282 for text-only to 0.288 for CLIP variants; ModernBERT trails at 0.243). On test concepts, CLIP-based encoders are the strongest: their *Concept Fidelity* drops only $\sim$0.02 (from 0.288 to 0.268–0.271), while text-only encoders fall further (MiniLM goes from 0.282 to 0.251, an 11% reduction). The takeaway is that CLIP's multimodal alignment yields the most structured embedding space for visual concepts, making it the best encoder choice for predicting interventions in this setting. ModernBERT shows the highest *Input Fidelity* but the weakest *Concept Fidelity*: it preserves prompts but does not induce concepts as effectively.

Concepts poorly represented in the chosen encoder's embedding space (*e.g.,* highly technical or domain-specific imagery) may yield less effective interventions. Integrating a domain-specific encoder requires only retraining the hypernetwork, with no architectural changes.

F.2 One-Shot Capability

Table 9: **One-shot comparison on DMD2.** At $N{=}1$, per-concept baselines exhibit distinct failure modes: LinEAS overfits (high *Concept Fidelity* but degraded *Input Fidelity*), Linear-AcT retreats to a conservative intervention, and CAA/ITI approach the unsteered baseline. HyperTransport degrades gracefully, maintaining competitive performance on both metrics. This ablation measures degradation under a shrinking conditioning budget, not absolute ranking. Since prompting has no N, it has no degradation to measure and it is excluded.

Method	*Input Fid.* ↑	*Concept Fid.* ↑
Unsteered	0.467	0.205 ± 0.025
LinEAS	0.328 ± 0.058	0.307 ± 0.050
Linear-AcT	0.358 ± 0.044	0.296 ± 0.041
CAA	0.451 ± 0.013	0.215 ± 0.029
ITI	0.459 ± 0.007	0.207 ± 0.026
HyperTransport	0.435 ± 0.044	0.247 ± 0.053

A practical advantage of HyperTransport is its effectiveness with minimal conditioning data. While in our main comparison (Table 1) we condition on 32 target sentences that describe each concept, this section runs an ablation to assess how robust HyperTransport is to a reduction in the number of conditioning examples. At deployment the 32 descriptions are produced automatically from a single

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[3] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

concept label (Figure 4), so N=32 is the setting a user actually gets. In Table 9 we compare the performance at $N = 1$ versus that at $N = 32$, where $N = 1$ represents the minimal one-shot scenario (see Figure 7 for intermediate values $N = 2, 4, 8, 16$). We observe that reducing the conditioning samples from 32 to 1 increases the fidelity to the prompt (*Input Fid.*) at the expense of a moderate reduction in conditioning capability (11.5% drop in *Concept Fid.*, from 0.280 at N=32 in Table 1 to 0.247). This trade-off is expected: with fewer conditioning examples, the model has less information to constrain the concept representation, allowing it to remain more faithful to the original prompt while slightly reducing its ability to capture the full nuances of the target concept. Despite this expected degradation, the results show that HyperTransport can instantiate new steerings, even on unseen test concepts, with access to only one sentence, which is not achievable with any of the traditional steering methods in the literature.

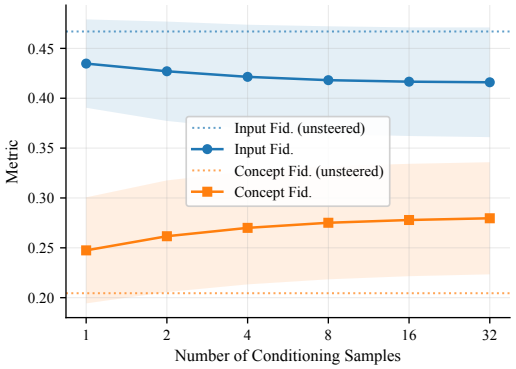

Figure 7: HyperTransport on DMD2 as a function of the number of conditioning sentences N, from N=1 (one-shot) to N=32; dotted lines mark the unsteered model.

Table 10: Full results on training and test concepts. Training concepts are those used during HyperTransport amortization; test concepts are held out. Per-concept baselines are trained individually on each concept in both settings. *Unsteered* is the generator without intervention. Shaded rows (Unsteered, Prompting) are references and are not ranked. Both metrics are CLIPScores computed with CLIP-ViT-bigG [Ilharco et al., 2021] (mean ± std across concepts).

		Train		**Test**	
Model	*Method*	*Input Fid.*	*Concept Fid.*	*Input Fid.*	*Concept Fid.*
DMD2	*Unsteered*	0.467 ± 0.000	0.204 ± 0.023	0.467 ± 0.000	0.204 ± 0.025
	Prompting	0.447 ± 0.014	0.261 ± 0.037	0.446 ± 0.016	0.262 ± 0.036
	CAA	0.286 ± 0.046	0.278 ± 0.042	0.285 ± 0.048	0.275 ± 0.041
	ITI	0.307 ± 0.046	0.279 ± 0.042	0.305 ± 0.043	0.275 ± 0.040
	Linear-AcT	0.358 ± 0.043	$\mathbf{0.321 \pm 0.047}$	0.355 ± 0.045	$\mathbf{0.322 \pm 0.046}$
	LinEAS	$\mathbf{0.432 \pm 0.024}$	0.280 ± 0.043	$\mathbf{0.431 \pm 0.027}$	0.278 ± 0.044
	HyperTransport (ours)	0.404 ± 0.040	0.295 ± 0.046	0.416 ± 0.026	0.280 ± 0.036
Nitro-1-PixArt	*Unsteered*	0.440 ± 0.000	0.212 ± 0.025	0.440 ± 0.000	0.213 ± 0.029
	Prompting	0.417 ± 0.015	0.262 ± 0.038	0.416 ± 0.014	0.261 ± 0.036
	CAA	0.338 ± 0.053	0.258 ± 0.036	0.334 ± 0.056	0.254 ± 0.038
	ITI	0.280 ± 0.057	0.256 ± 0.042	0.279 ± 0.060	0.254 ± 0.040
	Linear-AcT	0.391 ± 0.024	0.267 ± 0.037	0.391 ± 0.024	0.265 ± 0.040
	LinEAS	$\mathbf{0.399 \pm 0.023}$	0.267 ± 0.042	$\mathbf{0.398 \pm 0.023}$	0.264 ± 0.039
	HyperTransport (ours)	0.387 ± 0.025	$\mathbf{0.274 \pm 0.041}$	0.390 ± 0.020	$\mathbf{0.266 \pm 0.038}$

F.3 Amortization on FLUX.2 [klein]: HyperTransport versus per-concept LinEAS

Because LinEAS optimizes parameters directly for each target concept, measuring the pure cost of amortization requires evaluating concepts that both methods have seen. Testing on held-out concepts would conflate amortization error with generalization error, which is instead the focus of Table 1.

We therefore fit LinEAS individually on a fixed sample of 32 concepts from the FLUX.2 [klein] training split, and score HT on the exact same concepts at $\lambda = 1$ using the protocol from Table 1. All comparisons are paired by concept.

Method	*Input Fid.* ↑	*Concept Fid.* ↑
Unsteered	0.459	0.200
LinEAS, fitted per concept	**0.414**	0.292
HyperTransport (ours)	0.370	**0.303**
Paired difference (HT − LinEAS), n=32, ± 95% CI	-0.044 ± 0.012	$+0.011 \pm 0.010$

On the 32-concept subset, HT is on par with the per-concept baseline it amortizes at inducing concepts ($+0.011$, 95% CI $[0.001, 0.021]$; higher on 21 of 32 concepts), at a cost of 0.044 in *Input Fidelity*. We note three important methodological points.

First, because this evaluation uses training concepts by construction, it isolates amortization fidelity specifically and does not evaluate generalization. Second, with n=32, the 95% margin of error is approximately ± 0.01. Third, all runtimes are measured directly on identical hardware using an NVIDIA A100 GPU. A per-concept LinEAS fit on FLUX.2 [klein] requires 688 ± 3 s across 1000 optimization steps (n=32). In contrast, HT predicts the intervention parameters in only 36.4 ± 0.5 ms (n=20).

This corresponds to a speedup of roughly $19\,000\times$ on the same GPU.

F.4 Loss Norm Ablation: p=1 vs. p=2

We compare HyperTransport trained with p=1 (L1) vs. p=2 (L2) in the Wasserstein alignment loss of Equation (3), evaluated on the same 158 held-out test concepts as Table 1.

Table 11: **L1 vs. L2 alignment loss.** HyperTransport trained with p=1 vs. p=2 in Equation (4), evaluated on the 158 held-out test concepts of Table 1 for both DMD2 and Nitro-1-PixArt.

Model	*Loss*	*Input Fid.* ↑	*Concept Fid.* ↑
DMD2	p=1 (L1)	0.416 ± 0.026	0.280 ± 0.036
	p=2 (L2)	0.437 ± 0.020	0.271 ± 0.036
Nitro-1-PixArt	p=1 (L1)	0.390 ± 0.020	0.266 ± 0.038
	p=2 (L2)	0.402 ± 0.016	0.255 ± 0.034

G Per-concept inference cost

Table 1 reports two distinct cost quantities: the per-concept training cost of each baseline (CAA, ITI, Linear-AcT, LinEAS) and HyperTransport's per-concept inference cost. For per-concept baselines, this is the time to fit a new intervention via optimization on the target model: 936s on DMD2 and 543s on Nitro-1-PixArt for LinEAS, the strongest baseline. For HyperTransport, this is the time of a single hypernetwork forward pass: 0.258s on DMD2 and 0.078s on Nitro-1-PixArt. The nominal ratios are $\sim 3600\times$ and $\sim 7000\times$ respectively.

These two quantities are not directly comparable: HyperTransport's inference cost follows a one-off training phase (~ 64 GPU-hours over 618 concepts; see Table 1 caption), whereas the baselines' time is incurred at every new concept. The per-concept inference comparison is informative when novel interventions must be produced at deployment, e.g. under interactive latency constraints or when the concept vocabulary is large, evolving, or unknown ahead of time.

Beyond this marginal cost, the end-to-end path from a single user phrase adds one LLM call ($\leq$2s) to expand it into the 32 descriptions (Figure 4), giving $936/(0.258 + 2) \approx 414\times$ on DMD2 and $543/(0.078 + 2) \approx 261\times$ on Nitro-1-PixArt, which we report as $\sim 400\times$ and $\sim 260\times$. The expansion is a shared cost since the per-concept baselines are conditioned on the same 32 sentences and degrade without them (Table 9), so charging the 2s to both sides leaves the ratio at $\sim 415\times$ and the comparison is insensitive to the accounting choice. Against prompting the comparison is of a different kind since prompting pays at every generation, whereas this expansion is incurred once per concept and θ_c is reused indefinitely.

H User Study

In this section we provide more information about the user study described in Section 4.2.2: the exact instructions given to participants and a screenshot of the interface.

The study is organized in two stages that are intentionally separated so that judgments on *Input Fidelity* do not bias judgments on *Concept Fidelity*: a single-image *prompt faithfulness* stage, followed by a pairwise *concept comparison* stage. The same two-stage protocol is later re-used, in adapted form, in the VLM-as-a-judge experiment (Section L).

The instructions provided to each user were as follows:

User Study Instructions

This study has **TWO stages**. Please complete them in order.

Stage 1: Prompt Faithfulness
In this stage you will see 30 images, one at a time. Each image was generated from a text prompt using a specific style (concept).

Your task: Judge whether the image is **faithful** to the prompt.

- Focus on the semantic content: are the objects, actions, relationships, and counts correct?
- **Ignore the artistic style** — the image has been intentionally stylized.
- Respond with **YES** (faithful) or **NO** (not faithful).

Please judge each image independently.

Stage 2: Concept Comparison
In this stage you will see 15 pairs of images side by side. Both images in each pair were generated from the **same prompt** and should display the **same concept/style**. They were generated by two different methods.

For each pair you will see:

- The concept name and its definition
- Up to 3 example images showing what the concept looks like
- **Note:** These examples are purely indicative and should not be taken as a strong reference. They were found automatically and some of them might not be representative of the concept.
- Two generated images (Image 1 and Image 2)

Your task: Which image better displays the concept?

- **Image 1**: if Image 1 better displays the concept
- **Image 2**: if Image 2 better displays the concept
- **BOTH**: if both images display the concept equally well
- **NONE**: if neither image displays the concept well

An example of how each request was shown to human judges is provided in Figure 8. The choice of T2I model for each pair was randomized, with the constraint that the study contained an approximately equal number of generations from each model: in total 450 pairs, 221 with DMD2 and 229 with Nitro-1-PixArt. Each judge rated 15 pairs sampled from this pool, yielding n=285 Stage 2 responses across 19 human judges; in 135 of these, the judge picked one of the two images (IMAGE 1 or IMAGE 2), and in the remaining 150 they selected BOTH or NONE.

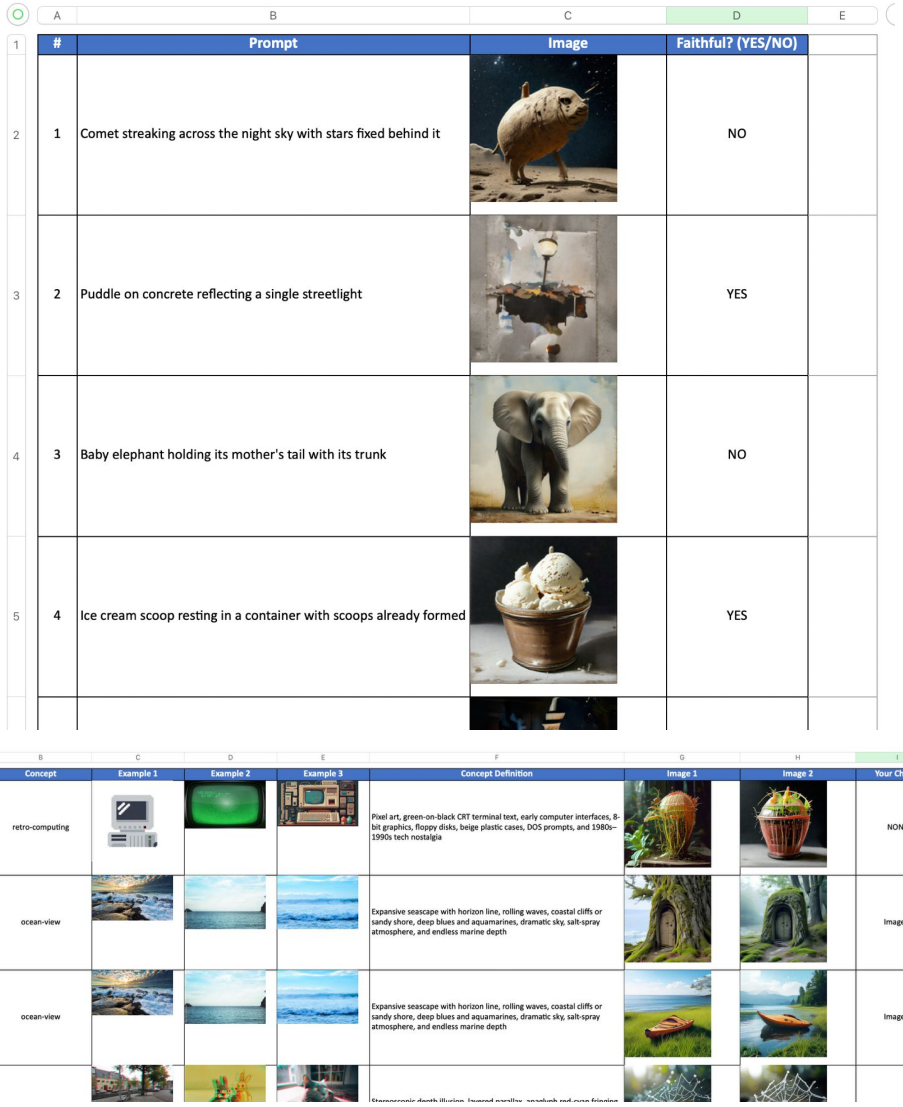

Figure 8: Examples of the user-study UI. **Top: Stage 1 (Prompt Faithfulness).** Each row shows the prompt and a generated image; the participant marks YES or NO for prompt-faithfulness. **Bottom: Stage 2 (Concept Comparison).** Each row shows the concept name, three indicative example images, the concept definition, and two generated images (Image 1 and Image 2, presented in random order from prompting or HyperTransport); the participant picks Image 1, Image 2, BOTH or NONE.

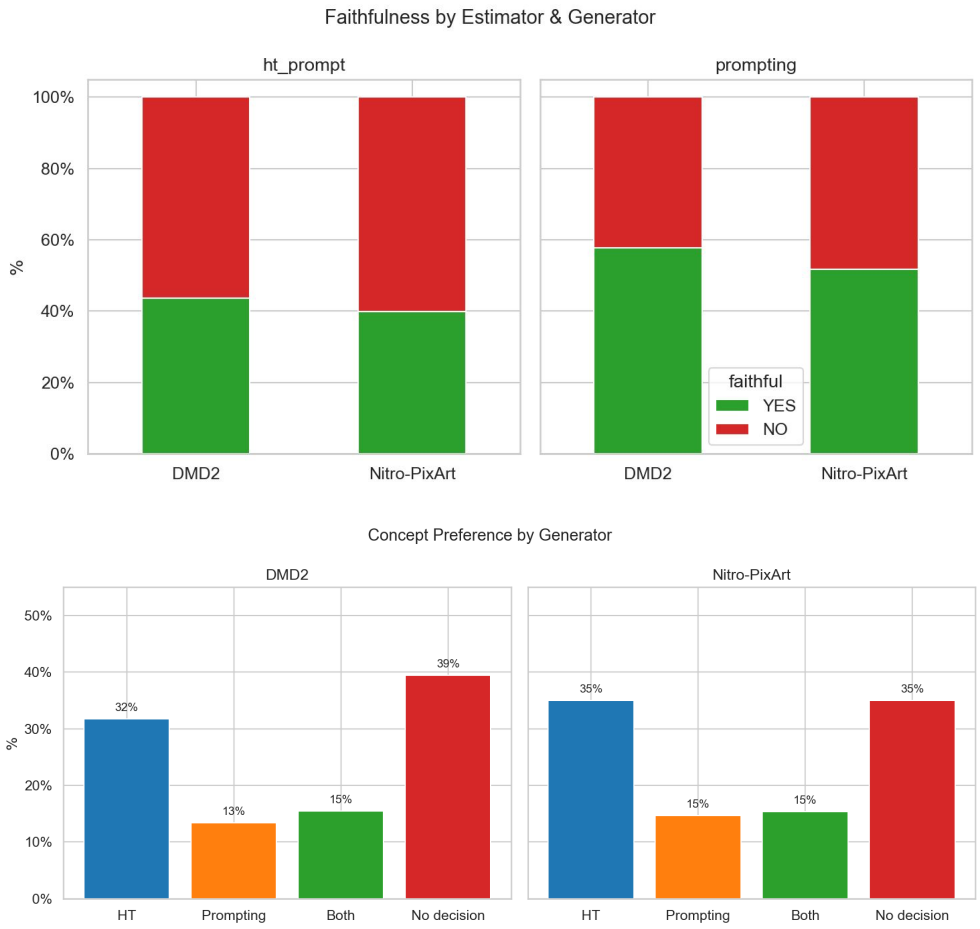

Figure 9: **Two-stage user study of HyperTransport generations** (n=285 Stage-2 responses from 19 human judges). **Top: Stage 1, Input Fidelity (per generator).** Prompting is slightly more often judged faithful than HyperTransport on both generators (aggregate 54.7% *vs.* 41.8%; McNemar p=0.0001, Cohen's h=-0.261, a small effect). **Bottom: Stage 2, Concept Fidelity (per generator).** Distribution of the four response options across all 285 pairs. Without the Stage-1 filter of Table 2, restricted to the 135 pairs where the judge picked one of the two images, HyperTransport is chosen 70.4% of the time ($95/135$; clustering by participant, a cluster-robust GEE logistic model gives 70.2%, 95% CI $[61.5\%, 77.6\%]$, p=2×10^{-5}); per generator, 70.3% on DMD2 and 70.4% on Nitro-1-PixArt (Fisher interaction p=1.0).

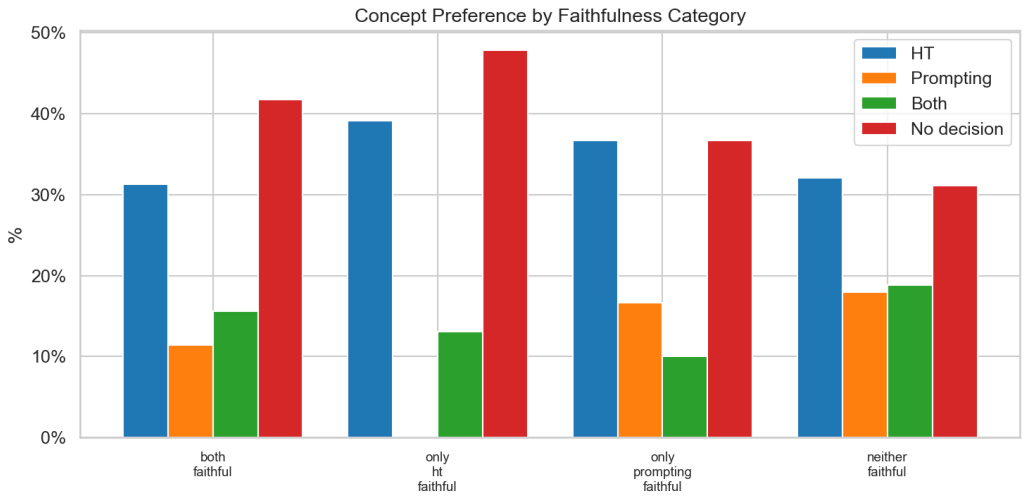

Figure 10: **Stage 2 (concept-fidelity) broken down by Stage 1 (input-fidelity) outcomes** (human annotations). HyperTransport is preferred across all faithfulness categories, including pairs where only the *prompting* image was Stage-1 faithful (HyperTransport 36.7% vs. prompting 16.7%). This is consistent with HyperTransport's advantage being independent of input fidelity. "No Decision" rates remain high across categories.

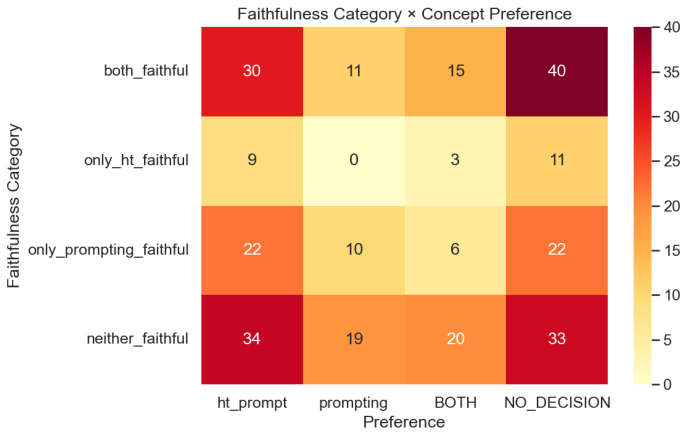

Figure 11: **Joint distribution of Stage-1 faithfulness category × Stage-2 response** (raw counts; rows are Stage-1 categories, columns are Stage-2 responses). When the human judge picks one of the two images (HyperTransport or prompting columns), HyperTransport is chosen more often than prompting in every row, including *only prompting faithful* (22 *vs.* 10): HyperTransport's concept-fidelity advantage holds even when only the prompting image is judged input-faithful. BOTH and NO DECISION are reported per row but excluded from the head-to-head share.

Stage-1 filter. Table 2 compares concept fidelity only on pairs where both images follow the user's prompt meaning that an image that ignores the prompt cannot win or lose the comparison, whichever method produced it. In the human study, this filter keeps 96 of the 285 responses; among the 41 decisive ones (i.e., where the judge expresses a preference between the two methods without answering BOTH or NONE), HyperTransport is chosen 30 times (73.2%, binomial p=0.004). Without the filter (Table 12), the conclusion is similar (70.4%), and HyperTransport is preferred for both generators and both judges (VLM and human) in either case.

Table 13: **Full breakdown of VLM and human evaluation, including BOTH/NONE responses.** Companion to Table 2: same Stage-1 numbers, but Stage-2 here is *un-normalized* and reports the raw share of each response category (Prompting / HyperTransport/ *No preference*), so each Stage-2 column sums to 100 within a model. "*No preference*" aggregates BOTH and NONE responses (Human) and unmatched pairs (VLM). For VLM, Stage 2 is over both-faithful pairs (n_{DMD2}=7827, $n_{Nitro-1-PixArt}$=7377); for Human, over all 285 Stage-2 responses, no Stage-1 filter (150 of which were BOTH/NONE).

		Stage 1: Input Fidelity (% faithful)		*Stage 2: Concept Fidelity* (% of responses)	
Model		*VLM*	*Human*	*VLM*	*Human*
DMD2	Prompting	**83.1**	**57.7**	20.5	13.4
	HyperTransport	70.1	43.7	**46.7**	**31.7**
	No preference	–	–	32.8	54.9
Nitro-1-PixArt	Prompting	**82.7**	**51.7**	22.8	14.7
	HyperTransport	71.1	39.9	**38.8**	**35.0**
	No preference	–	–	38.4	50.4

Table 12: **Stage-2 preference without the Stage-1 filter.** Head-to-head share of *Concept Fidelity* preferences among ratings that picked one of the two methods, over all pairs (VLM) and all 135 decisive human responses (64 on DMD2, 71 on Nitro-1-PixArt). Filtered counterpart in Table 2.

Model		*VLM*	*Human*
DMD2	Prompting	29.2	29.7
	HyperTransport	**70.8**	**70.3**
Nitro-1-PixArt	Prompting	37.9	29.6
	HyperTransport	**62.1**	**70.4**

Robustness analyses. The unfiltered 70.4% preference rate for HyperTransport is robust to alternative tie handling: scoring BOTH as half-credit to each method yields 65.4%, and scoring all ties (BOTH and NONE) as half-credit yields 59.6%, both well above the 50% null. Ties favor neither method by definition, so excluding them does not create the ratio. Keeping all 150 in the pool, HyperTransport is the sole winner 95 times against 40, i.e. 33.3% vs. 14.0% of all 285 responses. Across the 135 pairs in which the human judge picked one of the two images, the winner is judged Stage-1 faithful in 44.4% of cases and the loser in 46.7% (essentially equal), inconsistent with a faithfulness-driven explanation. The cross-generator interaction is non-significant (Fisher p=1.0): HyperTransport's preference rate is 70.3% on DMD2 and 70.4% on Nitro-1-PixArt. The conclusion also does not depend on treating the 135 responses as independent. Cluster-robust GEE logistic models give 70.2% by participant (95% CI $[61.5\%, 77.6\%]$, $p=2\times10^{-5}$), 69.9% by concept ($[60.4\%, 78.0\%]$), and 69.5% by prompt ($[60.1\%, 77.5\%]$); a crossed random-effects GLMM (participant + concept) agrees, and a participant cluster bootstrap gives $[62.0\%, 78.2\%]$. All intervals sit well above the 50% no-preference line.

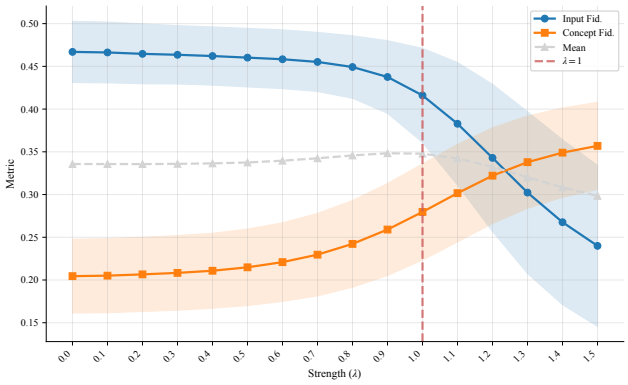

Figure 12: **Controllability of HyperTransport.** Increasing the conditioning parameter λ increases the conditioning (*Concept Fid.*) while minimally diverging from the prompt (*Input Fid.*), achieving the optimal compromise (measured as arithmetic mean) for $\lambda = 1$.

I Controllability of HyperTransport

Unlike prompting, where modifiers such as "more" or "less" yield qualitative, unpredictable shifts [Cheng et al., 2026], steering methods expose an explicit strength parameter λ. However, λ is typically unbounded and therefore difficult to interpret. Following LinEAS [Rodriguez et al., 2025b], HyperTransport adopts an optimal transport formulation that grounds λ in a predictable and interpretable $[0, 1]$ range: $\lambda = 0$ means no intervention, $\lambda = 1$ means full transport from the source to the target distribution, while every intermediate value corresponds to a partial transport that progressively closes the gap between the source and the target distribution.

This provides an intuitive "knob" for control. As shown in Figure 12, increasing λ monotonically and smoothly increases *Concept Fidelity* at a gradual *Input Fidelity* cost up to λ=1. Out of curiosity, we also tested values of $\lambda > 1$. As expected, *Concept Fidelity* continues to increase but at the cost of a drop in *Input Fidelity*. Computing the arithmetic mean between the two metrics shows that indeed the optimal value is reached for $\lambda = 1$.

Figure 13 shows qualitative examples of this controlled generation on DMD2: we start at $\lambda = 0$ (no conditioning) and progressively increase λ to 1.5, with the $\lambda = 1$ generation highlighted in red.

I.1 λ versus prompted strength: protocol

Table 3 asks whether prompting can replicate graded strength control through explicit modifiers. **Metric.** Mean CLIP cosine similarity between the generated image and a text embedding of the style, averaged over validation styles $\times$ 5 content prompts; all rows are scored by the same evaluator, CLIP-ViT-bigG. **Rows.** On DMD2, we compare (1) no steering, as a reference; (2) appending "N% ⟨style⟩ style" to the prompt; (3) editing the unstyled image with Qwen-Image-Edit [Wu et al., 2025] at its default 50 inference steps ("Apply N% ⟨style⟩ style to this image."); (4) sweeping HyperTransport's λ on interventions predicted from the style's text embedding. Routes (1), (2) and (4) are then repeated on FLUX.2 [klein]-4B at its 4-step, 512×512 generation settings, steered by a hypernetwork trained on that generator and intervening after every normalization layer of its transformer (42 modules); the plain and prompting rows use the unmodified generator. The 32 validation styles are unseen by both hypernetworks. Base prompt and seed are held fixed across routes, and the initial noise is fixed per sample across strengths, making λ=0 a paired control against λ=1 — it recovers the plain value exactly on both generators. **Sweep.** The prompting and editing rows request $N \in \{0, 25, 50, 75, 100\}\%$. The FLUX.2 [klein] λ row is measured on that same 5-point grid; the DMD2 λ row was swept at $\lambda \in \{0, 0.1, \ldots, 1\}$ and its 25% and 75% entries are linearly interpolated from the neighboring measured points. **Slope.** Mean over styles of a per-style least-squares fit of cosine similarity against the requested amount on $[0, 1]$, *i.e.,* the change in cosine from no style to full style, fitted on each style's own measured points rather than the interpolated columns. **Rank correlation.** ρ is the Spearman correlation between the requested amount and cosine similarity, computed on each style's measured points and averaged over styles. Restricting the DMD2

Figure 13: Examples of controlled conditioning using DMD2. The first image for each generation corresponds to $\lambda = 0$ (*i.e.,* no conditioning), the last to $\lambda = 1.5$. The image highlighted in red corresponds to $\lambda = 1$.

Prompt: *Hands holding a melting ice cream.*
Concept: *Expressionist*



Prompt: *Hailstones collecting in a bucket. More falling around.*
Concept: *Contemporary Art*



Prompt: *A four-leaf clover among thousands of three-leaf ones.*
Concept: *Contemporary Art*



Prompt: *A tiny mouse peeking out from behind a wheel of cheese.*
Concept: *Art Deco*



Prompt: *Coins stacked in uneven towers of different currencies.*
Concept: *Contemporary Art*



λ row to five evenly spaced points of its eleven, like the prompting rows, leaves it at 0.98–0.99, so the denser grid does not inflate it.

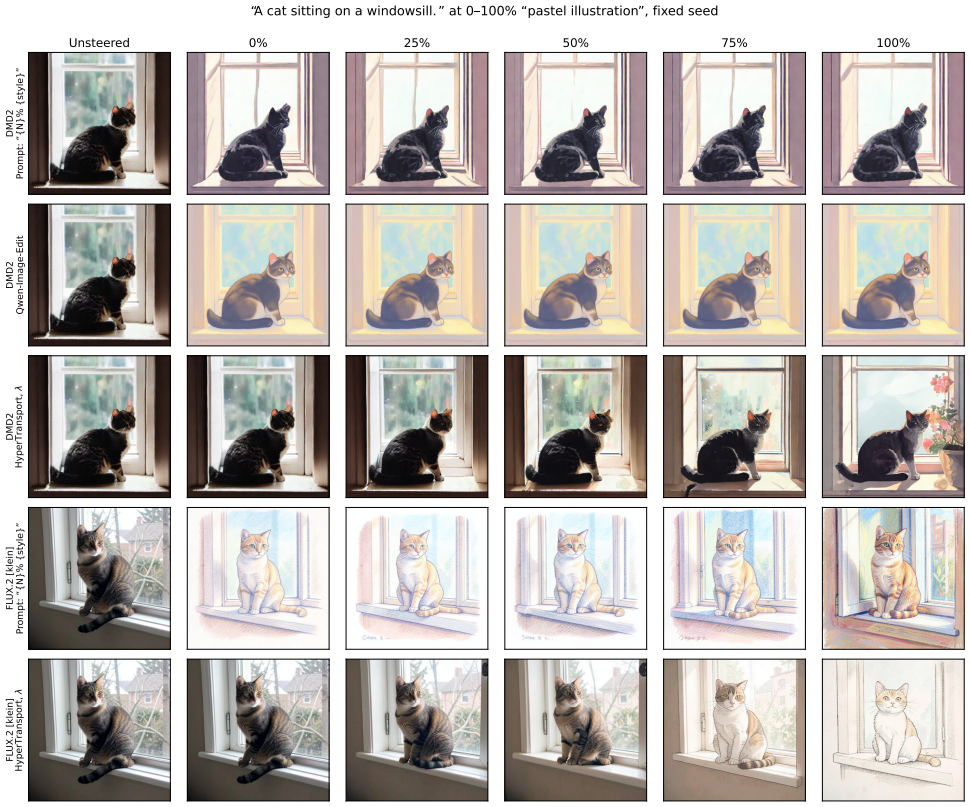

Figure 14: **Visualizing graded strength control.** Generations for "A cat sitting on a windowsill" across requested intensities from 0% to 100% of *pastel-illustration* with a fixed random seed. Along each row, only the requested strength varies. Prompting and editing barely respond to the requested percentage. The style is already applied at 0%, so the percentage acts as an on-off switch (FLUX.2 [klein] prompting changes only at 100%). In contrast, HT's λ moves the image gradually away from the unsteered baseline on both generators, with most of the change above $\lambda = 0.5$. At $\lambda = 0$, generations are identical to the unsteered baseline by construction. This concept was selected systematically to maximize the slope margin among concepts displaying monotonic trends on both generators. At extreme intervention levels ($\lambda = 1$), global distribution matching can occasionally transfer object content alongside style (the flower pot in the DMD2 λ=1 image).

J Target Concepts

Below is the full list of 777 concepts used in our studies, organized across 10 categories. The dataset is randomly split into 618 training and 158 test concepts stratifying across categories.

Art Techniques (100). abstract-drawing, acrylic, action-painting, airbrush, alcohol-technique, assemblage, automatic-drawing, blending, blind-contour, brush-painting, calligraphy, ceramic-art, charcoal, chiaroscuro, collage, color-mixing, continuous-line, contour-drawing, crackle-medium, cross-hatching, cut-and-paste, dead-coloring, decoupage, divisionism, drip-painting, dry-brush, dry-brush-technique, dry-media, egg-tempera, engraving, etching, fabric-collage, fan-brush, figure-drawing, finger-painting, fluorescent-paint, forced-perspective, fresco, gel-medium, gestural-painting, gesture-drawing, glazing, glowing-paint, gouache, graphite, green-underpainting, impasto, ink-wash, interference-paint, lettering, life-drawing, line-art, linocut, lithography, masking-fluid, matte-painting, metallic-paint, milk-paint, mixed-media-collage, mosaic, neo-impressionism, oil-glazing, oil-painting, optical-illusions, optical-mixing, palette-knife, palette-knife-painting, paper-collage, papier-mache, perspective-illusion, perspective-tricks, photomontage, plastic-wrap-technique, plein-air-painting, pour-painting, relief-painting, resist-technique, salt-technique, sand-texture, scratch-art, screen-print, sketch-art, speed-painting, splattering, sponge-technique, spray-painting, stained-glass, stippling, tempera, texture-paste, thumb-painting, underpainting, unmixed-colors, wash-painting, watercolor, wax-painting, wet-brush, wet-media, wet-on-wet, woodcut.

Artistic Movements (47). abstract-expressionist, art-deco, art-nouveau, arte-povera, avant-garde, baroque, bauhaus, conceptual-art, constructivist, contemporary-art, cubist, dadaist, de-stijl, expressionist, fauvism, fluxus, futurist, geometric-abstract, gothic-revival, impressionist, installation-art, kinetic-art, land-art, magic-realism, mannerism, minimalist, modernist, muralism, neo-baroque, neo-expressionist, neoclassical, neue-sachlichkeit, op-art, photorealistic, pointillism, pop-art, post-impressionist, psychedelic, realism, renaissance, rococo, romanticism, street-art, suprematist, surrealist, transavantgarde, ukiyo-e.

Color Treatments (80). analogous-colors, atmospheric-glow, bleach-bypass, bleached, blue-highlights, blue-hour-light, bold-contrast, brass-finish, bronze-tint, cinematic-grade, cinematic-lighting, color-grading, color-pop, color-wash, complementary-colors, cool-palette, cool-shadows, copper-patina, copper-tint, cyberpunk-lighting, deep-shadows, desaturated-tones, digital-display, dramatic-lighting, duotone, earth-palette, earth-tones, faded, film-look, fluorescent, gentle-highlights, god-rays, golden-hour-light, golden-tint, gradient-map, holographic-color, iridescent, jewel-tones, metallic-tones, midnight-colors, monochromatic-colors, monochrome, muted, muted-pastels, neon, ocean-palette, orange-teal, pastel, pearlescent, psychedelic-color, rainbow-gradient, retro-computing, retro-filter, retro-tint, rich-contrast, rim-lighting, rust-tint, saturated-colors, screen-glow, screen-reflection, selective-color, sepia, sharp-contrast, silver-shine, silver-tint, soft-lighting, split-toning, subtle-contrast, sunrise-colors, sunset-colors, synthwave-colors, twilight-colors, vaporwave-aesthetic, vibrant, vintage-color, vivid-highlights, vivid-saturation, volumetric-lighting, warm-palette, warm-shadows.

Digital Methods (44). algorithmic-art, ambient-light, ascii-art, atmospheric-haze, augmented-reality, back-lighting, bitmap, blender-render, cel-shading, cgi, chrome-reflection, color-fringing, color-temperature, cyber-aesthetic, data-visualization, digital-illustration, digital-painting, flat-shading, fractal, generative-art, glass-distortion, glass-material, glitch-art, high-poly, hologram-effect, holographic-art, isometric-view, low-poly, motion-graphics, net-art, octane-render, particle-effects, photo-manipulation, photorealistic-render, pixel-art, procedural-art, ray-tracing, toon-shader, unreal-engine, vector-art, vector-graphics, virtual-reality, voxel-art, wireframe-render.

Environmental Settings (124). abandoned-city, abandoned-factory, airport, alien-planet, alleyway, ancient-ruins, ancient-temple, arctic, arctic-station, art-gallery, autumn-forest, ballroom, bamboo-grove, beach-house, bridge, calm-lake, canyon, castle-ruins, cathedral, cave, cherry-blossom, coastal, concert-hall, conservatory, coral-reef, countryside, crystal-cave, crystal-palace, deep-forest, deep-sea, desert, desert-camp, desert-landscape, digital-dystopia, downtown, enchanted-garden, farmland, floating-city, floating-islands, flower-field, foggy-moor, forest, forest-clearing, fortress, frozen-lake, futuristic-city, ghost-town, glacier, grand-hotel, greenhouse, highway, hot-spring, industrial-zone, jungle-camp, kelp-forest, laboratory, lava-tube, library, lighthouse, lighthouse-cliff, lush-jungle, magical-forest, medieval-castle, misty-valley, monastery, moon-base, mountain-lodge, mountain-vista, mountainous, mystical-grove, neon-alley, neon-cityscape, neon-metropolis, neon-tokyo, night-market, oasis, observatory, ocean-view, old-town, orchard, railway, rainforest, rainy-street, research-station, retro-diner, rooftop, rushing-river, sand-dunes, savanna, seaport, shelter, shopping-arcade, sky-city, snowy-peaks, space, space-station, speculative-future, spring-meadow, steampunk-factory, stormy-sea, stratosphere, street-festival, subterranean, suburb, summer-beach, summer-cottage, suspension-bridge, theater-stage, train-station, tropical, tundra, tunnel, underwater, underwater-city, urban, vineyard, vintage-diner, volcanic, volcano-observatory, watchtower, wilderness, windy-cliff, winter-cabin, winter-wonderland.

Fantasy Genres (77). 80s-aesthetic, academic-aesthetic, adventure-game, alternate-history, arthurian, atom-punk, aurora-aesthetic, biopunk, brutalist-aesthetic, cosmic-aesthetic, cottagecore, crystal-core, cybernetic-style, cyberpunk, cyberpunk-aesthetic, dark-electronic, dark-fantasy, decora-kei, dieselpunk, dreamscape, dreamy-electronic, dystopian, epic-fantasy, fairy-core, fairy-tale-retelling, fantasy-realm, folkloric, forest-core, future-funk, galactic-aesthetic, gem-core, gothic-fantasy, gothic-lolita, harajuku, hard-sci-fi, heroic-fantasy, high-fantasy, holographic-aesthetic, iridescent-aesthetic, liminal-space, magical-realism, marble-aesthetic, medieval-fantasy, military-aesthetic, modern-fantasy, multiverse, mystical, nature-spirituality, neo-noir, neon-noir, ocean-core, otherworldly, parallel-universe, paranormal-romance, prismatic-aesthetic, psychedelic, psychedelic-tech, retro-futurism, romantic-gothic, science-fantasy, sky-core, soft-sci-fi, solarpunk, space-opera, speculative-fiction, spiritual, steampunk, superhero-style, surreal-fantasy, sword-and-sorcery, synthwave, tech-noir, time-travel, toy-render, urban-fantasy, utopian, vaporwave.

Historical Periods (99). 1920s, 1950s, 1960s, 1970s, 1980s, 1990s, ai-era, ancient-egypt, ancient-greece, ancient-rome, art-deco-era, art-nouveau-period, atomic-age, atomic-future, aztec, baroque-era, belle-epoque, biotech-era, bronze-age, byzantine-empire, chrome-age, chrome-era, classical-antiquity, climate-change-era, cold-war-era, colonial, computer-age, computer-revolution, counter-reformation, cyberpunk-era, dark-ages, digital-age, dot-com-era, early-medieval, early-renaissance, edwardian, electric-age, enlightenment, folk-tradition, galactic-era, gilded-age, gothic-period, great-depression, grunge-nineties, hellenistic, heritage-style, high-medieval, high-renaissance, incan, industrial-era, internet-age, interplanetary-age, iron-age, jazz-age, jet-age, late-medieval, machine-age, mannerism, mars-colonization, mayan, mid-century-modern, millennium-era, modernist-period, nanotech-era, neoclassical-period, neolithic, neon-eighties, neon-era, northern-renaissance, nuclear-age, paleolithic, pandemic-era, post-war-1950s, prehistoric, psychedelic-seventies, quantum-age, renaissance-era, renewable-energy-era, retro-futuristic, retro-gaming, rococo-period, roman-empire, romanesque, romantic-period, smartphone-era, social-media-era, social-network-era, space-age, space-exploration, steam-age, streaming-era, swinging-sixties, television-age, victorian, viking, vintage-computing, vintage-revival, wartime-1940s, y2k-era.

Illustration Styles (64). 3d-illustration, abstract-illustration, acrylic-illustration, album-cover, anime-style, architectural-drawing, art-deco-poster, book-cover, botanical-illustration, brushwork, caricature, cartoon, character-design, charcoal-illustration, children's-book, colored-sketch, comic-book, concept-art, costume-design, creature-design, decorative, digital-sketch, disney-style, editorial-illustration, environment-design, fairy-tale, fantasy-illustration, fine-art, flat-colors, geometric-illustration, gouache-illustration, graphic-novel, illustrated-book, ink-drawing, isometric-illustration, japanese-animation, kawaii-style, korean-pop-style, magazine-illustration, manga, map-illustration, marker-sketch, maximalist, miyazaki-style, modern-illustration, newspaper-cartoon, oil-illustration, pastel-illustration, pen-sketch, pixar-style, prop-design, refined-sketch, retro-illustration, rough-sketch, sci-fi-illustration, sketch-style, studio-ghibli-style, traditional-illustration, travel-poster, vector-illustration, vehicle-design, vintage-poster, watercolor-illustration, web-illustration.

Individual Artists (61). alex-katz, alphonse-mucha, amy-sherald, andrew-wyeth, annie-leibovitz, anselm-kiefer, artgerm, banksy, basquiat, bourgeois, caravaggio, cindy-sherman, cy-twombly, da-vinci, dali, david-hockney, diane-arbus, edward-hopper, francis-bacon, georgia-okeefe, gerhard-richter, greg-rutkowski, haring, helen-frankenthaler, helmut-newton, hokusai, irving-penn, jasper-johns, kahlo, kara-walker, katsuhiro-otomo, kehinde-wiley, klimt, kusama, lucian-freud, magritte, makoto-shinkai, mamoru-hosoda, michelangelo, monet, munch, norman-rockwell, peter-halley, picasso, pollock, rembrandt, richard-avedon, richard-diebenkorn, robert-rauschenberg, rothko, satoshi-kon, takashi-murakami, takeshi-obata, tim-burton, van-gogh, vermeer, vivian-maier, warhol, wayne-thiebaud, wes-anderson, willem-de-kooning.

Photography & Cinema (80). 16mm-film, 35mm-film, 360-photography, 3d-photography, 4k-cinema, 70mm-film, 8k-cinema, 8mm-film, aerial-photography, analog-photography, anamorphic, architectural-photography, astrophotography, black-and-white, blurred-background, bokeh-background, boudoir-photography, camera-obscura, cinematic, cross-processed, cross-processing, digital-cinema, digital-photography, documentary, double-exposure, drone-photography, event-photography, fashion-photography, film-grain, film-noir, film-photography, fisheye-lens, food-photography, glamour-photography, headshot-photography, high-speed-photography, imax, infrared-photography, instant-film, instant-photography, interior-photography, kodachrome, landscape, large-format, light-painting, lomographic, lomography, long-exposure, macro-photography, medium-format, microscopy, milky-way-photography, multiple-exposure, nature-photography, pinhole-camera, polaroid-style, portrait, product-photography, sepia-tone, slide-film, star-trails, steel-wool-photography, stereoscopic, still-life-photography, stop-motion, street-photography, studio-lighting, super-8-film, technicolor, telephoto-lens, tilt-shift, time-lapse, ultraviolet-photography, underwater-photography, vintage-photography, vr-photography, wedding-photography, wide-angle-lens, wildlife-photography, x-ray-photography.

K Qualitative examples

We show here qualitative examples of different conditioning methods on DMD2.

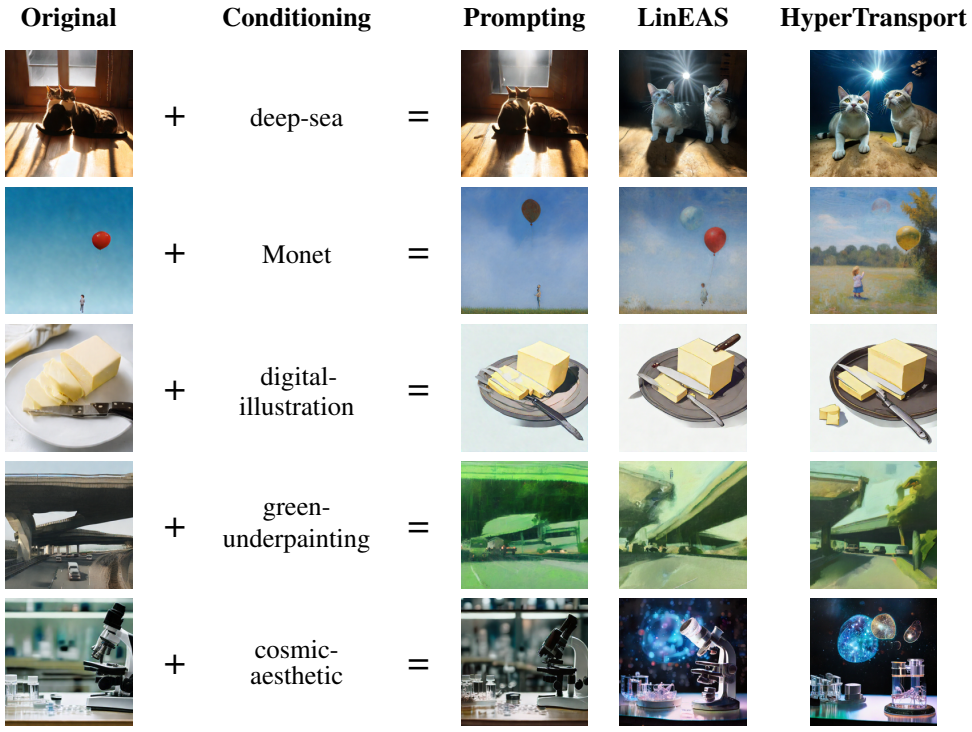

Figure 15: Qualitative comparison of conditioning methods on DMD2. Original image (generated with the original prompt), conditioning concept, and outputs from three methods. The last row shows a failure case of HyperTransport, where the microscope has been removed in favor of conditioning more on the style.

L VLM-as-a-Judge Evaluation

We evaluate HyperTransport in a VLM-as-a-judge setting using InternVL2.5-26B [Chen et al., 2024b] on $33{,}400$ test image pairs (text-conditioned and image-conditioned, summed across both models) covering all test concepts. A two-stage protocol decouples input fidelity from concept fidelity. Unlike the human judges, who saw the concept definition and up to three reference images, the VLM receives the concept name only.

Stage 1: Input fidelity. Each image is judged independently for faithfulness to the original prompt, regardless of applied style. The VLM prompt reads:

```
``Look at this image.  It was generated from this prompt:  ``{PROMPT}''
Reason first.  Is this image faithful to the prompt?  Note:
The image has been stylized using the ``{CONCEPT}'' style.  When
judging faithfulness, focus on whether the semantic content
(objects, actions, relationships, counts, etc.)  matches the prompt,
regardless of the artistic style applied.  Respond with ONLY: YES or
NO''
```

Stage 2: Concept fidelity. Images are shown side-by-side and the judge selects which better represents the target concept:

```
``Look at this image showing two pictures arranged {FIRST} and
{SECOND}.  Both pictures should display this concept/style:
``{CONCEPT}'' Reason first.  Which image better displays the concept?
Respond with ONLY: {FIRST}, {SECOND}, BOTH (if equally good), or
NONE (if neither displays the concept well)''
```

To control for position bias, each comparison runs twice with swapped positions; only position-agnostic answers are retained. We report concept preference among prompt-faithful pairs.

Results. Prompting more often produces prompt-faithful images (82.9% vs. $\sim 71\%$), but with weaker concept induction. Among the faithful subset ($15{,}204$ text-conditioned pairs; $15{,}426$ image-conditioned pairs), HyperTransport is preferred 42.9% vs. 21.6% for text-conditioning, and 39.6% vs. 23.5% for image-conditioning.

Relation to the user study. The VLM-as-a-judge protocol above mirrors the two-stage design of the user study (Section H), which serves as the reference protocol: Stage 1 corresponds to the single-image *Prompt Faithfulness* stage ("Does the image generated from the prompt faithfully depict the original prompt?"), and Stage 2 to the pairwise *Concept Comparison* stage ("Which image better displays the concept?"). The VLM prompts above are the per-instance instructions presented to the judge model; the instructions presented to human judges are reproduced verbatim in Section H.

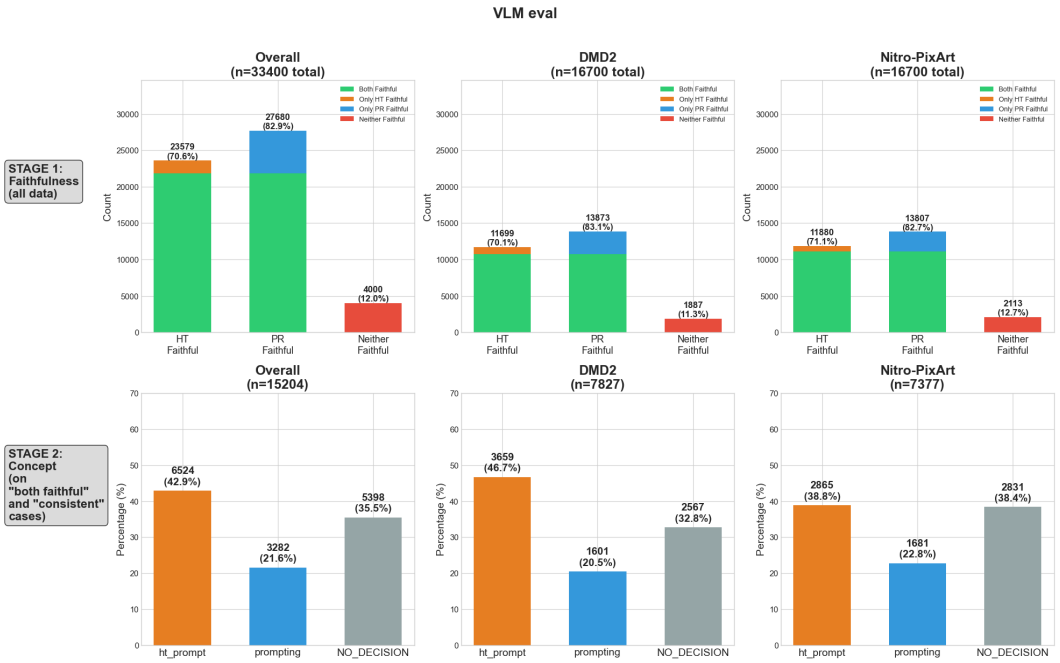

Figure 16: **Two-stage evaluation of images generated via text-conditioned HyperTransport (Test).** As expected, input-fidelity trades-off concept-fidelity. **Top: Stage 1, Input Fidelity (images judged one by one).** Distribution of faithfulness judgments across all samples and per model (DMD2, Nitro-1-PixArt). Prompting produces images that remain more faithful to the original input ($\sim$83% *vs.* $\sim$70% for HyperTransport). **Bottom: Stage 2, Concept fidelity (images judged in pairs).** Among samples where both models produce faithful outputs, HyperTransport generations are favored (42.9% *vs.* 21.6%). Results are consistent across the two models, with DMD2 yielding slightly stronger preferences for HyperTransport than Nitro-1-PixArt (46.7% *vs.* 38.8%).

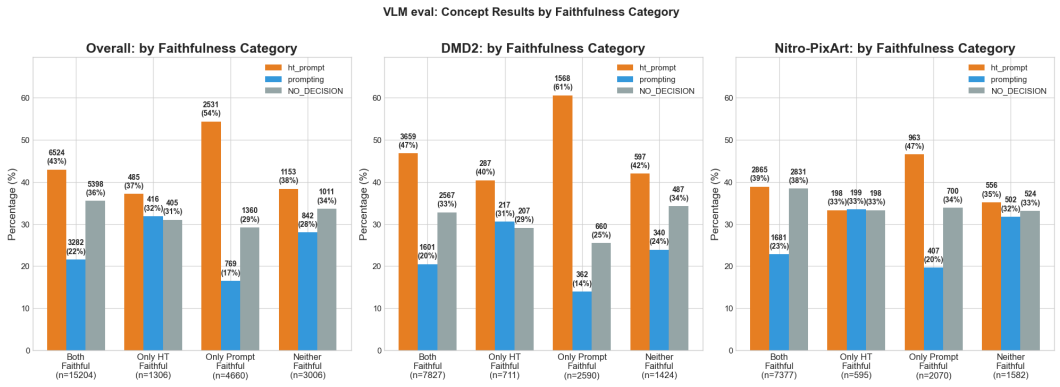

Figure 17: **Stage 2 (concept-fidelity) broken down by results of Stage 1 (input-fidelity)** (HT text-conditioned, Test). HyperTransport generations are preferred irrespectively of their "faithfulness" category, confirming that the concept-fidelity advantage of HyperTransport is not merely a byproduct of input-faithful samples. Results hold across models.

M Concept Distribution Analysis

To verify that the random concept split is balanced and that test concepts are not simply interpolations of training concepts, we compute mean CLIP embeddings per concept over each sentence and analyse the resulting cosine-distance structure between splits.

Split balance. For every concept we summarise its pairwise CLIP cosine distances to the rest of the dataset by the 25th, 50th and 75th percentile, and plot the distribution of those quantiles across concepts in each split (Figure 18). Train, val and test occupy near-identical regions of each quantile axis, indicating that the random split does not concentrate the easier or harder concepts in any one split.

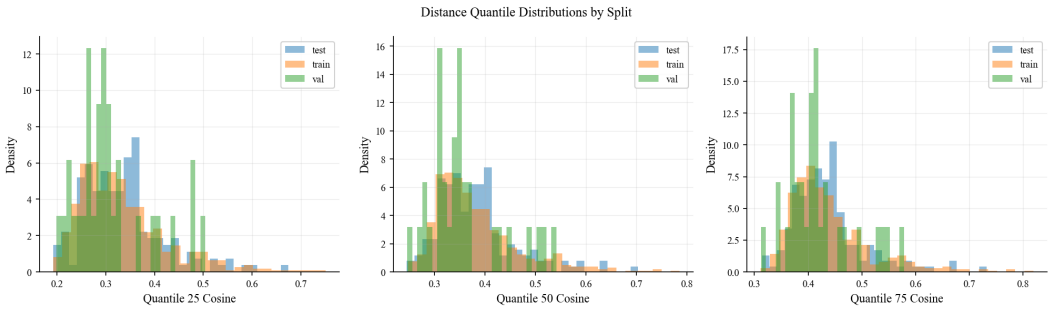

Figure 18: **The random split is balanced across train/val/test.** For each concept we compute pairwise CLIP cosine distances to the rest of the dataset and summarise them by the 25th, 50th and 75th percentile (left, middle, right panel). The three splits occupy nearly identical regions of each quantile axis, indicating that the random split does not concentrate the easier or harder concepts in any one split.

Difficulty proxy. We then quantify the difficulty of a test concept as the mean CLIP cosine distance of its embedding to all training-concept embeddings. Figure 19 (left) validates this proxy: it correlates strongly (Pearson $r{=}0.80$) with the minimum distance to the closest training neighbor, so it captures both global novelty and nearest-neighbor proximity rather than averaging them away.

Difficulty distribution. Figure 19 (right) shows the distribution of this difficulty across test concepts: it spans $\sim$0.27 to $\sim$0.70 (mean 0.394), with substantial mass beyond 0.4, i.e., many test concepts lie far from every training neighbor. Section M.1 analyzes directly if performance varies across this range, with results showing it does not.

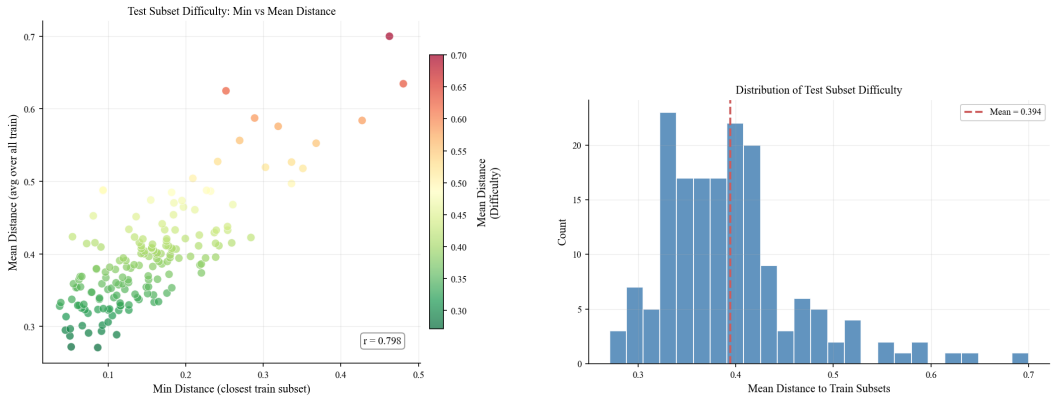

Figure 19: **Test concepts span a wide range of distances from training. Left:** per-test-concept mean cosine distance to all training concepts (y) vs. minimum cosine distance to the closest training concept (x); the strong correlation (r=0.80) confirms that the mean-distance proxy reflects both nearest-neighbor and global novelty. **Right:** distribution of test-concept difficulty (mean cosine distance to training concepts); difficulties span $\sim$0.27 to $\sim$0.70 (mean 0.394), with substantial mass at distances well above 0.4. Performance stratified by this distance is reported in Section M.1.

M.1 Distance-Stratified Generalization

For each test concept we compute its *minimum* cosine distance to the nearest training concept and correlate *Concept Fidelity* against it. On DMD2 the two are essentially uncorrelated (Spearman $\rho = -0.024$); on Nitro-1-PixArt the correlation is weakly negative ($\rho = -0.208$). Table 14 reports the binned version, splitting the test concepts into distance terciles. Bin-to-bin variation stays within the per-concept standard deviations, including the widest case (0.276 to 0.257 on Nitro-1-PixArt), and *Input Fidelity* is flat throughout. Near-duplication thus has no significant effect on our tables.

These bins also suggest three distinct regimes. **Near** tests consistency under lexical variation, where different descriptions denote the same or a closely related attribute at a high level but not in embedding space, and where any near-duplicates such as *dry-brush* and *dry-brush-technique* necessarily fall — mapping such pairs to the same intervention is the behaviour we want, since a user should not have to describe a concept exactly as the model expects it. **Medium** is where we expect recombination, a novel concept sharing attributes with several training concepts without paraphrasing any. **Far** tests clear extrapolation beyond the hypernetwork training distribution.

Table 14: **Held-out performance does not degrade with distance from training.** Test concepts split into terciles by minimum CLIP cosine distance to the nearest training concept. HyperTransport on held-out test concepts; both metrics are CLIPScores computed with CLIP-ViT-bigG [Ilharco et al., 2021].

Model	*Distance*	*Input Fid.*	*Concept Fid.*
DMD2	Near	0.417 ± 0.024	0.282 ± 0.034
	Medium	0.419 ± 0.026	0.274 ± 0.036
	Far	0.412 ± 0.030	0.284 ± 0.039
Nitro-1-PixArt	Near	0.390 ± 0.020	0.276 ± 0.038
	Medium	0.391 ± 0.022	0.263 ± 0.041
	Far	0.390 ± 0.019	0.257 ± 0.032

N Training Curves: MLP vs. PerceiverIO

Both the MLP and PerceiverIO architectures converge with comparable loss profiles when trained with our 1D Wasserstein alignment loss, confirming the stability of the objective across architectures. The MLP achieves marginally better downstream metrics and was selected for simplicity.

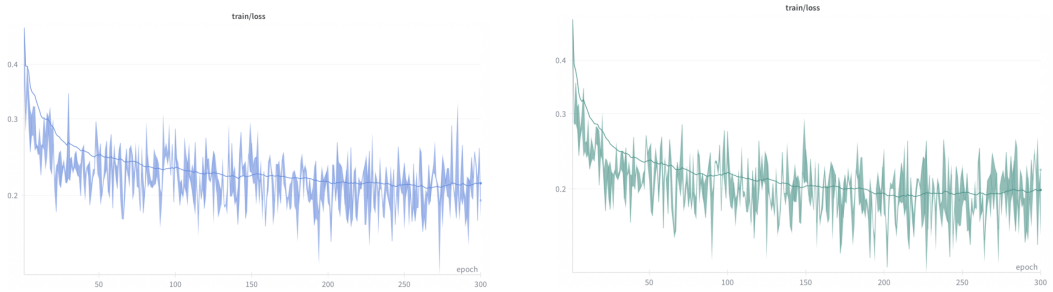

Figure 20: **Training curves of Perceiver and MLP architectures.** Training loss over 300 epochs for the Perceiver (left) and MLP (right). Both architectures converge to similar final loss values ($\sim$0.2), with the MLP exhibiting slightly lower variance in later epochs.

O Broader Impact

HyperTransport targets the cost and brittleness of concept-level conditioning of generative models. Because we discuss potential societal effects in the NeurIPS checklist, we expand on them here.

Positive impacts. By amortizing per-concept optimization into a single forward pass (Section G), HyperTransport lowers the compute and engineering overhead of steering frozen generators. We see three concrete benefits. (i) *Accessibility:* concept-level control becomes affordable for users and labs that cannot run a per-concept fitting pipeline at request time, broadening who can adapt open-weight generative models. (ii) *Safety-relevant conditioning:* the same machinery that steers a model *toward* a target concept can steer it *away* from one, giving practitioners a lightweight handle for content control on top of existing model-level safeguards. (iii) *Auditable knobs:* the bounded strength parameter $\lambda \in [0, 1]$ from optimal-transport steering [Rodriguez et al., 2025b] (Section I) makes the magnitude of the intervention explicit and graded, which is easier to log, threshold, and review than free-form prompt edits.

Negative impacts and intended-use harms. HyperTransport shares the misuse surface of any concept-steering method applied to a pretrained generator. We highlight two categories that follow directly from the technology working as intended. (i) *Style and persona impersonation:* cheap, request-time conditioning on a target concept can be used to mimic the style of a specific artist or the appearance of a specific public figure without consent. The base generator's existing safeguards remain the primary line of defense, but HyperTransport does not add new ones. (ii) *Amplification of biased or unsafe content:* if the concept set used to train or query HyperTransport contains harmful, stereotyping, or unsafe descriptions, the hypernetwork will faithfully amortize them. Because HyperTransport is encoder-agnostic, biases in the underlying CLIP embedding (or any replacement encoder) are inherited by the predicted interventions.

Harms from incorrect operation. When HyperTransport underfits a concept, the failure mode is benign: the generator falls back toward its unconditional behavior, which is the same distribution a user would have sampled without steering. When HyperTransport overfits or distorts a concept, the resulting samples are visibly off-distribution rather than convincingly deceptive, which limits (but does not eliminate) the risk of weaponizing failure cases for disinformation.

Mitigations. Several properties of the method support mitigation rather than requiring external machinery. The bounded $\lambda \in [0, 1]$ control surface lets deployers cap intervention strength globally or per concept. Training data is generated entirely from LLM prompts (Section B); curating, filtering, or red-teaming this concept set is a direct lever on what HyperTransport will be able to steer toward at deployment. Because HyperTransport does not modify the base generator's weights, all model-level safeguards (safety filters, watermarking, identity blocklists) of the underlying generator continue to apply unchanged. We do not release a new generative model or scraped dataset, so the additional misuse surface introduced by this work is bounded by the artifacts of the base models we build on.

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 Section 1 enumerate our three contributions—amortized steering for unseen concepts, zero-shot cross-modal conditioning, and a comprehensive evaluation against per-concept baselines and prompting—and these are directly supported by the results in Section 4 (Tables 1 and 2). The scope is also explicit: globally-applied visual concepts (Section 3.1) on one-step T2I generators (DMD2, Nitro-1-PixArt) and the 4-step FLUX.2 [klein], with an affine-elementwise parameterization inherited from the steering literature (Section 3.2).

   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: A dedicated Limitations paragraph at the end of Section 5 discusses the two scope choices inherited from the affine elementwise parameterization (global low-frequency edits only; few-step generators only: one- and 4-step models are tested, not long sampling trajectories), the dependence of concept coverage on the chosen encoder, and the use of a single strength λ across all intervened layers.

   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 does not include formal theorems or proofs; its contribution is empirical.

   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: Sections 3 and 4.1 describe the intervention parameterization, intervention sites (post-normalization layers in DMD2 and Nitro-1-PixArt), the p=1 Wasserstein training objective, the dataset construction pipeline, and the per-concept baselines under matched conditions. Section C reports the hypernetwork architecture and training recipe (AdamW at lr $= 10^{-4}$ with cosine schedule, EMA decay 0.99, hypernetwork in $\mathrm{float32}$, 300 epochs over the 618 training concepts, data-parallel across 8 H100 GPUs); Section D reports the alternative Perceiver IO variant; Section E the incremental layer-by-layer fit used for enhanced baselines; Section B the dataset construction; Section J the full concept list; Section M the train/test split analysis. Code release plans are described in the next item.

   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 will be released at `https://github.com/apple-aiml-research/ml-hypertransport`. To support reproducibility in the meantime, the paper provides full architecture, training, and dataset-construction details (Sections B to C, 4.1, E and J).

   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: Section 4.1 reports the dataset, intervention positions (all GroupNorm/LayerNorm layers of the U-Net for DMD2, the two LayerNorms of each transformer block for Nitro-1-PixArt), the affine-elementwise parameterization, and the intervention strength (λ=1) shared across all steering methods. Section C reports the hypernetwork hyperparameters (AdamW at lr $= 10^{-4}$ with default (β_1=0.9, β_2=0.999, ϵ=10^{-8}) under a cosine schedule, EMA decay 0.99, hypernetwork in $\mathrm{float32}$ over a $\mathrm{bfloat16}$ target, 300 epochs,

data-parallel across 8 H100 GPUs with concepts sharded across ranks and a per-rank batch of 1 concept). Section B reports dataset statistics (777 concepts, 32 prompts each, CLIPScore-filtered) and the 80/20 train/test split. Section E reports the incremental layer-by-layer enhanced fitting variant used for the baselines.

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: Table 1 reports CLIPScore-based metrics as mean $\pm$ one standard deviation, with variability taken across the 158 held-out test concepts (the same 100 prompts over 158 concepts are scored under every method, so ranges are directly comparable). User-study uncertainty is reported as a 95% confidence interval on HyperTransport's head-to-head preference rate ($95/135$ decisive responses; a cluster-robust GEE logistic model by participant gives 70.2%, 95% CI $[61.5\%, 77.6\%]$), with paired binary outcomes assessed by McNemar's test, the cross-generator interaction by Fisher's exact test, and robustness checked under alternative tie-handling (Section H).

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: Training and all experiments were run on NVIDIA H100 GPUs, except the FLUX.2 [klein] cost comparison of Section F.3, which was timed on an NVIDIA A100. Table 1 and its caption report per-concept baseline training times, HyperTransport's amortized inference cost, total HyperTransport training cost ($\sim$64 GPU-hours: 8 GPUs $\times$ 8h), and the equivalent total for fitting LinEAS on the same set ($\sim$170 GPU-hours);

Section G contextualizes the per-concept inference cost. The compute reported corresponds to the experiments reported in the paper.

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 paper against the NeurIPS Code of Ethics. (i) Third-party assets (CLIP, DMD2, Nitro-1-PixArt, and the encoders in Section F.1) are used in accordance with their permissive MIT/Apache-2.0 licenses (item 12). (ii) The training data is generated entirely from LLM prompts (Sections B and 4.1); no web-scraped data, real-person data, or personally identifiable information is collected or used at any stage. (iii) The user study (Sections 4.2.2 and H) involved volunteer participants from within the research community, not crowdsourced workers; the protocol, risk profile, and oversight are discussed in items 14 and 15. (iv) Foreseeable positive and negative societal impacts are discussed in Section O.

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: HyperTransport reduces the cost and brittleness of concept-level conditioning of generative models, enabling lighter-weight content control, safety-related interventions, and more accessible creative tools. Potential negative impacts are the same as any steering method and include misuse for stylistic or persona-style impersonation and amplification of biased generation when steered toward harmful concepts. The bounded strength parameter $\lambda \in [0, 1]$ provides a graded-application handle that can be paired with usage guidelines and curated concept sets to mitigate misuse. A full discussion is provided in Section O.

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: HyperTransport does not introduce a new generative model or scraped dataset. It produces lightweight intervention parameters for existing pretrained generators, and the misuse surface is bounded by those base models' existing safeguards.

    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 third-party assets used—including CLIP [Radford et al., 2021], DMD2 [Yin et al., 2024], Nitro-1-PixArt (a one-step distillation of Chen et al. [2024a]), and the encoders compared in Section F.1—are released under permissive licenses (either MIT or Apache-2.0) and are credited at first use.

    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: [Yes]

    Justification: The two new assets introduced—the trained hypernetwork and the curated 777-concept training corpus—are fully documented in the paper. Sections C, D and 4.1 document the hypernetwork architecture (including an auxiliary Perceiver IO variant not used in the main results) and its training recipe; Sections B, J and M document the dataset construction pipeline, per-category counts, the full concept list, and the train/test split distribution.

    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 user study (Sections 4.2.2 and H) involved 19 volunteer participants and no compensation was provided. The full task instructions and screenshots of the rating interface are reproduced verbatim in Section H (Figure 8).

    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: [Yes]

    Justification: The user study was a minimal-risk image-preference rating task with voluntary participation, no exposure to potentially harmful content, and no collection of personal data. Under our institution's research policy, image-rating tasks of this nature fall below the threshold requiring formal institutional review; no IRB approval was therefore sought. Participants were informed of the task's nature and voluntary basis prior to participation.

    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: A frontier large language model was used to generate the 32 representative prompt sentences per concept that constitute our training data (Sections B and 4.1). This usage is part of the data-generation pipeline, not the core methodology (the hypernetwork architecture, training objective, and intervention parameterization).

    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.