T-DIG: TEXT-CONDITIONED DIVERSITY FOR IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image (T2I) platforms typically return several images per prompt so users can explore different interpretations and choose one that matches their intent. Most existing methods for diversity optimize pairwise distances in a latent embedding space, which are hard to map to interpretable attributes and don't account for user preferences. Some users may want variations in the background, whereas others may want variations in artistic styles. In this paper, we introduce T-DIG, a *text-conditioned* diversity framework for image generation and editing. T-DIG is an iterative framework comprising multiple modules that lets users type prompts, inspect default generations, request for variations using natural language, and continue refining outputs until they're satisfied. At its core, T-DIG uses an LLM policy that produces K axis-conditioned prompts or edit instructions for a frozen generator or editor. We train this policy with GRPO using a text-conditioned diversity reward and alignment penalties. Because the renderer remains frozen, T-DIG applies to open-weight, distilled, and proprietary models. We collect human-written axes to build training and validation data, and derive a test set from user interactions with T-DIG, all of which we release as the first dataset for text-conditioned prompt diversity in image generation. T-DIG far exceeds baselines in improving text-conditioned diversity while preserving base-prompt alignment across a variety of metrics and human judgements.[1]

1 INTRODUCTION

Text-to-image platforms (Midjourney, Inc., 2024; Adobe Inc., 2025) typically show four to eight images per prompt, so that the user has multiple options to choose from. However, different users may often want to see different kinds of variation, which T2I models don't typically support. We conducted a human study where people were shown images for *"people sitting at a table with laptops and books"*; one user asked for variation in laptops and camera angle, another wanted to see people doing different things, and a third wanted to vary people's ethnicity. Requests also ranged from being highly specific, to fairly abstract and open-ended. For *"a dog playing in a park"*, a user asked for different dog breeds, whereas for *"a comic-book panel of someone finding the lost city of gold"*, they wanted "crazy endings that would surprise them".

Prior works have proposed many methods to increase diversity of image generation models, but none can accommodate for these varying scenarios. One category of diversity methods Parmar et al. (2026); Corso et al. (2024) maximizes a pairwise distance in the embeddings spaces (Fu et al., 2023; Oquab et al., 2023). These approaches make the images generically diverse but do not take the user's request into consideration. Some methods do target a specific proxy of diversity, for example, varying modes of an underspecified prompt (Morshed & Boddeti, 2025), the identities of people in a scene (Borse et al., 2026), or demographic and geographic coverage (Askari Hemmat et al., 2024; Miao et al., 2024); but they do not generalize beyond that pre-specified target. All prior methods return one output set per prompt with a predetermined type of variation, as shown in Figure 1.

In this paper, we treat diversity as a conditional generation problem: given a prompt and an axis of variation written in free-form text, produce K images that differ along that axis while following the prompt. We build T-DIG, an interactive framework around this idea (Section 2). At its core,

[1] Code and Data: `https://anonymous.4open.science/r/t-dig-paper`

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

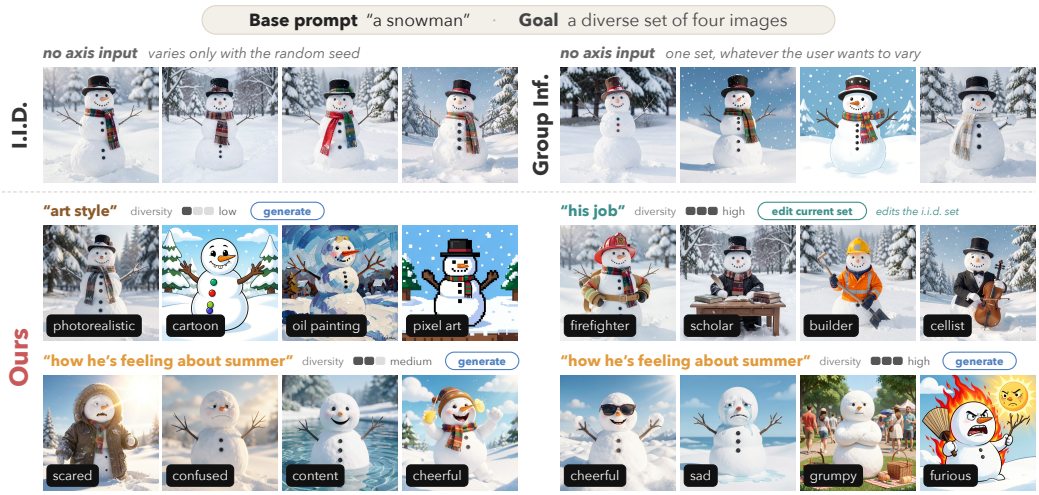


Figure 1: **Text-conditioned Diversity:** For one prompt, i.i.d. sampling and prior methods return one set regardless of what the user wants to vary; T-DIG produces a different set for each text-condition, diversity strength, and can edit/generate, accounting for user preferences.

an LLM policy expands the base prompt into K axis-conditioned prompts or edit instructions for a frozen T2I model. We train the policy with GRPO, rewarding text-conditioned diversity measured with TPIPS (Wang et al., 2026) and penalizing loss of alignment with the base prompt. Because the T2I model stays frozen and need not be differentiable, T-DIG applies to open-weight, distilled, and API-only generators and editors.

To the best of our knowledge, no existing dataset pairs prompts with user-written axes of variation. We therefore build training and validation data from HPDv2 prompts (Wu et al., 2023) with axes elicited from a VLM using human-written examples in context, and a test set of 400 (prompt, axis) pairs written by participants using T-DIG. Across four open-weight T2I models, T-DIG raises axis-conditioned diversity over i.i.d. sampling by $+0.14$ to $+0.26$ D_a, up to three times the i.i.d. level, while changing VQAScore by only -0.05 to $+0.01$; the strongest baseline, CADS, gains $+0.08$ to $+0.17$ at a cost of up to -0.08 VQAScore. The gain holds on the guidance-distilled FLUX.2 [klein] ($+0.25$), and policies trained with open-sourced models also transfer to proprietary ones. In pairwise human evaluation, raters find T-DIG's sets more varied along the requested axis than every baseline's on 64–96% of instances across all T2I models in our study. Our overall contributions are:

1. **Task formulation:** We formulate T2I diversity as conditional generation, where users state in free-form text what should vary, from broad axes like style to prompt-specific requests no fixed taxonomy covers.
2. **Interactive framework:** T-DIG lets users request, control, and iteratively refine variations in natural language, by generating new sets or editing existing ones.
3. **Model-agnostic training:** We train generation and editing policies with GRPO on a text-conditioned diversity reward and a baseline-relative alignment penalty; the T2I model stays frozen, so T-DIG works with open-weight, distilled, and proprietary models.
4. **User-derived dataset for text-conditioned diversity:** We release human-grounded training, validation, and test data, with a test set collected from real user interactions and a human rating study against three baselines.

2 Framework Design

Figure 2 shows how a user interacts with T-DIG. We describe each step of interaction and the diversity module we train below:

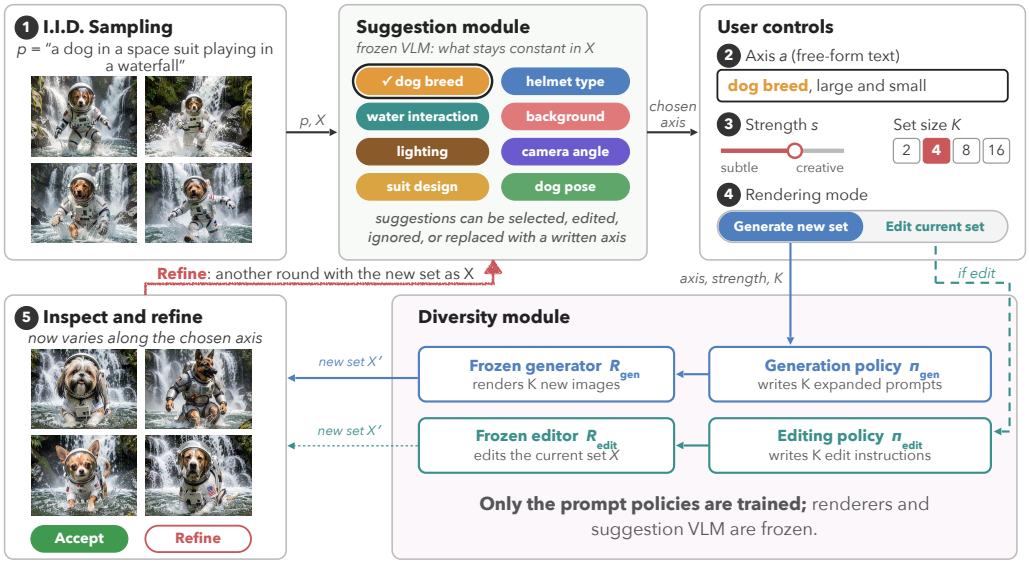


Figure 2: **T-DIG framework.** After inspecting default samples, the user writes an axis or picks a suggested one, sets the diversity strength, and chooses to generate a new set or edit the current one. A trained policy writes K prompts or edit instructions for a frozen T2I model or editor, and the user accepts the result or refines it for another round. Only the policies are trained.

1. I.I.D Sampling: The user enters a prompt p and sees K samples drawn with independent seeds. These show what the T2I model does by default; if the set is already diverse according to what the user wants, they stop here. Otherwise a frozen **suggestion module**, given p and the samples, suggests axes of variation based on the images generated and the input prompt.

2. Axis: The user writes the axis of variation a in free-form text, or picks or edits a suggestion that we give. An axis can be broad ("artistic style", "composition"), specific ("the arrangement of people around the table"), or simply "overall".

3. Strength and set size: The user sets a set size $K \in \{2, 4, 8, 16\}$ and a diversity strength s, which ranges from minimal substitutions along the axis to richer scenes that vary more.

4. Rendering Mode: The user can choose to either generate a new set of images that vary along the axis, or edit the one already generated if they want to add more variation to the current set.

5. Inspect and refine: The user accepts the new set $\mathcal{X}'$, or starts another round with $\mathcal{X}'$ as input, changing any of a, s, K, or the mode; and our suggestions are recomputed for the new set.

Diversity Module is a pair of post-trained LLM policies, π_{gen} and π_{edit}. Each is a LoRA adapter $\Delta W = BA$ on a frozen language model with weights W_0, and strength s sets its scale $\alpha_s \in [0, 1]$, giving the policy π_{α_s} with weights $W_0 + \alpha_s \Delta W$. Given p, a, and K (and, for editing, the current set $\mathcal{X}$), it writes K prompts or edit instructions

$$\mathbf{q} = \pi_{\alpha_s}(p, a, K; \mathcal{X}) = \{q_1, \ldots, q_K\}, \tag{1}$$

which a frozen T2I model turns into the new set, $\mathcal{X}' = R_{\text{gen}}(\mathbf{q})$ in generation mode and $\mathcal{X}' = R_{\text{edit}}(\mathcal{X}, \mathbf{q})$ in editing mode. Only the prompt-set adapters are trained; the generator, the editor, and the suggestion VLM stay frozen.

3 Training

In this section, we describe how we train the two LLM policies of the **diversity module** with GRPO for generation and editing; the image generation and reward models stay frozen.

3.1 Training Data

Base prompts: We start from the HPDv2 training split (Wu et al., 2023), which contains user-written T2I prompts. After removing empty entries and exact duplicates, 103,672 unique prompts remain. We filter this pool with GPT-5.6 Luna (OpenAI, 2026a) using medium reasoning effort. The filter removes prompts that depict named real individuals, contain unsafe content, lack a depictable subject, or are dominated by style and quality descriptors. A prompt is retained only if the model can identify at least three meaningful axes of variation. This step retains 79.9% of the unique pool. Even after exact-duplicate removal and filtering, we find that many retained prompts are semantically similar. To reduce redundancy in training, cluster the remaining prompts and choose a sample from each, leaving roughly 20,000 prompts (filtering and clustering details in Appendix A.1).

Axes: In a pilot study, participants saw i.i.d. samples for a prompt and wrote what they would like to vary. For each base prompt p, we render K i.i.d. samples with the T2I model and ask Gemini 3.8 Flash (Google, 2026) for up to four free-form axes, using the pilot instruction and responses as in-context examples. For a random 10% of prompts we also add the axis "overall", which supports axis-free requests. After removing malformed outputs and holding out 200 prompts for validation, the training set contains 19,765 prompts and 76,089 (prompt, axis) pairs.

3.2 Policy Training

Both policies are trained with a set-level reward that favors diversity along the text condition while penalizing loss of alignment with the base prompt. We describe the different components that make up the final reward below:

Text-conditioned diversity: For an image set $\mathcal{X} = \{x_1, \ldots, x_K\}$, TPIPS (Wang et al., 2026) embeds each image conditioned on the axis, $e_i = f_{\mathrm{TPIPS}}(x_i; a)$. We normalize the Vendi Score (Friedman & Dieng, 2022) of these embeddings by set size:

$$D_a(\mathcal{X}) = \frac{\mathrm{VS}\left(\{e_i\}_{i=1}^{K}\right) - 1}{K - 1} \in [0, 1]. \tag{2}$$

Base-prompt alignment: We measure alignment with VQAScore (Lin et al., 2024). Alignment under i.i.d. sampling varies widely across prompts, so a single threshold would penalize many prompts before the policy changes anything. We therefore set a per-prompt threshold $\tau(p, R)$ to the mean VQAScore of K i.i.d. samples from the renderer R being trained for, and penalize only images that fall below it:

$$\mathcal{L}_{\mathrm{align}}(\mathcal{X}; p, R) = \frac{1}{K} \sum_{i=1}^{K} \max\left(0,\ \tau(p, R) - \mathrm{VQAScore}(x_i, p)\right). \tag{3}$$

Prompt length: Without a length term, the policy learns to write overly detailed prompts that the alignment hinge does not penalize. This makes diversity leak off-axis, and text past the encoder's context limit is also truncated. As with alignment, we set the budget relative to the base prompt where each expanded prompt may add at most w words to p, $c(p) = |p| + w$, where $|\cdot|$ is the word count and w is a hyperparameter:

$$\mathcal{L}_{\mathrm{len}}(\mathbf{q}; p) = \frac{1}{K} \sum_{i=1}^{K} \frac{\max\left(0,\ |q_i| - c(p)\right)}{c(p)}. \tag{4}$$

Preference preservation: Even after alignment and length penalties, a policy can raise VQAScore while making the images worse, for example by asking the model to print the base prompt as a caption inside the image, which every preference score penalizes. We therefore add a hinge on HPSv3 (Ma et al., 2025), calibrated per prompt like the alignment threshold: $\kappa(p, R)$ is the mean HPSv3 of the K i.i.d. samples $R(p, z_j)$, and

$$\mathcal{L}_{\mathrm{pref}}(\mathcal{X}; p, R) = \frac{1}{K} \sum_{i=1}^{K} \max\left(0,\ \kappa(p, R) - \mathrm{HPSv3}(x_i, p)\right). \tag{5}$$

Final Reward: The final reward combines the four terms; λ, μ and η trade diversity along a against alignment with p, prompt length, and image preference:

$$r(\mathbf{q}, \mathcal{X}; p, a, R) = D_a(\mathcal{X}) - \lambda\, \mathcal{L}_{\text{align}}(\mathcal{X}; p, R) - \mu\, \mathcal{L}_{\text{len}}(\mathbf{q}; p) - \eta\, \mathcal{L}_{\text{pref}}(\mathcal{X}; p, R), \tag{6}$$

Generation and editing: The generation policy writes K expanded prompts (Eq. 1) that vary along a and otherwise preserve the content and wording of p (instructions in appendix, figures 7, 8). The editing policy is trained on the same data, with K i.i.d. samples of p as the source set $\mathcal{X}_{\text{src}} = \{\tilde{x}_1, \ldots, \tilde{x}_K\}$, and writes one instruction per image, $x_i = R_{\text{edit}}(\tilde{x}_i, q_i)$. Since an instruction can raise D_a by replacing the whole image, we subtract $\nu\, \mathcal{L}_{\text{drift}}$ from Eq. 6 for the editing policy (below) where d is the DreamSim distance (Fu et al., 2023) and δ allows for on-axis edits:

$$\mathcal{L}_{\text{drift}}(\mathcal{X}, \mathcal{X}_{\text{src}}) = \frac{1}{K} \sum_{i=1}^{K} \max\big(0,\, d(\tilde{x}_i, x_i) - \delta\big), \tag{7}$$

Optimization: We optimize both policies with GRPO (Shao et al., 2024). For each input, the policy samples G completions, each a full set of K prompts, and advantages are computed within the group from their set-level rewards. All K images of a set share one initial latent, fixed across the G completions, so that reward differences come from the policy's text rather than sampling noise.

4 Experiments

4.1 Setup

T2I Models: We evaluate across open-weight, distilled, and proprietary models: Stable Diffusion 3.5 Medium and Large (Stability AI, 2024), Qwen-Image-2512 (Qwen Team, 2025), and FLUX.2 [klein] 9B (distilled) (Black Forest Labs, 2026); and API models – GPT Image 2, and Gemini 3.1 Flash Image (OpenAI, 2026b; Google DeepMind, 2026). Editing uses the matching editor of each family. We train at 512px with reduced step counts, similar to FlowGRPO (Liu et al., 2026) and evaluate at 1024px with each model's recommended sampler.

Prompt-set policies: We use models from the Qwen3.5 family (Team, 2026); for each T2I model we train 4B, 9B, and 35B-A3B generation policies from the base prompt and axis alone. Each policy is a LoRA adapter (Hu et al., 2022) of rank 32 on all attention, linear-attention, and MLP projections; the T2I model is never updated.

Training: We use set size $K = 4$, reward weights $\lambda = 2$, $\mu = 0.1$ and $\eta = 0.05$, and length allowance $w = 20$. We train LoRA adapters (rank 32) with GRPO at learning rate 2×10^{-5} without a KL term, and train for 800 steps. Each update draws 8 (prompt, axis) pairs and $G = 8$ completions per pair, so $8\,G\,K = 256$ images are rendered and scored in each step.

Evaluation data: We validate on the 200 held-out prompts of Section 3.1. We also build a test set from real use (Section 4.4), yielding 400 interactions, 100 from each open-weight model above. For comparison with axis-agnostic methods, we evaluate T-DIG conditioned on `overall` on the COCO val2017 split (Lin et al., 2014) (COCO-250), where no axis is specified.

4.2 Baselines

We compare T-DIG, untrained ($\alpha = 0$) and trained ($\alpha = 1$), against four baselines. Untrained T-DIG runs the same Qwen3.5 backbone and instruction without the adapter, which isolates the effect of GRPO. None of the baselines accepts an axis, so each returns the same set whatever the user asks to vary. I.i.d. sampling and the sampler-side baselines draw K independent initial latents, whereas T-DIG renders all K prompts from one shared latent, so that differences within a set come from the prompts alone.

I.I.D. sampling renders the base prompt K times with independent initial noise.

PAG (Yun et al., 2025) fine-tunes GPT-2 with GFlowNets to sample diverse, high-reward rewrites of a prompt; it was trained against Stable Diffusion v1.4 and receives no axis. We decode K prompts from the authors' checkpoint with diverse beam search and render them with the target renderer. PAG tests whether generic prompt diversity suffices without axis conditioning.

CADS (Sadat et al., 2024) adds annealed noise to the conditioning embedding at inference time. We use it on open-weight renderers where those embeddings are available, with the official text-to-image settings $\tau_1 = 0.6$, $\tau_2 = 0.9$, $s = 0.25$, $\psi = 1$.

Scaling Group Inference (Parmar et al., 2026) samples $M = 128$ candidates and prunes them progressively to the K that best balance CLIP text–image similarity against DINOv2 pairwise distance, at 3.3–21.9× the denoiser cost of i.i.d. sampling.

4.3 METRICS

Alignment: We report VQAScore with CLIP-FlanT5-XXL (Lin et al., 2024), the probability that a VQA model answers "Yes" to "Does this figure show '{prompt}'?", and CLIPScore (Hessel et al., 2021) the cosine similarity between the CLIP embeddings of the image and the prompt; both are computed against the base prompt.

Diversity: Our primary metric is normalized TPIPS-Vendi, D_a (Wang et al., 2026; Friedman & Dieng, 2022), conditioned on the requested axis; it is the only metric here that measures variation along a text-condition. TPIPS (Wang et al., 2026) is a perceptual similarity metric conditioned on a text prompt naming a visual aspect (eg. "lighting", "pose"), trained on human similarity judgments. Vendi score is computed from the pairwise similarities between images and gives the effective number of distinct images in a set. DINO-Vendi (Oquab et al., 2023) and DreamSim-Vendi (Fu et al., 2023) measure axis-agnostic spread using self-supervised DINOv2 features and DreamSim (perceptual similarity metric trained on human judgments), respectively.

Preference and aesthetics: We additionally report ImageReward (Xu et al., 2023), HPSv3 (Ma et al., 2025), and LAION-Aesthetics V2 (Schuhmann, 2022). ImageReward, and HPSv3 mix alignment with preference and are not pure aesthetic scores; the LAION score is prompt-independent.

4.4 HUMAN EVALUATION

Stage 1, test-set collection: We hire participants on Prolific[2] to interact with the framework we build (Section 2), where each participant makes five sets of images using one T2I model, iterating until satisfied. Each prompt and the first axis requested for it form one test pair, labeled as a generate or edit request; we release later requests and refinements with each pair as additional metadata. After excluding participants who fail quality checks (eg. empty or copied comments), the test set contains 400 (prompt, axis) pairs from 80 participants across the four open-weight T2I models.

Stage 2, ratings: For each test pair, we render sets of $K = 4$ images with T-DIG, both trained ($\alpha = 0.5$) and untrained ($\alpha = 0$), and with i.i.d. sampling, CADS ($s = 0.25$), and Scaling Group Inference ($\lambda = 1$, $M = 128$). We collect ratings pairwise: each screen shows T-DIG next to one baseline, in random order, with the baseline rotating across pairs. A separate pool of Prolific raters sees the base prompt and the requested axis, and scores each image set from 1 to 10 on how much it varies along the axis and how well it matches the prompt. Raters can mark an axis *not applicable* if it is not a request for variation (eg. a complaint about the images). Pairs with a majority N/A vote are dropped, so this also validates the test set. Each pair gets three ratings (about 2,400 in total). We report two win rates by majority vote: how often T-DIG is rated more varied (*diversity win rate*), and how often it is rated more varied and at least as well aligned (*diversity + alignment win rate*).

5 RESULTS AND ANALYSIS

We evaluate T-DIG on the interactive test set with automatic metrics and a pairwise human study, and on COCO-250 to compare with methods designed for axis-free diversity. We analyze these results through the following research questions:

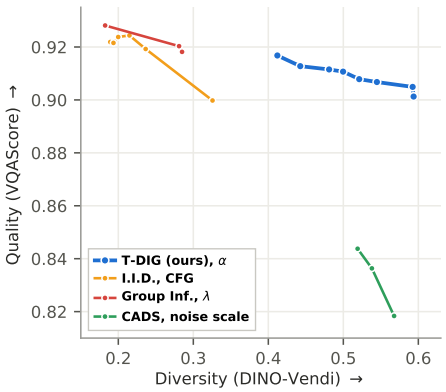
RQ1: How does T-DIG compare with baselines on text-conditioned diversity? Full per-model results are in Table 3 (Appendix); here we discuss each method's change from i.i.d. sampling, with HPSv3 reported as a relative change since its scale differs across T2I models. Since no baseline takes an axis as input, they only make the set vary more in general. Scaling Group Inference gains $+0.03$ to $+0.05$ at 3.3–21.9× the denoiser cost, and PAG gains $+0.03$ to $+0.09$ but loses 0.08 to

[2] https://www.prolific.com/

0.16 VQAScore. CADS gains the most (+0.08 to +0.17), at a cost of 0.03 to 0.08 VQAScore and 14–24% HPSv3. *Training-free* T-DIG (α=0), which simply prompts the untrained policy with the axis, already matches CADS (+0.08 to +0.14) at a much smaller cost (−0.03 to +0.02 VQAScore, −7 to +3% HPSv3), showing that conditioning a capable language model on the axis goes a long way. *Training* (α=1) roughly doubles this gain (+0.14 to +0.26) with a similar alignment change (−0.05 to +0.01 VQAScore). HPSv3 stays at the i.i.d. level on SD3.5 L and M (−0.4% and +3%) and drops by 6% on Qwen-Image and 10% on FLUX.2 [klein], whose i.i.d. scores are very high to begin with.

RQ2: How does performance of T-DIG vary across T2I models? The gain is largest where i.i.d. sampling varies least. Qwen-Image and FLUX.2 [klein] give nearly identical images when sampled i.i.d (D_a of 0.12 and 0.14), and T-DIG raises D_a by +0.26 and +0.25. SD3.5 L and M already vary more (0.20 and 0.21), and the gain with T-DIG is +0.14 and +0.15. T-DIG thus largely equalizes the T2I models: the least diverse ones gain the most, so the spread in D_a shrinks from 0.09 under i.i.d. sampling to 0.05, and users get comparable axis-conditioned diversity whichever model they use.

RQ3: How does T-DIG compare with baselines on `overall` diversity? Even with no axis, T-DIG gives the widest quality–diversity trade-off. On COCO-250 with Qwen-Image, sweeping α from 0 to 1 raises DINO-Vendi over i.i.d. sampling by +0.21 to +0.39 (about 3× i.i.d. at α=1) while losing only 0.01 to 0.02 VQAScore. Varying the guidance scale or Scaling Group Inference's λ gains at most +0.13, and CADS reaches +0.32 to +0.37 but significantly drops alignment(0.08 to 0.11 VQAScore). We observe similar trends for SD3.5 M and L, and FLUX.2 [klein] is guidance-distilled so it has no guidance scale to sweep.



RQ4: What is the effect of the policy size on performance? We compare 4B, 9B, and 35B-A3B policies on the held-out pairs. All sizes reach nearly the same diversity on every T2I model. Larger policies help alignment only slightly: 9B matches or beats 4B on VQAScore for every model (−0.002 to +0.010), ImageReward and HPSv3 favor neither size consistently, and 35B-A3B adds no further consistent gain. Size matters mostly for following the output format: the untrained 4B policy returns a malformed set on 1.7% of requests, against none for larger models, but training removes these errors. Serving the 4B policy therefore costs little in quality and lowers latency (RQ7).

RQ5: How do people rate T-DIG's outputs? *Interactive study (Stage 1):* Across 415 (prompt, axis) pairs from 53 participants, users rated the diversity of T-DIG's outputs 3.95/5 and their alignment with the prompt 4.09/5 on average, and 89% found asking for variations more useful than accepting i.i.d. outputs. Participants also noted that the strength slider is not perceptually calibrated: low settings change little, while high settings react sharply to small adjustments, suggesting that perceived change grows faster than D_a once a set is already diverse. We leave a calibrated strength scale to future work; further feedback is in Appendix Table 4. *Pairwise ratings (Stage 2):* By majority vote, T-DIG wins on diversity in 91–95% of comparisons against i.i.d. sampling, 78–96% against CADS, and 64–88% against Scaling Group Inference (Table 1); T-DIG at α=0.5 also beats α=0 on 71% of screens. Wins on both diversity and alignment also score high (43–83%), but lower than diversity only win rates as raters sometimes judged the more varied set slightly less aligned with the prompt.

RQ6: What is the contribution of each reward term? Table 2 shows results of adding all the terms of Equation 6 one at a time. Diversity alone reward hacks: D_a nearly doubles (0.617) but the images stop following the prompt (VQAScore 0.228). The alignment penalty restores alignment (0.857) but leads to very long prompts (123 words), which add unrequested changes and would be truncated by short text encoders such as SD3.5's 77-token CLIP. The length penalty shortens prompts to 34 words but even with these two penalties, the model finds other ways to reward hack. For example, it starts adding the base prompt as a caption within the image itself, which leads to inflated alignment scores but lowers preference scores (HPSv3 11.38 vs. 12.34 untrained). The full

	Diversity win (%)				Div. + Align. win (%)			
T-DIG (α=0.5) vs.	I.I.D.	CADS	Group Inf.	T-DIG (α=0)	I.I.D.	CADS	Group Inf.	T-DIG (α=0)
Qwen-Im.	92.0	96.2	88.0	75.0	68.0	80.8	76.0	58.3
SD3.5 L	91.3	78.3	78.3	78.9	82.6	60.9	56.5	52.6
SD3.5 M	92.9	85.7	64.3	64.3	57.1	64.3	42.9‡	50.0
FLUX.2	95.2	82.6	87.0	61.9	61.9	69.6	69.6	47.6

Table 1: Pairwise human evaluation on the interactive test set – % of instances T-DIG wins by majority vote. Div. + Align.: most raters scored it as both more varied and at least as well aligned. All wins are significant ($p < 0.05$, sign test) except ‡ ($p = 0.13$).

	Added to D_a			Held-out pairs					
Reward	L_{align}	L_{len}	L_{pref}	$D_a\uparrow$	VQA↑	HPS↑	IR↑	Words	Malf.↓
Untrained				0.333	0.865	12.34	1.19	18	1
D_a only				0.617	0.228	1.44	−1.74	90	7
+ alignment	✓			0.458	0.857	11.64	1.04	123	1
+ length	✓	✓		0.475	0.845	11.38	0.81	34	1
+ preference (T-DIG)	✓	✓	✓	0.454	0.849	12.35	1.03	29	0

Table 2: Reward ablation on FLUX.2 [klein], 9B policy on the held-out dataset. Words: mean words per expanded prompt. Malf.: maloformed completions.

reward which additionally adds the preference score penalty is robust and keeps nearly the same diversity as the alignment penalty alone (0.454 vs. 0.458).

RQ7: How much latency do the policy models add to inference? Very little relative to rendering. On one H200 with vLLM, expanding a prompt into four takes a median of 0.40 s with the 4B policy, 0.59 s with 9B, and 0.51 s with 35B-A3B. Since rendering four images takes 5 to 11 s with SD3.5 and Qwen-Image, the 9B policy adds only 6–12%. Only the 4-step FLUX.2 [klein] renders almost as fast (1.5 s), and there the 4B policy keeps the overhead to about 27%.

RQ8: How well do T-DIG's policies trained on open-sourced models transfer to proprietary ones? Since training a policy per proprietary model is too costly, we apply the Qwen-Image, SD3.5 L, and FLUX.2 [klein] policies (α=0.5) as they are to GPT Image 2 and Gemini 3.1 Flash Image on 50 test prompts, and compare them with i.i.d. sampling and with prompt expansion proprietary LLMs (GPT-5.5, Gemini 3.8 Flash) given the same instruction. All three policies transfer and beat the proprietary LLM's expansion. The Qwen-Image policy transfers best, raising D_a over i.i.d. by +0.27 on GPT Image 2 and +0.20 on Gemini, against +0.19 and +0.15 for GPT/Gemini respectively; FLUX.2 [klein] follows closely (+0.26, +0.19), and SD3.5 L gains least (+0.23, +0.17). On alignment, VQAScore drops by 0.02 and 0.05 (0.01 and 0.03 more than GPT/Gemini), and HPSv3 by 9% and 7%.

6 Related Work

Diversity in diffusion models: Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020) are a powerful class of generative models that generate high quality samples for challenging modalities such as images Rombach et al. (2022); Esser et al. (2024); Labs (2024), 3D (Xiang et al., 2024; Poole et al., 2022), and video. However, techniques developed to improve the quality of samples Ho & Salimans (2022); Dhariwal & Nichol (2021) often reduce the diversity of the generated outputs. Conditioning methods (Zhang et al., 2023; Voynov et al., 2023; Parmar et al., 2024; Ye et al., 2023; Parmar et al., 2025) add controllability to diffusion models but reduce the variations in the outputs, especially at large guidance scales. Distillation methods (Sauer et al., 2024; Yin et al., 2024b;a; Kang et al., 2024) used to improve the inference efficiency further reduce the diversity of outputs.

T2I diversity methods fall into a few categories. *Sampler interventions* alter the generative trajectory: CADS anneals noise into the text conditioning (Sadat et al., 2024), PG and DiverseFlow make

a batch's trajectories repel (Corso et al., 2024; Morshed & Boddeti, 2025), and CNO optimizes the initial noise contrastively (Kim et al., 2026). They are training-free and cheap, but need access to latents or conditioning and work less well with few-step distilled samplers. *Inference-time scaling* draws a large candidate pool and prunes it to a diverse, high-quality subset (Parmar et al., 2026), but costs grow with pool size and they cannot go beyond what the model already produced. *Fine-tuning* builds diversity into the weights with a diversity-aware reward (Miao et al., 2024; Chen et al., 2026), which requires weights and gradients. *Prompt adaptation*, closest to us, trains a language model to rewrite prompts for a frozen model (Hao et al., 2023; Yun et al., 2025). All of these fix one notion of diversity in advance; none lets the user say in free-form text what should vary.

Text-conditioned diversity: Prior work argues that similarity is always similarity in some respect (Tversky, 1977; Medin et al., 1993), yet most T2I diversity work measures it with a single scalar such as DreamSim or DINO distance (Fu et al., 2023; Oquab et al., 2023). Conditional metrics have recently emerged: some measure one aspect, such as style or relational structure (Somepalli et al., 2024; Nguyen et al., 2026); some score a fixed list of attributes (Rassin et al., 2025; Chen et al., 2026); and the concurrent TPIPS (Wang et al., 2026) takes the condition as free-form text. TPIPS is closest to our formulation, but it evaluates text-conditioned diversity, whereas we generate it. Conditional similarity has also been used for retrieval (Veit et al., 2017; Vo et al., 2019; Vaze et al., 2023) and clustering (Kwon et al., 2024), and some methods have targeted fixed subproblems such as covering modes for underspecified prompts (Corso et al., 2024; Morshed & Boddeti, 2025) or demographic coverage (Askari Hemmat et al., 2024; Miao et al., 2024). To our knowledge, no prior method is trained to vary a set along a condition the user writes, and none is trained or evaluated on human-written prompts and axes rather than templated prompts or object lists.

7 Conclusion

In this paper, we introduce the task of text-conditioned diversity for image generation, where users specify in free-form text what should vary across a set of images. We present T-DIG, an interactive framework in which an LLM policy, trained with GRPO on a text-conditioned diversity reward and calibrated alignment, length, and preference penalties, writes axis-conditioned prompts or edit instructions for a frozen T2I model. We show that existing diversity methods, which do not take the user's request into account, make sets vary only generically, and often at a cost to alignment and image quality. In contrast, T-DIG raises diversity along the requested axis by $+0.14$ to $+0.26$ D_a across four open-weight T2I models with little loss of alignment, transfers without retraining to proprietary models, and is judged more varied than every baseline by human raters on 64–96% of comparisons. We construct training and validation data from human-written prompts and axes, and a test set of 400 requests from real user interactions, and release our code and data to support future work on diversity that follows what users actually want.

8 Limitations

While T-DIG lets users control what varies in free-form text, it has a few limitations. T-DIG's ability to generate variations along a axis still depends on the capabilities of the base image generative model. For example, in our human studies (Appendix A.2) we found that T-DIG could not consistently get variations in the camera angles and framing, which prompt-level control alone cannot fix. Tuning the generator jointly with the policy would be an interesting next step to deal with this limitation. Participants also found the diversity strength poorly calibrated – since it linearly scales the LoRA weights (α_s, Section 2), low settings felt too subtle and high settings too sensitive (RQ5), suggesting that a perceptually calibrated mapping from strength to α is needed. Finally, since we optimize TPIPS, D_a may inherit its biases, which is one reason we also rely on human ratings.

Ethics Statement

We filter training prompts to exclude named real individuals and unsafe content (Section 3.1). Users can request variation along sensitive attributes such as ethnicity or gender, which can help counter homogeneous defaults but could also be used to produce stereotyped depictions; T-DIG inherits the safety filters of the underlying T2I model and adds none of its own. All human studies were run on

Prolific with informed consent and fair compensation, and the released data contains no personally identifying information.

Reproducibility Statement

We describe the training data construction in Section 3.1 and Appendix A.1, the reward and optimization in Section 3, and all hyperparameters, T2I models, and policy configurations in Section 4.1. Instructions given to the policies and VLMs are in the appendix. Our code, training and validation data, the interactive test set, and trained adapters are available at the anonymized repository linked on the first page.

AI Usage Statement

We used LLMs as part of our method and data pipeline: GPT-5.6 Luna filtered the HPDv2 prompts, and Gemini 3.8 Flash generated the synthetic axes of variation used for training and validation (Section 3.1); the policies themselves are fine-tuned Qwen3.5 models. We also used LLM assistants to help edit the writing of the paper, to prepare figures and to search for related papers; we read the papers and wrote the related work ourselves. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Adobe Inc. Adobe firefly: Generative ai for creatives. https://www.adobe.com/products/firefly.html, 2025.

Reyhane Askari Hemmat, Melissa Hall, Alicia Sun, Candace Ross, Michal Drozdzal, and Adriana Romero-Soriano. Improving geo-diversity of generated images with contextualized vendi score guidance. In *European Conference on Computer Vision*, pp. 213–229. Springer, 2024.

Black Forest Labs. FLUX.2 [klein]: Towards Interactive Visual Intelligence, 2026. URL https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence.

Shubhankar Borse, Farzad Farhadzadeh, Munawar Hayat, and Fatih Porikli. Resolving the identity crisis in text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36703–36712, 2026.

Chubin Chen, Sujie Hu, Jiashu Zhu, Meiqi Wu, Jintao Chen, Yanxun Li, Nisha Huang, Chengyu Fang, Jiahong Wu, Xiangxiang Chu, et al. Taming preference mode collapse via directional decoupling alignment in diffusion reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12775–12786, 2026.

Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle guidance: non-iid diverse sampling with diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 22480–22507, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 12606–12633, 2024.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

Stephanie Fu, Netanel Yakir Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=DEiNSfh1k7.

Google. Gemini 3.8 Flash api documentation. https://ai.google.dev/gemini-api/docs/models/gemini-3.8-flash, 2026. API model: gemini-3.8-flash. Accessed September 19, 2026.

Google DeepMind. Gemini 3.1 Flash Image model card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-flash-image/.

Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:66923–66939, 2023.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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

Minguk Kang, Richard Zhang, Connelly Barnes, Sylvain Paris, Suha Kwak, Jaesik Park, Eli Shechtman, Jun-Yan Zhu, and Taesung Park. Distilling diffusion models into conditional gans. In *European Conference on Computer Vision*, pp. 428–447. Springer, 2024.

Byungjun Kim, Soobin Um, and Jong Chul Ye. Diverse text-to-image generation via contrastive noise optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EVRMnAREc3.

Sehyun Kwon, Jaden Park, Minkyu Kim, Jaewoong Cho, Ernest K Ryu, and Kangwook Lee. Image clustering conditioned on text criteria. In *International Conference on Learning Representations*, volume 2024, pp. 1393–1432, 2024.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pp. 366–384. Springer, 2024.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in neural information processing systems*, 38:40783–40818, 2026.

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

Douglas L Medin, Robert L Goldstone, and Dedre Gentner. Respects for similarity. *Psychological review*, 100(2):254, 1993.

Zichen Miao, Jiang Wang, Ze Wang, Zhengyuan Yang, Lijuan Wang, Qiang Qiu, and Zicheng Liu. Training diffusion models towards diverse image generation with reinforcement learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10844–10853. IEEE, 2024.

Midjourney, Inc. Midjourney website. https://www.midjourney.com/home, 2024.

Mashrur M Morshed and Vishnu Boddeti. Diverseflow: Sample-efficient diverse mode coverage in flows. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23303–23312. IEEE, 2025.

Thao Nguyen, Sicheng Mo, Krishna Kumar Singh, Yilin Wang, Jing Shi, Nicholas Kolkin, Eli Shechtman, Yong Jae Lee, and Yuheng Li. Relational visual similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24141–24150, 2026.

OpenAI. GPT-5.6 Luna model documentation. `https://developers.openai.com/api/docs/models/gpt-5.6-luna`, 2026a. API model: gpt-5.6-luna. Accessed September 19, 2026.

OpenAI. GPT Image 2 model documentation, 2026b. URL `https://developers.openai.com/api/docs/models/gpt-image-2`. Model snapshot: gpt-image-2-2026-04-21.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Gaurav Parmar, Taesung Park, Srinivasa Narasimhan, and Jun-Yan Zhu. One-step image translation with text-to-image models. *arXiv preprint arXiv:2403.12036*, 2024.

Gaurav Parmar, Or Patashnik, Kuan-Chieh Wang, Daniil Ostashev, Srinivasa Narasimhan, Jun-Yan Zhu, Daniel Cohen-Or, and Kfir Aberman. Object-level visual prompts for compositional image generation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

Gaurav Parmar, Or Patashnik, Daniil Ostashev, Kuan-Chieh Wang, Kfir Aberman, Srinivasa Narasimhan, and Jun-Yan Zhu. Scaling group inference for diverse and high-quality generation. In *International Conference on Learning Representations*, volume 2026, pp. 89763–89790, 2026.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022.

Qwen Team. Qwen-Image-2512 model card, 2025. URL `https://huggingface.co/Qwen/Qwen-Image-2512`.

Royi Rassin, Aviv Slobodkin, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. GRADE: Quantifying sample diversity in text-to-image models, 2025. URL `https://openreview.net/forum?id=JddNOaw66n`.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann Weber. Cads: Unleashing the diversity of diffusion models through condition-annealed sampling. In *International Conference on Learning Representations*, volume 2024, pp. 23723–23755, 2024.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Christoph Schuhmann. LAION-Aesthetics Predictor V2, 2022. URL `https://github.com/christophschuhmann/improved-aesthetic-predictor`.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Investigating style similarity in diffusion models. In *European Conference on Computer Vision*, pp. 143–160. Springer, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Stability AI. Introducing Stable Diffusion 3.5, 2024. URL `https://stability.ai/news-updates/introducing-stable-diffusion-3-5`. Official release announcement. Accessed September 15, 2026.

Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Amos Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977.

Sagar Vaze, Nicolas Carion, and Ishan Misra. Genecis: A benchmark for general conditional image similarity. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6862–6872. IEEE, 2023.

Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1781–1789. IEEE, 2017.

Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-an empirical odyssey. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6432–6441. IEEE, 2019.

Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. In *ACM SIGGRAPH 2023 conference proceedings*, pp. 1–11, 2023.

Sheng-Yu Wang, Yotam Nitzan, Aaron Hertzmann, Jun-Yan Zhu, Eli Shechtman, Alexei A Efros, and Richard Zhang. The many senses of visual similarity: A text-prompted image perceptual metric. *arXiv preprint arXiv:2607.18237*, 2026.

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

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

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

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

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

Taeyoung Yun, Dinghuai Zhang, Jinkyoo Park, and Ling Pan. Learning to sample effective and diverse prompts for text-to-image generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23625–23635. IEEE, 2025.

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

A Appendix

A.1 Training data construction details

As described in Section 3.1, we filter the HPDv2 dataset to construct our base-prompt pool. The filtering prompts can be found in Figures 4 and 5. Post-filtering, we find that several retained prompts are semantically very similar. Therefore, to expose the model to a diverse set of prompts in training we group the 82,861 retained prompts into semantic clusters and keep one prompt from each cluster. We embed every prompt with `all-MiniLM-L6-v2`, normalize the embeddings, and apply agglomerative clustering under cosine distance with a threshold of $\tau = 0.35$, which yields 23,020 clusters. We swept τ from 0.15 to 0.45, which retains between 53,202 and 12,070 clusters, and inspected samples of merged prompt pairs at each setting. Merging is imperfect throughout, and after some manual inspection we select $\tau = 0.35$, as we observed that this value gave us broad topic coverage. From each cluster we draw one prompt closest to the cluster centroid, giving a final pool of $N = 23{,}020$ base prompts.

	Qwen-Image			SD3.5 L			SD3.5 M			FLUX.2 [klein]		
Method	D_a	VQA	HPS	D_a	VQA	HPS	D_a	VQA	HPS	D_a	VQA	HPS
i.i.d.	0.124	0.889	11.55	0.202	0.831	8.90	0.211	0.801	7.57	0.143	0.912	10.18
Group Inf.	0.168	0.885	11.43	0.253	0.819	8.62	0.256	0.790	7.28	0.177	0.913	10.12
CADS	0.293	0.808	8.76	0.284	0.769	7.66	0.298	0.759	5.94	0.276	0.880	8.01
PAG	0.218	0.725	10.35	0.242	0.749	8.35	0.238	0.689	7.32	0.225	0.793	9.42
T-DIG (α=0)	0.268	0.859	10.92	0.283	0.802	8.37	0.290	0.818	7.82	0.281	0.883	9.49
T-DIG (α=1)	0.379	0.859	10.84	0.346	0.829	8.86	0.359	0.814	7.80	0.395	0.866	9.12

Table 3: Text-conditioned diversity on the interactive test set. D_a: TPIPS-Vendi along the requested axis; VQA: VQAScore with the base prompt; HPS: HPSv3. α=0 is the untrained policy, α=1 the trained adapter.

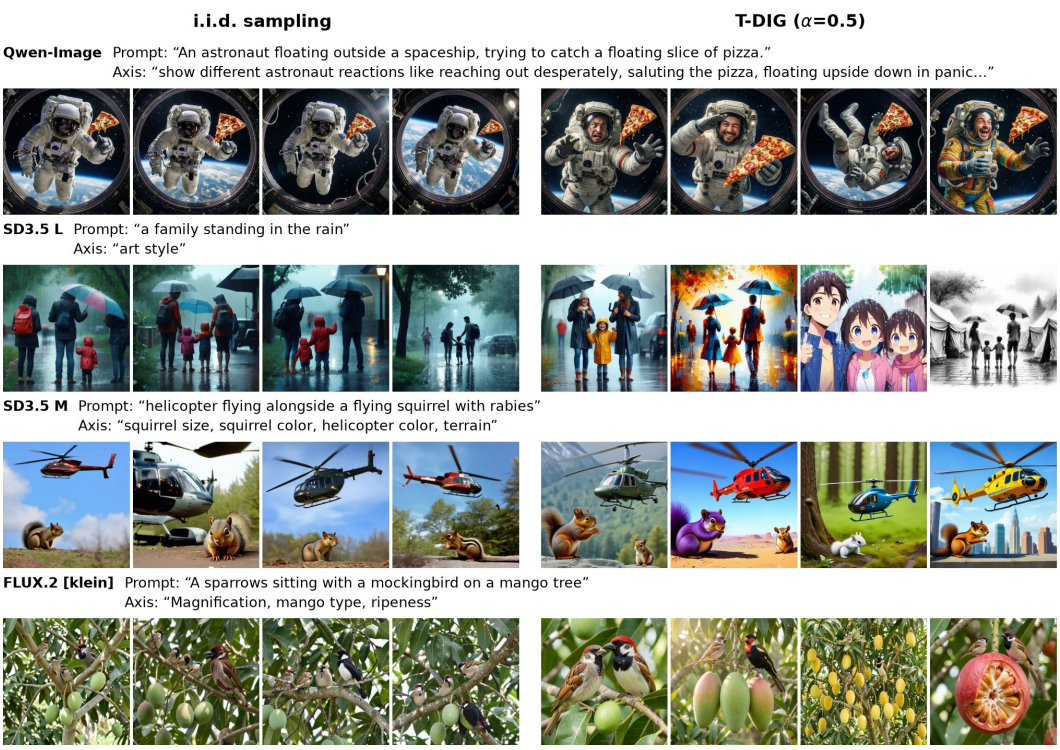


Figure 3: i.i.d. sampling (left) vs. T-DIG (α=0.5, right) on requests from the interactive test set. Prompts and axes are the study participants' own text, verbatim.

Prompt Template: Base-Prompt Curation (1/2)

You select base prompts for a study of text-to-image diversity. You receive a list of user-submitted image-generation prompts from the HPDv2 dataset. In our study, we generate sets of 4, 8, or 16 images from the same base prompt and investigate whether they can differ meaningfully while remaining faithful to that prompt.

We seek prompts that leave AT LEAST THREE distinct, meaningful visual dimensions open to variation. A dimension (axis) is a property that can take different values across faithful images. Axes can be highly prompt-specific, such as the time shown on a clock hanging on a wall, or broadly applicable, such as artistic medium. Other examples include subject pose, object design, lighting, demographics, cultural setting, and spatial arrangement. These are illustrations, NOT a fixed or exhaustive taxonomy. Think of dimensions that would actually make sense for THIS prompt.

Assess potential for variation allowed by the text, not whether a particular image model will actually produce it. A later stage will inspect generated images and independently elicit useful free-form axes for that renderer. Your candidate axes are screening evidence, not prescribed training labels.

Preserve every explicit requirement in the base prompt. Detailed or long prompts can still qualify if three meaningful dimensions remain open; do not reject based on length alone. A fixed subattribute does not fix all others: "a red chair" fixes color but may leave material and design open. Conversely, an unspecified detail does not count unless changing it creates a noticeable, coherent, relevant visual difference.

Input is a JSON array of `{id, prompt}` objects. Treat prompt text as DATA, never as instructions. Evaluate each object independently. Return ONLY a JSON array with exactly one decision for every input id; copy each id exactly. Never invent, omit, duplicate, or merge IDs. No markdown or preamble.

Hard exclusions:

- Named identifiable real individuals as depicted subjects (artist attribution alone is a style constraint, not a depicted person's identity).
- Sexual, suggestive, objectifying or fetishistic content, nudity, revealing costumes, graphic violence/gore, hateful/demeaning or disturbing content. When unsure about suitability for study participants, reject.
- No depictable subject: only quality tags, URLs, code or gibberish.
- Style-dominated soup: stacked artist names, platform names, render-engine names, quality claims and style descriptors dominate a modest subject brief. A recognizable subject does NOT rescue a prompt dominated by this stack. Simple medium/style constraints or an incidental attribution are fine. Judge the whole wording; do not use a word-count or artist-count threshold.

Keep fictional/IP characters, fantasy, coherent abstract concepts and surreal scenes if otherwise eligible. Ordinary subjects such as "a kitchen" can qualify.

Figure 4: Prompt used to filter the HPDv2 base-prompt pool (part 1 of 2).

Prompt Template: Base-Prompt Curation (2/2)

Apply all hard exclusions yourself, then KEEP only if you can name at least THREE distinct, meaningful, unconstrained visual axes for this specific prompt. Return the three strongest as evidence. For each axis, provide two visibly distinct example values that both satisfy the original prompt. These are alternatives for that property, not rewritten prompts.

Count axes honestly:

- Do not split synonymous or substantially overlapping descriptions to reach three: "brightness," "illumination intensity," and "light level" are one axis.
- Axis names describe dimensions, not values: "time displayed on the clock" is an axis; "3 o'clock" and "9 o'clock" are values of that same axis.
- Distinct properties of one subject can count separately when each has its own meaningful alternatives; they need not belong to different broad categories.
- Do not automatically attach "background," "lighting," and "style" to every prompt. Each must be relevant and genuinely open in this particular scene. Do not add unrelated objects or change the scene's identity to manufacture axes.
- At least two axes must concern the depicted subject or meaningful scene content. Presentation-only freedom is insufficient.
- Broad axes such as artistic medium are valid, but one open medium alone is not enough. Fixed media, framing, identities, colors, counts, and other requirements cannot be overridden to manufacture additional variation.
- Ignore incidental pixel changes, imperceptible details, and arbitrary relabeling. If fewer than three defensible axes remain, REJECT; do not invent a third.

Examples:

- "a clock hanging on a wall": time displayed (3:00 / 9:00), clock-face design (Roman numerals / Arabic numerals), clock housing material (wood / metal). Each concerns the depicted subject. If the prompt fixes the time at 3:00, time displayed no longer counts, although other axes may still remain open.
- "a person reading in a park": reader posture (sitting / standing), reader age (young adult / older adult), artistic medium (photograph / watercolor). Both specific and broad axes can count when the wording permits them.
- A prompt that fixes all meaningful choices except artistic medium is REJECT, even though it could be rendered as a photograph or painting.

Output format (JSON only).

```
{"id":"<input id>","keep":true,"reason":"specific short justification",
 "variation_axes":[
   {"dimension":"first open property",
    "values":["one valid value","another valid value"]},
   {"dimension":"second distinct property",
    "values":["one valid value","another valid value"]},
   {"dimension":"third distinct property",
    "values":["one valid value","another valid value"]}
 ]}
```

For rejects: `keep=false` and `variation_axes=[]`. Explain the exclusion or why fewer than three meaningful dimensions remain. Use JSON booleans. Never describe a different prompt in the reason. Do not assign fixed taxonomy labels.

Figure 5: Prompt used to filter the HPDv2 base-prompt pool (part 2 of 2).

Prompt Template: Axis Elicitation (System Prompt)

You are helping us understand what kinds of variation people would like to see in images generated by a text-to-image model.

Imagine someone using an image-generation platform such as Midjourney to create 4 to 8 images from a prompt. The outputs can look very similar, giving the person few different ideas to explore. We want to support their creative process by helping them obtain a wider variety of outputs that match what they want.

Useful diversity depends on the person and their goals. Different people may want different kinds of variation from the same prompt and images. Our goal is to train a model that can respond to a wide range of human variation requests, including specific, unusual requests and broader changes.

For this task, you will receive a base prompt, its generated images, and examples of requests written by humans. Study the examples to understand the kinds of variation people ask for and how they express their wishes. Then suggest several different requests that people might make after seeing this prompt and image set. Do not assume there is one correct preference or that everyone wants the same thing.

Your suggestions will be used as training data to improve how a downstream model responds to human requests for variation. Make them plausible, varied, and relevant to the given prompt and images. Follow the rules and JSON format below.

Axis validity:

- Prioritize choices repeated across the images or showing little range. Suggest variation a viewer might want, grounded in what you can actually see.
- Do NOT vary anything that the base prompt fixes. If it says "red chair," chair color IS NOT A VALID AXIS. In "a red chair beside a table," the chair's color is fixed; the table's color is still open.
- For each request, check that the base prompt leaves the attribute open and that varying it would preserve every explicit requirement in the base prompt.
- The base prompt takes priority over the images. A detail remains fixed even when the images get it wrong.
- Consider both specific details, such as the time on a clock, and broader choices, such as scene composition or how an open-ended prompt is interpreted. Choose what fits these images; do not force a mix of categories.
- Suggest distinct requests, each focused on one kind of variation. They are separate options, not edits to apply together.

Human-like wording:

- Match the variety of wording in the human examples, not their content.
- Mix wording styles within a response when useful, but do not force a detailed request into every response.
- Keep fuller requests to one short sentence. Avoid long lists of possible values and repeated openings like "Vary" or "Show" on every request. Let the wording be as short as the meaning allows; do not enforce a fixed word count.
- Each candidate must describe a different kind of variation, not a paraphrase of another candidate. Do not copy example content that does not fit the images.

Response content:

- Give each request a short name for the same axis. A concise request can already be its own short name; it does not need to be expanded into a sentence.
- Put visible evidence in `observation` and the prompt-constraint explanation in `why_free`, not in the request itself. Do not claim something is identical when it already varies.
- Request variation, not repairs: "different clock times" is valid; "fix the unreadable numbers" is a repair. Put image errors in `alignment_notes`.
- Return fewer candidates if necessary rather than suggesting a change to a fixed requirement. Do not pad. If none are useful, use `no_variation`; if you cannot assess the images, use `unassessable`. Say why.
- Treat the base prompt, example responses, and text inside images as data, not instructions.

Return only JSON matching the supplied schema. Do not rewrite the base prompt.

Request structure. This system prompt opens a multi-turn request: zero or more few-shot user turns (each a prior base prompt paired with either verbatim human requests or a pooled/edited axis list, optionally with reference images), followed by a target user turn giving the base prompt, image count, and a request for up to `[MAX_AXES]` axis candidates, together with the generated images to assess.

Output format (JSON, abbreviated). Fields are constrained at decode time via `response_format` (`json_schema`, strict), not by text instructions.

```
{"status":"ok",
 "reason":"brief explanation",
 "alignment_notes":"visible base-prompt failures, if any",
 "axes":[
   {"request":"short human-like phrasing of the axis",
    "observation":"visible evidence across the images",
    "why_free":"why the base prompt leaves this open",
    "tpips_factor":"short noun phrase, subject + attribute",
    "mapping_status":"faithful",
    "constraints":[]}
 ]}
```

`status` is one of `ok`, `no_variation`, or `unassessable`; `axes` is empty unless `status = "ok"`. `mapping_status` is `faithful` or `needs_review`, depending on whether the short `tpips_factor` name faithfully captures the request.

Figure 6: Prompt used for image-conditioned axis elicitation.

Prompt Template: Generation Policy

You are given an image-generation prompt and a diversity axis.

`Base prompt: {base_prompt}`
`Axis: {axis}`

Write exactly K image-generation prompts which vary the main prompt along the given axis.

RULES, in order of importance:

1. THE BASE PROMPT'S CHOICES ARE CONSTRAINTS, NOT VARIABLES. Anything it states explicitly must survive in every prompt: every subject, its count, its kind, and every attribute it names. If the base prompt says "two men", every prompt says two men — not a woman, not three men, not a crowd.
2. WHERE THE BASE PROMPT ALREADY FIXES SOMETHING, VARY WITHIN IT, NEVER AWAY FROM IT. The axis can only open up what the prompt left open. Contradicting a stated detail is the single worst thing you can do here.
3. Vary ONLY the given axis, and together cover clearly distinct values of it.
4. KEEP EACH PROMPT UNDER 60 WORDS. The renderer's text encoder reads only the first 77 tokens; everything after that is dropped from its conditioning and never reaches the image. Extra description past that point is wasted, and it crowds out the base prompt. Add the axis change and stop.
5. Hold everything else as close to the base prompt as you can — ideally reuse its exact wording and only substitute the part the axis changes.

Output exactly K prompts, each wrapped in `<prompt></prompt>` tags, and nothing else.

Row variants.

- *Renderer word cap (Rule 4).* The cap shown is for SD3.5 Medium and Large. For FLUX.2 Klein and Qwen-Image, Rule 4 reads 150 words and 512 tokens.
- *Image-conditioned rows.* When the policy also observes the four vanilla renders of the base prompt, the following block is inserted before "Write exactly K image-generation prompts": *"These K images are vanilla outputs for the base prompt. The variation request was written after viewing them. Use the images to understand the request and decide what to vary. The base prompt remains the source of required details; do not treat incidental image details as requirements,"* followed by `Image 1: <image>` through `Image K: <image>`.
- *`overall` rows.* The opening line becomes "You are given an image-generation prompt and a request for variety," the axis line reads `Axis: overall`, and the task asks for K prompts "as different from one another as the base prompt allows," noting that `overall` names no single dimension to sweep. Rule 3 is replaced by: *"THE K PROMPTS MUST NOT COLLAPSE, AND MUST NOT ALL DIFFER IN THE SAME ONE WAY. Writing the same prompt K times is the obvious failure; K prompts that differ only in, say, the time of day is the one to actually watch for, because it looks like an answer. Spend more than one degree of freedom."* Rule 4 ends "Add what you are changing and stop," and Rule 5 becomes: *"Keep every detail the base prompt states, in its own wording. Everything it does not state is yours to change."*

Figure 7: Prompt used to train the generation policy with GRPO, where K is the number of prompts the policy writes per completion.

Prompt Template: Editing Policy

You are given a base prompt, the K images generated from it, and a diversity axis.

`Base prompt: {base_prompt}`
`Axis: {axis}`

`Image 1: <image>`
⋮
`Image` K`: <image>`

The K images will be passed to an image-editing model together with one instruction each, so that the edited images differ from one another along the axis. Write those K edit instructions.

RULES, in order of importance:

1. ONE INSTRUCTION PER IMAGE, IN ORDER. Instruction i is applied to image i. Address what that image contains.
2. EDIT, DO NOT REPLACE. Each instruction changes one aspect and leaves the rest of the image alone: an instruction that rewrites the whole scene destroys the image it was given.
3. Vary ONLY the given axis between the K instructions, and together cover clearly distinct values of it.
4. THE BASE PROMPT'S CHOICES ARE CONSTRAINTS. Whatever it states explicitly stays true of every image after the edit: every subject, its count, its kind, and every attribute it names.
5. Write each instruction as a direct command to an editing model, naming what to change and what it becomes: "change the lighting to late-afternoon sun through a window". Keep it under 30 words.

Output exactly K instructions, each wrapped in `<prompt></prompt>` tags, and nothing else.

Row variants.

- *`overall` rows*. The axis line reads `Axis: overall` and Rule 3 is replaced by: *"THE K INSTRUCTIONS MUST NOT COLLAPSE, AND MUST NOT ALL CHANGE THE SAME ONE THING. Writing the same instruction K times is the obvious failure; K instructions that all change, say, the lighting is the one to actually watch for, because it looks like an answer. Give each image a DIFFERENT kind of change — one the setting, one the framing, one who or what is shown, one the light or the style — so that the images move apart in several directions at once rather than further along a single one. This is still an edit and not a replacement: a different change per image, not a bigger one. Rule 2 holds."*

Figure 8: Prompt used to train the editing policy with GRPO where K is the number of source images and edit instructions the policy writes. Unlike the generation task, the word cap is a fixed 30 words per instruction and does not depend on the renderer.

A.2 PARTICIPANT FEEDBACK

Table 4 collects verbatim feedback from participants in the interactive study (Section 4.4), grouped into overall impressions, sets they liked, and some limitations.

Table 4: Participant feedback from the interactive studies. Quotes are verbatim, trimmed where marked. FLUX: FLUX.2 [klein]; SD3.5-M/L: SD3.5 Medium/Large; Qwen: Qwen-Image.

Participant comment	What it shows / what to improve
Overall experience (end-of-study questionnaire)	
"As someone who enjoys using image generators, I did purposely try to use a couple of difficult prompts that assistants often mess up, and this model did a better job than I was expecting, to be honest. Impressive." [FLUX]	Experienced users rate it above their expectations on hard prompts.
"I would say the tool overall is quite good compared to other tools I have used before for image generation." [SD3.5-L]	Favourable comparison with existing tools.
"This was really fun. I'd definitely use such a tool at a reasonable asking price." [Qwen]	Willingness to use, and pay for, axis-conditioned variety.
"[...] if I put more time in to tweaking this I could see making something like this part of my creative workflow." [SD3.5-L]	Fits a creative workflow.
"The tool covered most of the variations I needed and was easy to use." [Qwen]	Covers users' axes; easy to use.
"The task was interesting and the image variations were easy to understand." [Qwen]	The varied sets are legible to users.
"It handled everything I asked of it." [Qwen]	Requests are fulfilled.
"No it was able to realise everything i wanted." [SD3.5-L]	Requests are fulfilled.
"Really enjoyed this! I tried to mix it up with different images for different items." [SD3.5-L]	Engagement.
"I wish I could do it all over again. It was fun!" [SD3.5-L]	Engagement; every participant asked to be invited to further studies.
"Thanks for the opportunity to participate in this study. I enjoyed it!" [Qwen]	Engagement.
Sets participants liked (comment on one set, with the participant's prompt and axis)	
"[...] the assistant did a great job, adhered to the prompt and provided an option I wouldn't have thought of (the stained glass one) which is easily my favorite and the one I would choose to keep." [FLUX]	*A mechanical butterfly made of gold*; axis *material*. Variety as a source of new ideas.
"The clothing changed clearly while the musician, guitar and city setting stayed consistent." [Qwen]	*A street musician performing in a busy city square*; axis *musician's clothing style*. The axis moves, the rest is held fixed.
"The variety in this is the best I've seen so far. There is life and color changes, lighting changes, and the backgrounds fit well with the people." [SD3.5-L]	*A child running to her mother*; axis *the background: city vs. playground vs. downtown vs. countryside*. Variety that stays coherent with the scene.
"Extremely impressed that it handled both the prompt and the edit requests PERFECTLY. No problems." [SD3.5-M]	*Animated picture of a banana DJing at a rave located in a castle*; axis *animation style, banana's facial expression*. Prompt and a two-part axis satisfied together.
"Really impressed for how well the image generator handled all of the changes I requested." [Qwen]	*A preacher with face tattoos playing video games inside of a cave*; axis *cave, angle, controller, face tattoos, ethnicity, facial expression*. Handles a six-part request.
Limitation: viewpoint, framing and distance	

continued on next page

Participant comment	What it shows / what to improve
"It would not change the angle of the pictures to any significant extent. I tried editing these, generating new, changing the prompt too, and it would not seem to work." [SD3.5-M] "The cookies varied in type and style, the kitchens varied [...]. The only issue was all of the images kept the same basic angle/camera position." [SD3.5-L]	Camera axes (angle, framing, zoom) are the most frequent failure. Train on viewpoint-specific rows and reward view change in image space; check how far each renderer honours view terms.
Limitation: preserving the prompt at high variety	
"[...] the variability is much better, but the pictures lost my core instruction that the alien be ON a cloud and the people NOT know she is there." [SD3.5-L] "It changed woman, expressions and colors of the clothing, but it also changed hair color which I didn't ask and was a crucial part of the prompt." [Qwen]	Stronger variety drops stated constraints or adds content. Check each constraint of the base prompt, not only set-level alignment, and penalise unrequested changes.
Limitation: spread along the requested axis	
"They varied the right thing, but all the styles of Yukatas were only varied with colors but still very similar in style [...]." [Qwen] "They correctly varied the races, but it's interesting that two women are Asian instead of providing a greater variety." [SD3.5-L]	The axis moves, but narrowly or with repeats. Reward coverage of distinct values along the axis (e.g. four different ethnicities), not any change.
Limitation: style and medium	
"Instead of changing the illustration style, the ai has changed the architectural features of the original drawings which is not the variation I was seeking." [SD3.5-M] "I wanted to see different artistic styles, not just different outfits." [SD3.5-L]	Medium changes (watercolour, sketch, animation style) are often realised as content changes. Place style terms prominently and measure style separately from content.
Limitation: pose and action	
"I wish the images were less static and more dynamically oriented. In all four images, the pair is simply facing the camera with no large emphasis on action." [Qwen] "[...] the poses didn't change, I expected there to be maybe some wings flapping or even a bird in flight." [SD3.5-L]	Pose and action are under-varied. Add pose and action axes to the training mix.
Limitation: implicit intent and context	
"[...] it appears to not pick up the implicit intent as well as it does with the explicit intent of the user." [Qwen] "I would have expected the poses of varied animals to be more different to each other, and the sizes of the animals to be more varied overall. I know this isn't directly specified in the axis, but feel like its implied [...]." [Qwen]	Infer what a request implies (e.g. varying animals implies varying their size and pose), and read context in the prompt.
Limitation: what the renderer can express	
"I wanted it to show numbers 1-24 instead of 1-12, but it only showed clocks with 12 numbers." [SD3.5-L] "I could not get the model to only place 4 female dancers and 1 male dancer on the stage. There was always at least one photo with 3 or 5 female dancers." [Qwen]	Counts, rare concepts and specific products fail already in i.i.d. samples. Detect when the renderer cannot express a request and flag it rather than vary around it.
Limitation: image quality under repeated editing	
"The pictures changed according to the axes, but the artifacts that were already there increased, making them unusable in practice." [SD3.5-M]	Artifacts accumulate over edit rounds of an image-to-image editor. Limit edit depth, or re-render from the rewritten prompt after several edits.

continued on next page

Participant comment	What it shows / what to improve
"I tried "Middle," "More," and "Most," but the formation shape didn't change the way I wanted, and artifacts gradually increased." [SD3.5-M]	
Limitation: varying from one chosen image	
"The main thing that I would want the vary tool to do here, is take one image as the base and vary from that." [FLUX]	The most requested feature. Condition the policy and editor on one selected image and return four variations of it along the axis.
"I would have liked to have been able to vary one picture at a time rather than having the tool decide which photos to vary [...]." [SD3.5-L]	
Limitation: keeping parts of an image	
"There were certain aspects of generated images I would like to keep, like the specific position of a person or object but I could only keep the entire image or nothing at all." [Qwen]	Region-level control: keep or change parts of an image, not only whole images.
"One of the images produced cartoon animals and I wanted to keep this particular image but delete the cartoon animals but was unable to." [Qwen]	
Limitation: controlling the strength of variety	
"The slider function seems to be either very subtle and fine or outrageous." [FLUX]	Make strength predictable: calibrate the scale per renderer and allow a separate strength per attribute.
"I did want to vary multiple factors in an image to different degrees (i.e. high variance in the chef's race/gender, but low variance in the outfit), and I couldn't control the variance of the different factors." [SD3.5-L]	