# UNDERSTANDING AND ENHANCING CONDITIONING ROBUSTNESS OF TEXT TO IMAGE DIFFUSION MODELS

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

## ABSTRACT

Recent advances in diffusion models have significantly improved the quality of text-to-image generation. Yet visual fidelity alone is not sufficient: reliable conditioning is essential for predictable control, and its limitations continue to constrain the broader success of these models. In particular, the sensitivity of generated images to non-semantic perturbations of the input prompt is a widely recognized problem. In this work, we develop a simple perturbative model of the mechanism underlying this instability and validate it through targeted experiments. We further introduce dedicated metrics and a benchmark (CoRBen) for systematically evaluating prompt robustness. The resulting theory and empirical findings motivate the $\varepsilon^*$ sensitivity measure and a family of training-free plug-and-play methods that improve generation stability at inference overheads of 100%, 17%, and 3%. Our code is available at `https://anonymous.4open.science/r/conditioning-robustness/`.




Figure 1: Generations from semantically equivalent prompts. Top: FLUX; bottom: FLUX.LA$_3$. Prompts, from left to right: "in front of a rhinoceros sits a motorcycle", "there is a motorcycle positioned in front of a rhinoceros", "motorcycle in front of rhinoceros", "a stop sign", "a halt sign", and "here stands a stop sign".

## 1 INTRODUCTION

Diffusion models have achieved remarkable progress in generative modeling (Dhariwal & Nichol, 2021; Saharia et al., 2022; Yu et al., 2024) and now deliver state-of-the-art visual quality across a broad range of tasks (Luo et al., 2026; Lv et al., 2025; Xie et al., 2026). Despite this strong visual fidelity, reliable adherence to conditioning signals remains a key limitation of modern text-to-image (T2I) models (Zhu et al., 2025; Rosenman et al., 2024; Betker et al., 2023). One important manifestation of this limitation is the sensitivity of generated images to semantics-preserving transformations of the input prompt. We illustrate this behavior in Figure 1 using generations from FLUX.1-dev (Labs, 2024) (FLUX) and its LA$_3$ variant. Invariance to paraphrasing has been explicitly identified as an important dimension of T2I robustness (Bakr et al., 2023). More strikingly, even minor semantics-preserving textual perturbations can induce substantial changes in generated images (Zhang et al., 2024). Recent evaluations further distinguish between linguistic permutations that alter the meaning of a prompt



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.

and those that preserve it. Ideally, visual semantics should change in the former case and remain invariant in the latter. However, strong contemporary T2I models exhibit visual variation in both settings and struggle to distinguish meaning-changing from meaning-preserving transformations (Zhu et al., 2025).

Although the sensitivity of conditioning to semantics-preserving perturbations is increasingly recognized and has motivated several lines of investigation, a transparent model of how this instability arises and effective techniques for mitigating it remain lacking. In this work, we aim to close this gap. We summarize our contributions as follows:

- We introduce a simple perturbative model of sensitivity to semantics-preserving prompt transformations, governed by a single scalar $\varepsilon$ (Section 3), and systematically validate its predictions through a diverse set of experiments (Section 4).
- We introduce the **Co**nditioning **R**obustness **Ben**chmark (CoRBen) and derive $\varepsilon^*$, a theory-grounded measure of conditioning sensitivity (Sections 4.3 and 6).
- We propose Latent Averaging (LA), a theory-driven, training-free, plug-and-play method for improving conditioning robustness (Section 4.2).
- We scale LA using $\text{SVRG}_N^K$ and $\text{SAGA}_N^{*K}$ estimators, reducing its inference overhead to approximately 3% relative to the baseline (Section 5.3).

## 2 Related Work

Additional background on conditioned diffusion models and guidance is provided in Appendix A.

**Classifier-free guidance.** At sampler state $x_t$, let $F_\theta(x_t, c)$ and $F_\theta(x_t, \varnothing)$ denote the raw model outputs under textual condition $c$ and the unconditional input $\varnothing$, respectively. Classifier-free guidance (CFG) constructs the guided output

$$F_{\theta,s}(x_t, c) = F_\theta(x_t, \varnothing) + s\left(F_\theta(x_t, c) - F_\theta(x_t, \varnothing)\right), \tag{1}$$

where $s > 0$ is the guidance scale (Ho & Salimans, 2022). Setting $s = 1$ recovers the ordinary conditional output, whereas $s > 1$ amplifies the influence of the condition. CFG improves sample fidelity and text-image alignment, but large values of $s$ reduce diversity and can introduce oversaturation and other visual artifacts (Nichol et al., 2021; Saharia et al., 2022; Sadat et al., 2024). Modern T2I distillation pipelines often absorb the guidance operation into the learned generative dynamics (Meng et al., 2023). The resulting network treats the guidance scale as an additional conditioning variable and directly approximates the CFG output:

$$F_{\theta,s}(x_t, c, s) = F_\theta(x_t, c, s). \tag{2}$$

**Text-image alignment and prompt following.** Beyond visual fidelity, accurately translating textual conditions into generated content remains a central challenge in T2I generation. Imagen (Saharia et al., 2022) marked an influential shift toward using large pretrained language models for conditioning: it employs a frozen T5-XXL encoder and shows that scaling the text encoder yields larger improvements in image-text alignment and sample fidelity than scaling the diffusion model itself. Complementary, DALL-E 3 (Betker et al., 2023) demonstrated that improving image-text supervision through highly descriptive synthetic captions substantially enhances prompt following. Nevertheless, systematic evaluations and suggested benchmarks continue to reveal failures in realizing the supplied conditions, including omitted objects, incorrect attribute binding and counting, spatial relations, and failures on complex compositions (Huang et al., 2025; Hu et al., 2023; Ghosh et al., 2023). These limitations have motivated improvements at different stages of the conditioning pipeline: ELLA (Hu et al., 2024) replaces limited CLIP representations with richer LLM features while training only the interface between the LLM and the generative network; Attend-and-Excite (Chefer et al., 2023) addresses the failure to generate one or more subjects specified in the input prompt through a training-free intervention in cross-attention that amplifies attention scores on neglected subject tokens; NeuroPrompts (Rosenman et al., 2024) automatically rewrites user prompts into model-effective descriptions; and SCoPE (Saichandran et al., 2025) progressively introduces coarse-to-fine prompt details throughout generation. While these works primarily target the faithful realization of a given prompt, we study a complementary reliability question: whether semantically equivalent conditions induce consistent generative behavior.

**Robustness to semantics-preserving prompt variations.** HRS-Bench explicitly introduced invariance to paraphrasing and typos as dimensions of T2I robustness (Bakr et al., 2023). Qiu et al. systematically evaluated semantically controlled character-, word-, and sentence-level perturbations and found that textual variations measurably affect T2I generation, with character-level corruptions being the most disruptive (Qiu et al., 2022). Complementarily, adversarial studies showed that semantically consistent human-like errors can be selected to substantially alter model behavior (Gao et al., 2023), while ProTIP (Zhang et al., 2024) formalized robustness probabilistically in terms of preserving the output distribution under semantically similar stochastic perturbations. More recently, MMDT (Xu et al., 2025) demonstrated substantial degradation in contemporary T2I systems when prompts are rewritten in semantically equivalent but out-of-distribution linguistic styles. SemVarBench (Zhu et al., 2025) provides an even more controlled analysis, explicitly contrasting meaning-changing and meaning-preserving linguistic permutations and showing that strong T2I models frequently respond to both rather than selectively tracking the underlying semantic change. These studies reveal persistent prompt brittleness but provide neither an explanation of its underlying mechanisms nor effective techniques for mitigating it.

## 3 A Perturbative Model of Conditioning Sensitivity

We begin by observing that some textual transformations do not alter the scene described by a prompt. These include changes in voice, register, word order, and the use of exact synonyms. Hereafter we refer to such transformations, or perturbations, as *semantics-preserving*. At the same time, as noted above, established practice is to use a frozen pretrained language transformer to map the textual prompt to a conditioning representation. We emphasize that the variations listed above are meaningful for the language-modeling task on which the frozen text encoder is trained. Consequently, its output, and hence the conditioning representation $c$, are not generally invariant to semantics-preserving transformations of the textual condition. Motivated by this observation, we propose the following decomposition of $c$:

$$c = (c_s, c_n),$$

where $c_s$ represents the semantic content of the prompt, while $c_n$ contains only encoder-induced artifacts of language modeling. Accordingly, under semantics-preserving transformations, the component $c_s$ remains fixed, while only $c_n$ changes.

We next turn to the conditional generative network. To remain agnostic to the particular generative parameterization, we denote by $F_\theta(x_t, c) = F_\theta(x_t, t, c)$ the raw output of the network at state $x_t$, time $t$, and condition $c$. Depending on the model, this output may parameterize a noise prediction, a score, a denoised sample, or a flow-matching velocity field.

Ideally, the model should condition only on prompt semantics and remain invariant to semantics-preserving prompt transformations, i.e. $F_\theta(x_t, c) = F_\theta(x_t, c_s)$ should hold. In practice, however, the generative network retains an undesirable sensitivity to the non-semantic component $c_n$. We model this deviation from ideal behavior as

$$F_\theta(x_t, c) = F_\theta(x_t, c_s) + \varepsilon f_\theta(x_t, c_s, c_n), \tag{3}$$

where the perturbation term $\varepsilon f_\theta(x_t, c_s, c_n)$ captures precisely the sensitivity studied in this work. At the same time, semantically equivalent prompts often yield consistent generations, indicating that $c_s$ dominantly governs the generative dynamics. Motivated by this observation, we assume that under normal operating conditions the contribution induced by $c_n$ is a small correction and make this assumption explicit through the scale parameter $\varepsilon \ll 1$. The scale $\varepsilon$ therefore directly controls the model's sensitivity to semantics-preserving prompt transformations. Indeed, as $\varepsilon \to 0$, the dependence on the non-semantic component $c_n$ vanishes, whereas increasing $\varepsilon$ induces progressively larger deviations from the dynamics governed by $F_\theta(x_t, c_s)$.

Finally, we note that the generative models evolve through an iterative process, whether by denoising or by integrating a velocity field. Consequently, even a small perturbation introduced at an early stage can produce substantial deviations in the final output. Moreover, for a fixed $\varepsilon$, the magnitude of this effect may depend on the geometry of the sampling trajectories, and in particular the stability, of the underlying generative dynamics.

# 4 EXPERIMENTAL EVALUATION

We test the predictions of the model introduced in Section 3 using Stable Diffusion 1.5 (Rombach et al., 2022) (SD1.5) and FLUX.1-dev (Labs, 2024) (FLUX). These systems differ substantially in scale, backbone architecture (U-Net vs DiT), and generative dynamics (denoising diffusion vs flow matching), allowing us to assess whether the predicted behavior persists across distinct model families.

## 4.1 RELATION TO CFG

Substituting the decomposition in (3) into the CFG definition in (1), we obtain

$$\begin{aligned} F_{\theta,s}(x_t, c) &= F_\theta(x_t, \varnothing) + s\,(F_\theta(x_t, c) - F_\theta(x_t, \varnothing)) \\ &= F_\theta(x_t, \varnothing) + s\,(F_\theta(x_t, c_s) - F_\theta(x_t, \varnothing)) + s\varepsilon f_\theta(x_t, c_s, c_n). \end{aligned} \tag{4}$$

Thus, CFG amplifies not only the semantic signal $(F_\theta(x_t, c_s) - F_\theta(x_t, \varnothing))$ but also the sensitivity-inducing term $\varepsilon f_\theta(x_t, c_s, c_n)$.

**Evaluating CFG-induced prompt sensitivity.** We evaluate the prediction of Equation (4) on 100 prompts from PartiPrompts (Yu et al., 2022). For every prompt, we use an LLM with few-shot prompting to construct six alternative formulations intended to preserve its visual semantics. We generate an image for each of the resulting seven formulations across different guidance scales. All generations use the same initial noise (seed 42). For each prompt group and guidance scale, we compare all $\binom{7}{2} = 21$ image pairs generated using the same $c_s$ but different semantics-preserving prompt perturbations, i.e. different $c_n$. We quantify generation consistency using two complementary image-space metrics: LPIPS distance (Zhang et al., 2018), which measures perceptual differences between images using calibrated deep visual features, and cosine similarity between DINOv2 (Oquab et al., 2023) image representations, which additionally captures higher-level visual content and semantic similarity. Accordingly, lower LPIPS distance and higher DINOv2 similarity indicate greater consistency between generations. Additional experimental setup details are reported in Appendix E.

We note that FLUX uses guidance-distilled flow matching (2), with the guidance value embedded as an additional conditioning variable rather than through the explicit CFG combination (1). Consequently, this experiment does not directly apply to FLUX.

The SD1.5 results correspond to the curves labeled "SD1.5" in the top panels of Figure 2. The monotonic increase in LPIPS with the CFG scale is consistent with the predicted amplification of the perturbation term $s\varepsilon f_\theta(x_t, c_s, c_n)$ in Equation (4). DINOv2 cosine similarity instead exhibits a nonmonotonic trend, suggesting competition between semantic amplification and prompt sensitivity. At moderate guidance scales, the semantic term $s\,(F_\theta(x_t, c_s) - F_\theta(x_t, \varnothing))$ dominates and similarity increases. Beyond $s = 10$, similarity decreases as the perturbation term becomes dominant. This behavior is consistent with the known saturation of semantic gains at high CFG scales (Saharia et al., 2022).

## 4.2 LATENT AVERAGING

Equation (4) motivates a second experiment. Given $N$ conditioning representations $c^i$ whose semantic components coincide, $c_s^i = c_s$, we evaluate the generative network for each representation and average its outputs, obtaining

$$\overline{F}_{\theta,s}(x_t, \{c^i\}_{i=1}^N) = F_\theta(x_t, \varnothing) + s\,(F_\theta(x_t, c_s) - F_\theta(x_t, \varnothing)) + s\varepsilon \cdot \frac{1}{N}\sum_{i=1}^{N} f_\theta(x_t, c_s, c_n^i). \tag{5}$$

As discussed above, $f_\theta(x_t, c_s, c_n^i)$ captures the component of the prompt signal that is uninformative for the generative task. Moreover, any association between this non-semantic component and the target image in the training data is expected to be at the level of dataset noise. This motivates modeling the perturbation terms across prompt formulations as independent, zero-mean random variables. Averaging $N$ such terms reduces their variance by a factor of $N$. In our setup, this is

equivalent to reducing the effective perturbation scale by a factor of $\sqrt{N}$ (see Figures 6 and 7), thereby improving conditioning robustness.

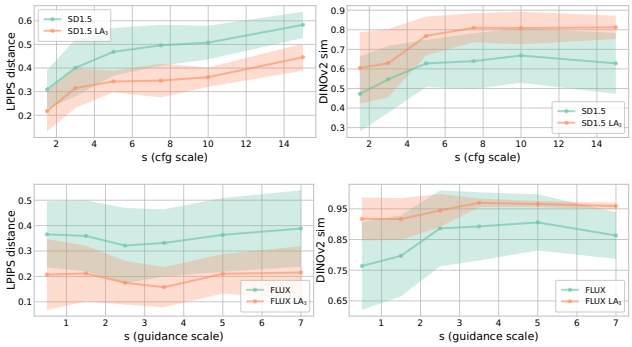



Figure 2: Generation consistency across guidance scales. Top: SD1.5 and SD1.5.LA$_3$; bottom: FLUX and FLUX LA$_3$. Left: LPIPS; right: DINOv2 cosine similarity.

We call the resulting scheme *Latent Averaging* (LA) and denote averaging over $N$ semantically equivalent prompts by LA$_N$. When LA is applied to a base generation method $\mathcal{M}$, we write $\mathcal{M}$.LA$_N$. Notably, Equation (5) modifies only the conditional prediction supplied to CFG, while all other components of the underlying sampling algorithm remain unchanged. LA is therefore agnostic to the particular choice of sampling method. For models with distilled guidance (2), we expect analogous behavior as their learned dynamics approximate the CFG-guided teacher field.

We conduct the corresponding experiment using the protocol described in Section 4.1. For LA$_3$, at each generation step, the network output for the input prompt is averaged with those for two fixed, randomly selected semantics-preserving variants of the same prompt. Figure 2 shows that LA$_3$ consistently improves stability for both SD1.5 and FLUX across all guidance scales.

### 4.3 Conditioning Robustness Benchmark (CoRBen)

To further investigate the improvement in conditioning stability provided by SD1.5.LA$_3$, we introduce the **Co**nditioning **R**obustness **Ben**chmark (CoRBen), which is partially based on the benchmark of Qiu et al. (2022). The evaluation population is the 5,000-image MS COCO Karpathy test split (Lin et al., 2014; Karpathy & Fei-Fei, 2015). Each image has multiple human-written captions; we designate one as the original prompt and use the remaining captions to form the *recaptions* group. Following Qiu et al. (2022), we additionally construct character-, word-, and sentence-level perturbation groups for each prompt. For the character- and word-level groups, we select the perturbation types and severity levels that least affect semantics.

Although these four groups are useful for measuring general prompt sensitivity, they have limited practical relevance: $(i)$ character-level token insertions, deletions, and substitutions can often be corrected during prompt preprocessing, while $(ii)$ recaptions and word- or sentence-level perturbations may substantially alter visual semantics, including colors, object counts, and depicted objects. CoRBen therefore introduces a fifth group, *Fixed-Semantics Perturbations* (FSP), explicitly constrained to preserve visual semantics and remain grammatical. In total, we obtain 42,967 additional FSP prompts.

We report FID, CLIP-FID, and Object Recall (OR), as these metrics are more directly interpretable for our study than the MMI and MOR measures proposed by Qiu et al. (2022). OR is the fraction of atomic visual concept instances explicitly specified in a prompt that are detected in the generated image. Following Qiu et al. (2022), we use GLIP-L (Li et al., 2022) for concept detection. In contrast to their evaluation based on COCO ground-truth instance categories, we extract 13,717 atomic concept records directly from the original prompts. Thus, OR evaluates only concepts explicitly requested by the prompt. In more detail, let $s_c^i$ denote the $i$-th concept extracted from prompt $c$, $p_c^i$ be the detector confidence assigned to that concept in the corresponding generated image, and let $\tau$ be the confidence threshold. We define

$$\mathrm{OR}_\tau = \frac{\sum_{c,i} \mathbb{I}\left[p_c^i > \tau\right]}{|\{s_c^i\}_{c,i}|}. \tag{6}$$

**Distributional sanity check.** For FID and CLIP-FID, we select one perturbation per source anchor and use the same frozen selection for SD1.5 and SD1.5.LA$_3$. Table 1 shows only small differences: SD1.5.LA$_3$ performs better for some family-metric pairs and SD1.5 for others. This is consistent

with the intended role of LA: improving conditioning stability without systematically shifting the generated-image distribution.

Table 1: CoRBen results. $LA_3$ denotes SD1.5.$LA_3$. The original-prompt $OR_{0.5}$ is 91.14%.

| | FID ↓ | | CLIP-FID ↓ | | $OR_{0.5}$ ↑ | | $OR^*_{0.5}$ ↑ | | |
|---|---|---|---|---|---|---|---|---|---|
| Perturbation | SD1.5 | $LA_3$ | SD1.5 | $LA_3$ | SD1.5 | $LA_3$ | Original | SD1.5 | $LA_3$ |
| Alt. captions | 24.77 | **24.43** | 11.09 | **10.93** | 82.80 | **88.40** | 83.71 | 59.06 | **76.15** |
| Character | 23.98 | **23.65** | **9.61** | 9.70 | 86.62 | **90.38** | 79.00 | 61.46 | **76.24** |
| Sentence | **24.77** | 25.08 | **11.03** | 11.08 | 90.59 | **91.14** | 65.83 | 58.97 | **65.95** |
| Word | **24.96** | 25.09 | **11.32** | 11.41 | 89.38 | **90.65** | 72.84 | 64.77 | **71.46** |
| FSP | **24.37** | 25.06 | **10.96** | 11.14 | 90.48 | **91.40** | 72.12 | 68.50 | **73.22** |

**Object recall.** $OR_{0.5}$ is also reported in Table 1. The original prompt attains 91.14%. Across perturbation families, SD1.5 drops by 0.55%-8.34%. We note that SD1.5.$LA_3$ recovers a substantial fraction of this drop. Moreover, for grammatical sentence-level reformulations, it matches the original-prompt result at 91.14%. Notably, on FSP, where every variant is explicitly constrained to preserve all visually grounded content of its original prompt, SD1.5.$LA_3$ reaches 91.40%, exceeding the original-prompt baseline. This is the expected signature of suppressing non-semantic conditioning variation: LA reduces the probability that a requested concept disappears under the perturbation term $\varepsilon f_\theta(x_t, c_s, c_n)$.

**Sensitivity-focused recall.** To isolate this effect more directly, we note that some concept targets are uninformative about prompt robustness: they are either recovered from every formulation or absent from all of them, so the perturbation does not change their outcome. We therefore retain only targets for which SD1.5 changes its decision across prompt variants - detected for at least one formulation and missed for at least one - and denote recall on this subset by $OR^*_\tau$. For $\tau = 0.5$ the filter retains 15,925 of 62,011 concept records, spanning 4,044 of 5,000 source prompts and 84,463 of 325,953 concept-prompt pairs. Results are reported in Table 1 and visualized in Figure 3. The sensitivity-focused subset exposes a substantially larger effect.

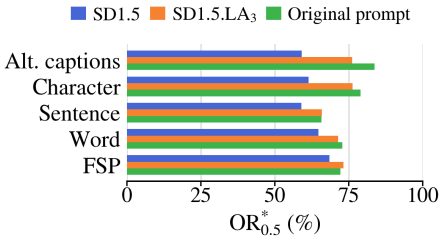



Figure 3: $OR^*_{0.5}$ across perturbations.

Results for $OR_{0.7}$ and $OR^*_{0.7}$, together with detailed retained-population counts, are reported in Appendix F.1.

## 5 Efficient Conditioning Stabilization

Latent Averaging in Section 4.2 improves conditioning robustness, but its direct implementation is expensive. For SD1.5.$LA_3$, each sampling step requires two additional conditional evaluations relative to the standard CFG baseline of one conditional and one unconditional evaluation, thereby doubling the number of function evaluations. For models with distilled guidance (2), whose baseline requires only one function evaluation per step, $LA_3$ triples the cost. We next identify where averaging is most consequential and use this structure to reduce its computational cost.

### 5.1 Timestep Importance

Different sampling timesteps need not contribute equally to the final image (Hertz et al., 2022; Pang et al., 2025). This asymmetry is particularly relevant to conditioning robustness: a perturbation introduced early in reverse diffusion or velocity-field integration can propagate through all subsequent updates and produce a larger final displacement.

We measure this effect using the prompt set introduced in Appendix E.1. We index the network evaluations in chronological sampling order by $k \in \{0, \dots, T\}$, where $k = 0$ is the highest-noise evaluation. For each ordered prompt pair $(c, c')$ and replacement step $k^\star$, we first generate a reference image using $c$ throughout. We then repeat the generation with the same initial noise, replacing $c$ by the semantically equivalent condition $c'$ only at step $k^\star$. Sensitivity is measured by the squared Euclidean distance between the DINOv2 representations of the resulting images.

Figure 4 shows a pronounced temporal asymmetry: substitutions during the first evaluations induce the largest changes in the final image. Additional construction and sampling details are provided in Appendix E.2.

### 5.2 Averaging at critical timesteps

Motivated by the results of Section 5.1, we average the SD1.5 noise predictions only during the first five evaluations ($k \in \{0, \dots, K-1\}, K = 5$) and use the original prompt thereafter. We denote this partial-averaging variant by SD1.5.LA$_N^K$ and FLUX.LA$_N^K$. For $N = 3$, restricting averaging to five steps adds only $(N-1) \times 5 = 10$ evaluations, reducing the overhead to $10/60 \approx 16.7\%$ and $10/29 \approx 34.5\%$ for SD1.5 and FLUX, respectively.

### 5.3 Scaling further with variance-reduced estimators

At an averaged step, the desired conditional output follows the notation of Equation (5) and is given by

$$\overline{F}_\theta(x_{t_k}, \{c^i\}_{i=1}^N) = \frac{1}{N} \sum_{i=1}^{N} F_\theta(x_{t_k}, c^i), \qquad (7)$$

which normally requires $N + 1$ network evaluations. We reduce this cost using SVRG (Johnson & Zhang, 2013) and SAGA$^*$, an adaptation of the SAGA (Defazio et al., 2014); both provide computationally efficient estimators of this mean. We retain the subscript and superscript convention from Sections 4.2 and 5.2.

**SVRG$_N^K$.** Similar to LA above, we modify only the conditional branch by replacing its ordinary model output with a new estimator. The unconditional CFG branch and all other components of the underlying sampling algorithm remain unchanged. Algorithm 1 constructs $N - 1$ semantics-preserving variants of the original condition in Line 2. Lines 3-5 evaluate and cache the complete mean and perform the first sampling update. During the next four evaluations, Lines 6-9 use a single conditional network call and an SVRG correction relative to the cached reference. From the sixth evaluation onward, Lines 10-11 revert to the original prompt conditioning.

The SVRG correction is an unbiased estimator of the latent average in (7). Standard SVRG periodically recomputes the complete reference mean. Guided by Section 5.1, we instead use a single reference at $k = 0$ and apply the estimator only through $k = 4$. This requires only $N - 1 = 2$ additional conditional evaluations for $N = 3$, corresponding to approximately $3.3\%$ and $6.9\%$ overhead over the standard guidance scheme for SD1.5 and FLUX, respectively. Appendix C gives more detail on the SVRG estimator, its interpretation, and a proof of unbiasedness.

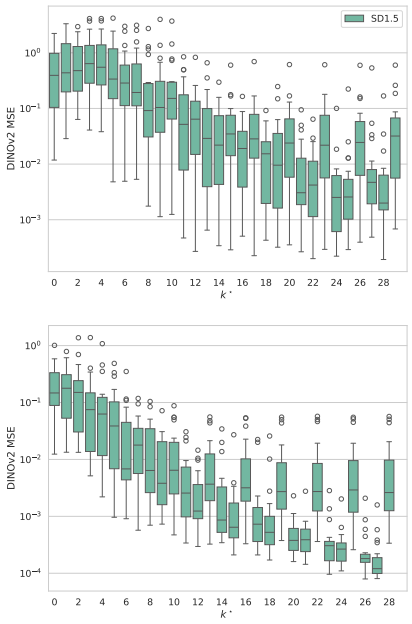



Figure 4: Final-image DINOv2 displacement caused by a one-step prompt substitution. Top: SD1.5; bottom: FLUX.

**Algorithm 1:** SVRG$_N^K$

1: **Input:** Condition $c^1$, horizons $K \le T$, initial state $x_{t_0}$
2: Construct semantic preserving $c^2, \dots, c^N$
3: $g_0^i \leftarrow F_\theta(x_{t_0}, c^i), \quad i = 1, \dots, N$
4: $\bar{g}_0 = N^{-1} \sum_{i=1}^N g_0^i$
5: Update $x$ using $\bar{g}_0$
6: **for** $k = 1, \dots, K-1$ **do**
7: Sample $i_k \sim \mathrm{U}\{1, \dots, N\}$
8: $\hat{g}_k \leftarrow F_\theta(x_{t_k}, c^{i_k}) - g_0^{i_k} + \bar{g}_0$
9: Update $x$ using $\hat{g}_k$
10: **for** $k = K, \dots, T-1$ **do**
11: Update $x$ using $F_\theta(x_{t_k}, c^1)$

**Algorithm 2:** SAGA$_N^{*K}$

1: **Input:** Condition $c^1$, horizons $K \le T$, initial state $x_{t_0}$
2: Construct semantic preserving $c^2, \dots, c^N$
3: $f_0^\varnothing \leftarrow F_\theta(x_{t_0}, \varnothing)$
4: $g^i \leftarrow F_\theta(x_{t_0}, c^i) - f_0^\varnothing, \quad i = 1, \dots, N$
5: $\bar{g} = N^{-1} \sum_{i=1}^N g^i$
6: Update $x$ using $f_0^\varnothing + \bar{g}$
7: **for** $k = 1, \dots, K-1$ **do**
8: Sample $i_k \sim \mathrm{U}\{1, \dots, N\}$
9: $f_k^\varnothing \leftarrow F_\theta(x_{t_k}, \varnothing), f_k^{i_k} \leftarrow F_\theta(x_{t_k}, c^{i_k})$
10: $\hat{g}_k = \bar{g} - g^{i_k} + f_k^{i_k}$
11: Update $x$ using $\hat{g}_k$
12: $\bar{g} = \bar{g} + N^{-1}(f_k^{i_k} - f_k^\varnothing - g^{i_k})$
13: $g^{i_k} = f_k^{i_k} - f_k^\varnothing$
14: **for** $k = K, \dots, T-1$ **do**
15: Update $x$ using $F_\theta(x_{t_k}, c^1)$

**SAGA$_N^{*K}$.** A naive transfer of SAGA to LA destabilizes the sampling process; Appendix D analyzes this failure. SAGA$^{*K}$ (Algorithm 2) instead stores condition residuals relative to the unconditional CFG branch, removing temporal drift from the memory states.

In more detail, Line 2 constructs the semantic variants. Lines 3-6 evaluate the initial unconditional branch, initialize the residual table $g^i = F_\theta(x_{t_0}, c^i) - f_0^\varnothing$ and its mean, and perform the first sampling update. During the next $K-1$ steps, Lines 7-13 sample one condition; Line 9 computes the unconditional and sampled conditional outputs, Line 10 forms the $\text{SAGA}_N^{*K}$ estimate used to perform a sampling step in Line 11. Lines 12-13 then update the running mean and the sampled cache entry. From step $K$ onward, Lines 14-15 revert to the original prompt. We use $K = 5$ in our experiments.

The estimator has the form

$$\hat{g}_k = \bar{g} - g^i + \big(F_\theta(x_{t_k}, c^i) - F_\theta(x_{t_k}, \varnothing)\big) + F_\theta(x_{t_k}, \varnothing) = \bar{g} - g^i + F_\theta(x_{t_k}, c^i).$$

This yields an unbiased approximation of the latent average in Equation (7).

Since standard CFG already evaluates the unconditional branch, $\text{SAGA}_N^{*K}$ requires one conditional evaluation per non-initial step, matching standard CFG, and reverts to the original prompt after the first $K$ steps. For $N = 3$, initialization adds $N - 1 = 2$ evaluations to the 60-evaluation SD1.5 baseline, again corresponding to approximately $3.3\%$ overhead.

Guidance-distilled architectures, however, do not evaluate an unconditional branch. For FLUX, we therefore replace the unconditional reference $f_k^\varnothing \leftarrow F_\theta(x_{t_k}, \varnothing)$ with a fixed conditional reference $f_k^{i^*} \leftarrow F_\theta(x_{t_k}, c^{i^*})$, e.g. $i^* = 1$. For $N = 3$ and $K = 5$, this requires $N - 1 = 2$ additional function evaluations at initialization and one at each of the subsequent $K - 1$ steps, adding $6/29 \approx 20.7\%$ overhead to the baseline scheme.

For further details on $\text{SAGA}_N^{*K}$ scheme we refer to Appendix D.

### 5.4 Evaluation of scaled Latent Averaging

We evaluate $\text{LA}_3^5$, $\text{SVRG}_3^5$, and $\text{SAGA}_3^{*5}$ using the protocol described in Section 4.1. Figure 5 compares these variants with the corresponding baselines and full $\text{LA}_3$ across guidance scales.

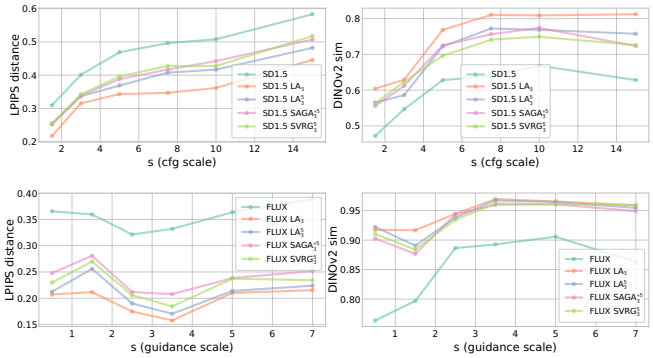



Figure 5: Generation consistency of full and scaled LA variants across guidance scales. Top: SD1.5; bottom: FLUX. Left: LPIPS; right: DINOv2 cosine similarity.

The scaled variants retain roughly half or more of the improvement over the baseline across the evaluated settings. The preservation is particularly strong for FLUX: the scaled LPIPS curves remain close to the full $\text{LA}_3$ curve, while their DINOv2 similarities are nearly indistinguishable from full $\text{LA}_3$ for $s \geq 2.5$. For reference, guidance scales commonly used in practice are 7.5 for SD1.5 and 3.5 for FLUX, both within the regimes where these gains persist.

Overall, reducing the additional number of function evaluations by up to approximately 33 times produces a much smaller reduction in robustness gains. With overheads as low as 3.3% for SD1.5 and 6.9% for FLUX, the scaled methods preserve a substantial fraction of the benefit of full LA at modest additional inference cost.

## 6 Conditioning Sensitivity Measures

Section 3 represents sensitivity to the non-semantic condition component $c_n$ through a single scale parameter $\varepsilon$. In practice, however, only the generative network outputs $F_\theta(x_t, c)$ are observable, so this latent scale is not directly identifiable. We give an operational construction below and defer its complete derivation to Appendix B.

Let $\mathcal{C}$ denote the uniform distribution over a set of semantically equivalent conditions. We define

$$\overline{F}_{\theta,t}(\mathcal{C}) = \mathbb{E}_{c' \sim \mathcal{C}} F_\theta(x_t, c'), \qquad \Delta_{\theta,t}(c') = F_\theta(x_t, c') - \overline{F}_{\theta,t}(\mathcal{C}). \tag{8}$$

Under the centered-perturbation assumption $\mathbb{E}_{c' \sim \mathcal{C}} f_\theta(x_t, c'_s, c'_n) = 0$, Equation (3) yields $\overline{F}_{\theta,t}(\mathcal{C}) = F_\theta(x_t, c_s)$; combining this identity with Equations (3) and (8) gives $\Delta_{\theta,t}(c') = \varepsilon f_\theta(x_t, c_s, c'_n)$.

We next normalize the perturbation direction to have the same expected magnitude as the semantic component:

$$\widetilde{f}_{\theta,t}(c') = \Delta_{\theta,t}(c') \frac{\|\overline{F}_{\theta,t}(\mathcal{C})\|}{\mathbb{E}_{\tilde{c} \sim \mathcal{C}} \|\Delta_{\theta,t}(\tilde{c})\|}, \qquad \varepsilon_{c,t} = \frac{\mathbb{E}_{\tilde{c} \sim \mathcal{C}} \|\Delta_{\theta,t}(\tilde{c})\|}{\|\overline{F}_{\theta,t}(\mathcal{C})\|}. \tag{9}$$

Table 2: $\varepsilon$ and $\varepsilon^*$.

| Method | $\varepsilon \downarrow$ | $\varepsilon^* \downarrow$ |
|---|---|---|
| SD1.5 | 0.023 | 0.015 |
| SD1.5.LA$_3$ | 0.013 | 0.008 |
| FLUX | 0.025 | 0.056 |
| FLUX.LA$_3$ | 0.014 | 0.033 |

This produces the exact analogue of Equation (3),

$$F_\theta(x_t, c') = \overline{F}_{\theta,t}(\mathcal{C}) + \varepsilon_{c,t} \widetilde{f}_{\theta,t}(c'),$$

with $F_\theta(x_t, c_s)$, $f_\theta$, and $\varepsilon$ replaced by their observable counterparts $\overline{F}_{\theta,t}$, $\widetilde{f}_{\theta,t}$, and $\varepsilon_{c,t}$. Once $\mathcal{C}$ is specified, $\varepsilon_{c,t}$ is directly computable from network outputs through Equations (8) and (9).

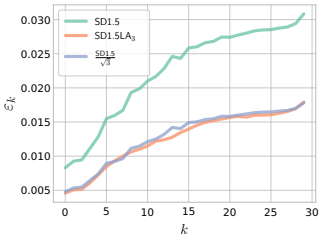
We aggregate this local quantity over the dataset $\mathcal{D} = \{\mathcal{C}_i\}_{i=1}^{100}$ used in Section 4.1, where each $\mathcal{C}_i$ comprises conditions with the same semantic component. The resulting global measure of the generative model's sensitivity to non-semantic prompt perturbations is

$$\varepsilon = \mathbb{E}_{\mathcal{C}_i \sim \mathcal{D}} \, \mathbb{E}_{t \sim \mathrm{Uniform}(\mathcal{T})} \left[ \varepsilon_{c_i,t} \right]. \tag{10}$$

The uniform temporal average in Equation (10) treats all time steps equally. However, equal-magnitude perturbations at different time steps need not have equal downstream effects. Section 5.1 quantifies this asymmetry through the final-image displacement $I_t$ induced by a one-step condition substitution. Normalizing these scores as $\pi_t = I_t / \sum_{u \in \mathcal{T}} I_u$ weights each step according to its influence on the final output and yields the following trajectory-aware metric:

$$\varepsilon^* = \mathbb{E}_{\mathcal{C}_i \sim \mathcal{D}} \, \mathbb{E}_{t \sim \pi} \left[ \varepsilon_{c_i,t} \right] = \mathbb{E}_{\mathcal{C}_i \sim \mathcal{D}} \, \mathbb{E}_{t \sim \mathrm{Uniform}(\mathcal{T})} \left[ |\mathcal{T}| \pi_t \varepsilon_{c_i,t} \right].$$

Table 2 and Figures 6 and 7 report the measurements. The results are consistent with the hypothesis in Section 4.2 of a $\sqrt{N}$-fold reduction in the perturbation scale. This scaling is observed both at individual sampling steps (Figure 6) and overall across different values of $N$ (Figure 7).



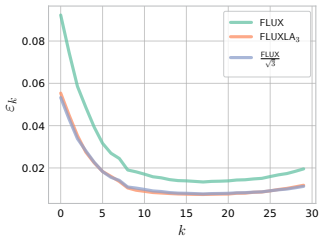




Figure 6: $\varepsilon_k = \mathrm{E}_{\mathcal{C}_i \sim \mathcal{D}}[\varepsilon_{c_i,t}]$. Top: SD1.5; bottom: FLUX.

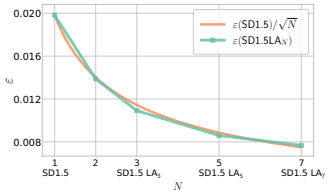



Figure 7: $\varepsilon(\mathrm{SD1.5.LA}_N)$ vs. $N$.

## CONCLUSION

We studied the sensitivity of text-to-image diffusion models to semantics-preserving prompt transformations. Our perturbative formulation isolates this failure mode through a single sensitivity scale and predicts how it interacts with guidance and iterative sampling. The resulting analysis motivated the Conditioning Robustness Benchmark (CoRBen), the trajectory-aware $\varepsilon^*$ metric, and Latent Averaging, a training-free plug-and-play method that suppresses prompt-specific variations. Experiments with SD1.5 and FLUX support the model's predictions and show that LA improves generation consistency. Finally, by concentrating computation at influential early steps and using variance-reduced mean-estimators, the scaled variants preserve a substantial fraction of the full LA benefit with overheads as low as 3.3% for SD1.5 and 6.9% for FLUX. Together, these results provide a unified account of prompt sensitivity and a practical route toward more reliable conditioning.

**Future work.** Latent Averaging also provides a natural teacher for robustness distillation. A student model could be trained to reproduce the averaged prediction field from a single prompt. Such a procedure could retain the stability benefits of LA while restoring distilled baseline inference cost.

## AI USE STATEMENT

We used large language models (LLMs) to assist with polishing the manuscript, finding related work and benchmarks, and assessing novelty. LLMs also generated synthetic prompt variations intended to preserve visual semantics (Appendix E.1), constructed CoRBen's Fixed-Semantics Perturbations (Appendix F), and extracted prompt-derived concepts and quantities for object recall evaluation (Appendix F.5). These sections document the generation and validation procedures. The authors remain responsible for all content of the manuscript, including its claims, references, and conclusions.

## REFERENCES

Eslam Mohamed Bakr, Pengzhan Sun, Xiaoqian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *2023 ieee/cvf international conference on computer vision (iccv)*, pp. 19984–19996. IEEE, 2023.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM transactions on Graphics (TOG)*, 42(4):1–10, 2023.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-$\backslash alpha$: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International conference on learning representations*, volume 2024, pp. 57611–57640, 2024.

Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. Cfg++: Manifold-constrained classifier free guidance for diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 30824–30850, 2025.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

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, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Hongcheng Gao, Hao Zhang, Yinpeng Dong, and Zhijie Deng. Evaluating the robustness of text-to-image diffusion models against real-world attacks. *arXiv preprint arXiv:2306.13103*, 2023.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

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.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *2023 ieee/cvf international conference on computer vision (iccv)*, pp. 20349–20360. IEEE, 2023.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3563–3579, 2025.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137, 2015.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

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

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10955–10965. IEEE, 2022.

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.

Yihong Luo, Tianyang Hu, Weijian Luo, Kenji Kawaguchi, and Jing Tang. Reward-instruct: A reward-centric approach to fast photo-realistic image generation. *Advances in Neural Information Processing Systems*, 38:166708–166733, 2026.

Zhengyao Lv, Chenyang Si, Tianlin Pan, Zhaoxi Chen, Kwan-Yee K Wong, Yu Qiao, and Ziwei Liu. Dual-expert consistency model for efficient and high-quality video generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14983–14993. IEEE, 2025.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14297–14306. IEEE, 2023.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

OpenAI. GPT-5.5 Instant System Card. `https://openai.com/index/gpt-5-5-instant-system-card/`, May 2026.

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.

Ziqi Pang, Xin Xu, and Yu-Xiong Wang. Aligning generative denoising with discriminative objectives unleashes diffusion for visual perception. In *International Conference on Learning Representations*, volume 2025, pp. 101991–102013, 2025.

Pinelopi Papalampidi, Olivia Wiles, Ira Ktena, Aleksandar Shtedritski, Emanuele Bugliarello, Ivana Kajić, Isabela Albuquerque, and Aida Nematzadeh. Dynamic classifier-free diffusion guidance via online feedback. In *International Conference on Learning Representations*, volume 2026, pp. 127362–127378, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Jielin Qiu, Yi Zhu, Xingjian Shi, Florian Wenzel, Zhiqiang Tang, Ding Zhao, Bo Li, and Mu Li. Benchmarking robustness of multimodal image-text models under distribution shift. *arXiv preprint arXiv:2212.08044*, 2022.

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.

Shachar Rosenman, Vasudev Lal, and Phillip Howard. Neuroprompts: An adaptive framework to optimize prompts for text-to-image generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 159–167, 2024.

Seyedmorteza Sadat, Otmar Hilliges, and Romann M Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2024.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Ketan Suhaas Saichandran, Xavier Thomas, Prakhar Kaushik, and Deepti Ghadiyaram. Progressive prompt detailing for improved alignment in text-to-image generative models. *arXiv preprint arXiv:2503.17794*, 2025.

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.

Rong Tang, Lizhen Lin, and Yun Yang. Conditional diffusion models are minimax-optimal and manifold-adaptive for conditional distribution estimation. *arXiv preprint arXiv:2409.20124*, 2024.

Chenxi Xie, Minghan Li, Shuai Li, Yuhui Wu, Qiaosi Yi, and Lei Zhang. Dnaedit: Direct noise alignment for text-guided rectified flow editing. *Advances in Neural Information Processing Systems*, 38:124456–124474, 2026.

Chejian Xu, Jiawei Zhang, Zhaorun Chen, Chulin Xie, Mintong Kang, Yujin Potter, Zhun Wang, Zhuowen Yuan, Alexander Xiong, Zidi Xiong, et al. Mmdt: Decoding the trustworthiness and safety of multimodal foundation models. In *International Conference on Learning Representations*, volume 2025, pp. 4069–4165, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Yi Zhang, Yun Tang, Wenjie Ruan, Xiaowei Huang, Siddartha Khastgir, Paul Jennings, and Xingyu Zhao. Protip: Probabilistic robustness verification on text-to-image diffusion models against stochastic perturbation. In *European Conference on Computer Vision*, pp. 455–472. Springer, 2024.

Xiangru Zhu, Penglei Sun, Yaoxian Song, Yanghua Xiao, Zhixu Li, Chengyu Wang, Jun Huang, Bei Yang, and Xiaoxiao Xu. Evaluating semantic variation in text-to-image synthesis: A causal perspective. In *International Conference on Learning Representations*, volume 2025, pp. 92089–92137, 2025.

# A Additional Review of Conditioned Diffusion Models

Diffusion models can be viewed as generative distribution models that learn to reverse a progressive noising process, with the reverse dynamics characterized by the score of the intermediate noisy distributions (Ho et al., 2020; Song et al., 2020). Conditional generation extends this formulation from modeling the distribution $p(x)$ to a family of conditional distributions $p(x \mid c)$. From a statistical perspective, this can be viewed as conditional distribution estimation: rather than fitting an independent generative model for every condition, a shared conditional score model $s_\theta(x_t, t, c)$ is learned from observations, each comprising a sample-prompt pair, and represents how the score field varies with the condition (Tang et al., 2024). Algorithmically, this extension requires conditioning the generative network on $c$, yielding, for example, a noise predictor $\epsilon_\theta(x_t, t, c)$ in DDPM parameterization (Dhariwal & Nichol, 2021). The score-based formulation additionally provides a useful connection between unconditional and conditional generation. By Bayes' rule, the conditional score can be decomposed as

$$\nabla_{x_t} \log p_t(x_t \mid c) = \nabla_{x_t} \log p_t(x_t) + \nabla_{x_t} \log p_t(c \mid x_t), \tag{11}$$

which allows an unconditional generative model to be steered toward a condition using the gradient of an auxiliary predictor. This principle underlies classifier guidance, introduced for high-fidelity class-conditional image generation (Song et al., 2020). As follows from (11), evaluating the conditional score $\nabla_{x_t} \log p_t(x_t \mid c)$ requires only the unconditional score $\nabla_{x_t} \log p_t(x_t)$ and the classifier score $\nabla_{x_t} \log p_t(c \mid x_t)$, which can be modeled using the gradient of an auxiliary model trained to classify noisy samples. However, classifier guidance defined in (11) does not fully achieve the desired outcome: generated samples do not always visually correspond to the specified classes. Dhariwal & Nichol (2021) report class consistency in only approximately 50% of cases. This issue can be addressed by strengthening the guidance signal, replacing (11) with

$$s_s(x_t, c) = \nabla_{x_t} \log p_t(x_t) + s \nabla_{x_t} \log p_t(c \mid x_t), \tag{12}$$

where increasing the parameter $s > 1$ trades diversity (mode coverage) for sample fidelity (Dhariwal & Nichol, 2021).

The need to train a classifier that operates at every noise level introduces additional complexity into the conditional generative modeling pipeline, especially when the condition $c$ is specified not by a fixed set of classes but, for example, by free-form text (Nichol et al., 2021). An alternative is to model the conditional score directly by conditioning the generative network itself on $c$. Importantly, this construction retains the ability to amplify the guidance signal analogously to (12):

$$s_s(x_t, c) = \nabla_{x_t} \log p_t(x_t \mid \varnothing) + s \left( \nabla_{x_t} \log p_t(x_t \mid c) - \nabla_{x_t} \log p_t(x_t \mid \varnothing) \right), \tag{13}$$

where the parameter $s$ retains its role as the guidance scale: $s = 1$ recovers the ordinary conditional prediction, whereas $s > 1$ amplifies the influence of the condition, as in (12). This construction is motivated by the fact that, for exact scores, the difference between the conditional and unconditional scores corresponds to the gradient of an implicit classifier, making classifier-free guidance (Ho & Salimans, 2022) (CFG), introduced in (13), a direct analogue of amplified classifier guidance (12). Empirically, CFG improves sample fidelity and text-image alignment (Nichol et al., 2021; Saharia et al., 2022) and has consequently become a standard inference component of modern T2I diffusion and flow-based systems (Podell et al., 2024; Chen et al., 2024; Esser et al., 2024). The guidance scale $s$, however, cannot be increased arbitrarily. Stronger guidance progressively concentrates the generated distribution and therefore reduces sample diversity (Ho & Salimans, 2022). Moreover, large guidance scales can degrade visual fidelity, producing oversaturated, unnatural images and other visual artifacts (Saharia et al., 2022; Sadat et al., 2024). Thus, practical T2I systems operate in a regime that balances the increased influence of the textual condition against the degradation induced by excessive guidance. The widespread practical adoption of CFG, together with the limitation described above, has motivated two complementary lines of research aimed at improving the method. One direction seeks to further enhance generation quality through adaptive scheduling of the CFG scale (Kynkäänniemi et al., 2024; Papalampidi et al., 2026). A complementary direction focuses on modifying the geometry of the CFG update to mitigate off-manifold deviations (Chung et al., 2025; Sadat et al., 2024).

Guidance-aware distillation offers an alternative implementation of the same principle. The student model is trained to reproduce guided teacher outputs while receiving $s$ together with the textual

condition, allowing inference to replace the explicit combination of conditional and unconditional branches with a single scale-conditioned network evaluation (Meng et al., 2023). Equation (2) summarizes the resulting parameterization.

Finally we note that for conditioning over a finite set of classes, the class index is typically mapped to a learned embedding, which is injected into the generative network through adaptive normalization or attention-based conditioning layers (Dhariwal & Nichol, 2021; Rombach et al., 2022; Peebles & Xie, 2023). For text conditioning, the input prompt is first mapped by a text encoder to a sequence of token embeddings, which is then supplied to the generative network through attention-based conditioning layers (Nichol et al., 2021; Rombach et al., 2022; Saharia et al., 2022).

# B Conditioning Sensitivity Measures: Derivation and Evaluation Protocol

**Observable decomposition.** At a fixed sampler state $x_t$, we observe only the network output $F_\theta(x_t, c)$. For any separation $c = (c_s, c_n)$, it can be written as

$$F_\theta(x_t, c) = F_\theta(x_t, c_s) + \delta_\theta(x_t, c_s, c_n). \tag{14}$$

We note that this identity is always attainable - in the degenerate case, one may take $F_\theta(x_t, c_s) = 0$ and assign the complete output to $\delta_\theta$. Empirically, however, generations frequently follow the prescribed semantics, indicating that the semantic component $F_\theta(x_t, c_s)$ governs the dynamics and typically has substantially greater amplitude, while $\delta_\theta$ acts as a perturbation.

Let $\mathcal{C}$ denote a random variable supported on prompts with the same semantic component $c_s$ and differing only in $c_n$. We formalize the non-semantic term as a centered fluctuation:

$$\mathbb{E}_{c' \sim \mathcal{C}} \delta_\theta(x_t, c_s, c'_n) = 0.$$

This assumption is natural for variation and entails no loss of generality within the free decomposition in Equation (14): any nonzero mean of $\delta_\theta$ can be absorbed into the semantic component $F_\theta(x_t, c_s)$. Taking expectation over $c' \sim \mathcal{C}$ therefore identifies

$$F_\theta(x_t, c_s) = \underbrace{\mathbb{E}_{c' \sim \mathcal{C}} F_\theta(x_t, c')}_{\overline{F}_{\theta,t}(\mathcal{C})}, \qquad \delta_\theta(x_t, c_s, c'_n) = \underbrace{F_\theta(x_t, c') - \overline{F}_{\theta,t}(\mathcal{C})}_{\Delta_{\theta,t}(c')}.$$

**Extracting the perturbation scale.** To separate direction from magnitude, we multiply and divide the centered residual so that its expected norm matches that of the semantic component. This gives the observable counterpart of Equation (3):

$$F_\theta(x_t, c') = \underbrace{\overline{F}_{\theta,t}(\mathcal{C})}_{F_\theta(x_t, c_s)} + \underbrace{\frac{\mathbb{E}_{\tilde{c} \sim \mathcal{C}} \|\Delta_{\theta,t}(\tilde{c})\|}{\|\overline{F}_{\theta,t}(\mathcal{C})\|}}_{\varepsilon_{c,t}} \underbrace{\left( \Delta_{\theta,t}(c') \frac{\|\overline{F}_{\theta,t}(\mathcal{C})\|}{\mathbb{E}_{\tilde{c} \sim \mathcal{C}} \|\Delta_{\theta,t}(\tilde{c})\|} \right)}_{\widetilde{f}_{\theta,t}(c')}.$$

Consequently, $\varepsilon_{c,t}$ is dimensionless and measures non-semantic output variation relative to the semantic signal at timestep $t$.

For a collection $\mathcal{D} = \{\mathcal{C}_i\}_{i=1}^{100}$, we obtain a model-level score by averaging the local ratio uniformly across prompt groups $\mathcal{C}$ and sampling steps $\mathcal{T}$. To account for propagation through the sampling trajectory, we additionally normalize the timestep-importance scores from Section 5.1 as $\pi_t = I_t / \sum_{u \in \mathcal{T}} I_u$ and average under the resulting distribution:

$$\varepsilon = \mathbb{E}_{\mathcal{C}_i \sim \mathcal{D}} \mathbb{E}_{t \sim \mathrm{Uniform}(\mathcal{T})} [\varepsilon_{c_i,t}],$$
$$\varepsilon^* = \mathbb{E}_{\mathcal{C}_i \sim \mathcal{D}} \mathbb{E}_{t \sim \pi} [\varepsilon_{c_i,t}].$$

The former measures average local sensitivity, whereas the latter emphasizes deviations occurring at steps with greater influence on the final image, making the metric trajectory-aware.

**Experimental instantiation.** We use the seven semantically equivalent formulations constructed for each of the 100 PartiPrompts anchors in Section 4.1. For ordinary single-prompt conditioning, $\mathcal{C}_i$ is uniform over two formulations. For methods based on LA, each realization is instead a three-prompt

tuple. We let $\mathcal{C}_i$ be uniform over two such tuples and apply the same construction to the averaged conditional outputs produced by these methods. The two triples are disjoint, preventing shared prompts from mechanically increasing measured agreement. All other sampler settings are held fixed across the compared methods.

For the ablation over $N$, we expand each of the 100 formulation groups introduced in Section 4.1 from seven to 14 semantically equivalent formulations (the anchor and 13 transformations). This provides two disjoint seven-prompt tuples for evaluating $\mathrm{LA}_7$ under the same $\varepsilon$ measurement protocol, while retaining the corresponding disjoint subsets for smaller values of $N$. The additional formulations are generated using a rewritten few-shot instruction, reproduced in Listing 2 of Appendix G.

## C SVRG Estimation for Latent Averaging

SVRG (Johnson & Zhang, 2013) and SAGA (Defazio et al., 2014) were originally developed for finite-sum gradient optimization. Consider

$$\min_{\theta} f(\theta), \qquad f(\theta) = \frac{1}{N}\sum_{i=1}^{N} f_i(\theta).$$

The gradient $\nabla f(\theta) = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\theta)$ defines the vector field followed by gradient-based optimization, while the finite-sum structure permits individual components to be evaluated without the costly computation of the complete field at every iteration. LA has the same algebraic structure: each condition contributes one separately evaluable component of the averaged generative field. The analogy is especially direct for flow-matching models, whose sampling procedure explicitly integrates a learned velocity field. This correspondence motivates applying finite-sum variance-reduction estimators to conditional generation.

Let $\mathcal{C} = \{c^i\}_{i=1}^N$ denote the set of semantically equivalent conditions. Their latent average at sampler state $x_t$ is

$$\overline{F}_\theta(x_t, \mathcal{C}) = \frac{1}{N}\sum_{i=1}^{N} F_\theta(x_t, c^i).$$

A direct evaluation requires $N$ network calls at every step. Full $\mathrm{SVRG}_N$ instead maintains a periodically refreshed snapshot $\widetilde{x}$. At each refresh, it evaluates $F_\theta(\widetilde{x}, c^i)$ for all conditions and stores their average. Between refreshes, one condition index is sampled and a single new network evaluation transfers the change in the generative dynamics from the snapshot to the current state.

**Algorithm 3** $\mathrm{SVRG}_N$ (Full SVRG for Latent Averaging)

**Input:** Condition $c^1$, refresh period $m$, horizon $T$, initial state $x_{t_0}$
Construct semantic preserving perturbations $c^2, \ldots, c^N$
**for** $r = 0, m, 2m, \ldots < T$ **do**
  $\widetilde{x} = x_{t_r}$
  $\widetilde{g}^i \leftarrow F_\theta(\widetilde{x}, c^i), \quad i = 1, \ldots, N$
  $\bar{g} = N^{-1}\sum_{i=1}^{N} \widetilde{g}^i$
Update $x$ using $\bar{g}$
  **for** $k = r+1, \ldots, \min(r+m-1, T-1)$ **do**
    Sample $i_k \sim \mathrm{U}\{1, \ldots, N\}$
    $\hat{g}_k \leftarrow F_\theta(x_{t_k}, c^{i_k}) - \widetilde{g}^{i_k} + \bar{g}$
    Update $x$ using $\hat{g}_k$

The key intuition is that condition-specific trajectories evolve coherently across nearby sampling states. The latent average at $\widetilde{x}$ anchors all $N$ conditions, while the difference $F_\theta(x_{t_k}, c^{i_k}) - F_\theta(\widetilde{x}, c^{i_k})$ estimates how this shared dynamics has changed since the snapshot. Thus, only one conditional network call is needed between refreshes. The early-step $\mathrm{SVRG}_N^K$ method in Algorithm 1 is the special case with a single snapshot at $k = 0$, no periodic refresh, and truncation after $k = 4$.

**Unbiasedness.** The following statement holds for vector-valued network outputs and therefore applies directly to noise, score, denoised-sample, and velocity parameterizations.

**Lemma 1.** Fix a current state $x_t$, a snapshot $\widetilde{x}$, and the cached full mean $\overline{F}_\theta(\widetilde{x}, \mathcal{C})$. If $i$ is sampled uniformly from $\{1, \dots, N\}$, then the SVRG estimator

$$\widehat{F}_\theta(x_t; \widetilde{x}, c^i) = F_\theta(x_t, c^i) - F_\theta(\widetilde{x}, c^i) + \overline{F}_\theta(\widetilde{x}, \mathcal{C})$$

is unbiased for the latent average: $\mathbb{E}_i[\widehat{F}_\theta(x_t; \widetilde{x}, c^i)] = \overline{F}_\theta(x_t, \mathcal{C})$.

**Proof.** By uniform sampling and linearity of expectation,

$$\begin{aligned}\mathbb{E}_i[\widehat{F}_\theta(x_t; \widetilde{x}, c^i)] &= \frac{1}{N}\sum_{i=1}^{N}\big(F_\theta(x_t, c^i) - F_\theta(\widetilde{x}, c^i) + \overline{F}_\theta(\widetilde{x}, \mathcal{C})\big) \\ &= \overline{F}_\theta(x_t, \mathcal{C}) - \overline{F}_\theta(\widetilde{x}, \mathcal{C}) + \overline{F}_\theta(\widetilde{x}, \mathcal{C}) \\ &= \overline{F}_\theta(x_t, \mathcal{C}).\end{aligned}$$

□

## D SAGA$_N^{*K}$ FOR LATENT AVERAGING

**Naive SAGA$_N$.** For consistency with Algorithm 2, let $f_k^i = F_\theta(x_{t_k}, c^i)$. A direct adaptation of SAGA caches the most recently evaluated output $g^i$ for each condition and maintains $\bar{g} = N^{-1}\sum_i g^i$. Algorithm 4 gives the complete procedure.

**Algorithm 4** SAGA$_N$ (Naive SAGA for Latent Averaging)

**Input:** Condition $c^1$, horizon $T$, initial state $x_{t_0}$
Construct semantics-preserving variants $c^2, \dots, c^N$
$g^i \leftarrow F_\theta(x_{t_0}, c^i), \quad i = 1, \dots, N$
$\bar{g} = N^{-1}\sum_{i=1}^{N} g^i$
Update $x$ using $\bar{g}$
**for** $k = 1, \dots, T-1$ **do**
  Sample $i_k \sim \mathrm{U}\{1, \dots, N\}$
  $f_k^{i_k} \leftarrow F_\theta(x_{t_k}, c^{i_k})$
  $\hat{g}_k^{\mathrm{SAGA}} = \bar{g} - g^{i_k} + f_k^{i_k}$
  Update $x$ using $\hat{g}_k^{\mathrm{SAGA}}$
  $\bar{g} = \bar{g} + N^{-1}(f_k^{i_k} - g^{i_k})$
  $g^{i_k} = f_k^{i_k}$

At each non-initial step, the conditional estimate is

$$\hat{g}_k^{\mathrm{SAGA}} = \bar{g} - g^{i_k} + f_k^{i_k}. \tag{15}$$

Although Equation (15) is unbiased conditional on the cache, its entries correspond to different timesteps, and the evolution of the full mean is extrapolated from a single sampled component. This stale, one-sample extrapolation can produce high-variance updates.

For example, suppose that at the current step $k$, entries 1, 2, and 3 were last updated at steps $k-1$, $k-2$, and $k-3$, respectively, and index $i_k = 1$ is sampled again. Equation (15) inserts the *one-step* change $f_k^1 - f_{k-1}^1$ into a mean whose other components remain *two* and *three* steps stale.

$$\begin{aligned}\hat{g}_k^{\mathrm{SAGA}} &= \bar{g} + f_k^1 - f_{k-1}^1 \\ &= \frac{1}{3}\Big[f_k^1 + \big(f_{k-2}^2 + f_k^1 - f_{k-1}^1\big) \\ &\quad + \big(f_{k-3}^3 + f_k^1 - f_{k-1}^1\big)\Big].\end{aligned}$$

The red indices expose the mismatch: the one-step increment from condition 1 is used to transport components whose cached states are two and three steps old. Thus, a correction calibrated to one-step dynamics is combined with entries that require substantially longer temporal transport. In practice, the resulting inconsistency destabilizes sampling, as illustrated in Figure 8.

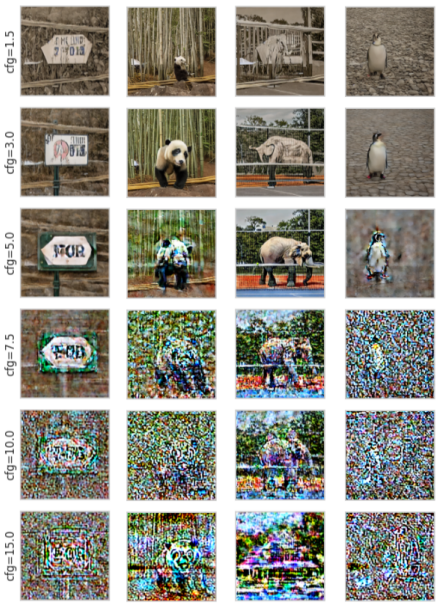



Figure 8: Examples of unstable generations produced by naive SAGA$_N$.

**Removing temporal dynamics from the cache.** To eliminate this failure mode, we fix a reference condition $c^{i^*}$ and store same-timestep residuals

$$g^i = f^i_{\ell_i} - f^{i^*}_{\ell_i},$$

where $\ell_i$ is the last update step of entry $i$. Both terms in every cache entry are evaluated at the same state, so temporal evolution is excluded from the SAGA memory.

At the current step, the corresponding estimator is

$$\hat{g}_k = \bar{g} - g^{i_k} + \left(f^{i_k}_k - f^{i^*}_k\right) + f^{i^*}_k \tag{16}$$

$$= \bar{g} - g^{i_k} + f^{i_k}_k. \tag{17}$$

The temporal dynamics now enter only through the term $f^{i^*}_k$ in Equation (16), rather than through stale cache states.

Returning to the three-condition example above, define the same-timestep Using a dedicated conditional reference would require two conditional network evaluations at every step. Classical CFG, however, already evaluates $f^{\varnothing}_k = F_\theta(x_{t_k}, \varnothing)$. We therefore set the reference branch to the unconditional prediction and store

$$g^i = f^i_{\ell_i} - f^{\varnothing}_{\ell_i}.$$

The reference terms cancel exactly as in Equation (17), yielding the SAGA$^{*K}_N$ estimator used in Algorithm 2. After computing $f^{\varnothing}_k$ and $f^{i_k}_k$, the running mean and sampled cache entry are updated as

$$\bar{g} = \bar{g} + \frac{1}{N}\left(f^{i_k}_k - f^{\varnothing}_k - g^{i_k}\right), \qquad g^{i_k} = f^{i_k}_k - f^{\varnothing}_k.$$

Consequently, SAGA$^{*K}_N$ matches the per-step cost of baseline CFG after initialization and incurs only $N-1$ additional network evaluations at the first step. It is applied only for $k < K$; for $k \geq K$, the sampler uses the original condition $c^1$.

**Unbiasedness.** We next establish that the SAGA$^{*K}_N$ estimator in Algorithm 2 remains unbiased.

**Lemma 2.** Fix $x_{t_k}$ and a cache $\{g^i\}_{i=1}^N$ with $\bar{g} = N^{-1}\sum_i g^i$. If $i_k \sim \mathrm{U}\{1, \ldots, N\}$, then the SAGA$^{*K}_N$ estimator from Algorithm 2,

$$\hat{g}_k = \bar{g} - g^{i_k} + f^{i_k}_k,$$

is unbiased for the latent average in Equation (7).

**Proof.** Conditioning on the cache and applying uniform sampling yields

$$\begin{aligned}
\mathbb{E}_{i_k}\left[\hat{g}_k \mid x_{t_k}, \{g^i\}_{i=1}^N\right] &= \frac{1}{N}\sum_{i=1}^N \left(\bar{g} - g^i + f^i_k\right) \\
&= \bar{g} - \frac{1}{N}\sum_{i=1}^N g^i + \frac{1}{N}\sum_{i=1}^N f^i_k \\
&= \overline{F}_\theta(x_{t_k}, \mathcal{C}).
\end{aligned}$$

The final equality follows from the invariant $\bar{g} = N^{-1}\sum_{i=1}^N g^i$, maintained by Lines 12-13 of SAGA$^{*K}_N$ (Algorithm 2). □

Notably, the proof is independent of when individual entries were last updated. SAGA$^{*K}_N$ therefore preserves unbiasedness while removing shared temporal drift from the cached quantities.

# E PROMPT-PERTURBATION DATASET AND TIMESTEP PROTOCOL

## E.1 PROMPT-VARIATION GENERATION

We construct a compact evaluation set of 100 source prompts, each accompanied by six semantics-preserving variations. The variations are generated with GPT-5.5 Instant (OpenAI, 2026) using a fixed few-shot instruction that requests diverse lexical and syntactic realizations while preserving the visual scene. The generated groups are frozen before image generation and reused across all experiments in Sections 4.1 and 5.1. The complete generation instruction is reproduced in Listing 1 of Appendix G.

## E.2 TIMESTEP-IMPORTANCE

For each of the 100 prompt groups, we randomly sample four ordered prompt pairs. Within each pair, the first prompt is used for the reference generation and the second replaces it only at the evaluated step $k^\star$. The two prompts in a pair are always distinct, and no ordered pair is repeated within a group. For every pair, we evaluate all $k^\star \in \{0, \ldots, 29\}$ using the same initial noise and otherwise identical sampling configuration. The reported statistic is the squared Euclidean distance between the DINOv2 representations of the reference and single-step-substitution images, aggregated over the resulting $100 \times 4$ ordered pairs.

# F CORBEN: CONSTRUCTION AND EVALUATION PROTOCOL

## F.1 ADDITIONAL CORBEN RESULTS

Table 3 reports recall at the stricter threshold $\tau = 0.7$. SD1.5.LA$_3$ improves OR$_{0.7}$ in every family and reduces the SD1.5 deficit throughout the sensitivity-focused subset.

Table 3: CoRBen results for OR$_{0.7}$. Columns labeled LA$_3$ correspond to SD1.5.LA$_3$.

| | OR$_{0.7}$ ↑ | | OR$^*_{0.7}$ ↑ | | |
|---|---|---|---|---|---|
| Perturbation | SD1.5 | LA$_3$ | Original | SD1.5 | LA$_3$ |
| Alt. captions | 56.11 | **60.67** | 62.43 | 52.06 | **59.12** |
| Character | 58.59 | **61.90** | 61.68 | 52.84 | **59.53** |
| FSP | 61.96 | **62.65** | 58.37 | 55.57 | **57.17** |
| Sentence | 62.37 | **62.90** | 56.97 | 52.98 | **55.44** |
| Word | 61.52 | **62.33** | 58.87 | 54.88 | **57.11** |

Table 4 reports the retained populations for both thresholds. Prompt-bound records are counted separately for each perturbation family; the overall prompt count is the number of unique source anchors represented after filtering.

Table 4: OR$^*_\tau$ selection populations. Entries are retained/initial. "Records" are detector-supported anchor-concept records; "pairs" are perturbation-level concept-prompt evaluation opportunities.

| | $\tau = 0.5$ | | | $\tau = 0.7$ | | |
|---|---|---|---|---|---|---|
| Perturbation | Records | Prompts | Pairs | Records | Prompts | Pairs |
| Alt. captions | 4,223/12,441 | 2,924/5,000 | 16,904/49,790 | 3,540/6,059 | 2,645/5,000 | 14,171/24,253 |
| Character | 3,600/12,441 | 2,695/5,000 | 14,400/49,764 | 3,296/6,059 | 2,503/5,000 | 13,184/24,236 |
| FSP | 3,214/12,441 | 2,441/5,000 | 27,763/107,071 | 3,819/6,059 | 2,759/5,000 | 32,913/52,130 |
| Sentence | 1,832/12,247 | 1,507/4,921 | 7,060/44,682 | 2,368/5,976 | 1,909/4,921 | 9,092/21,894 |
| Word | 3,056/12,441 | 2,284/5,000 | 18,336/74,646 | 3,341/6,059 | 2,496/5,000 | 20,046/36,354 |
| Overall | 15,925/62,011 | 4,044/5,000 | 84,463/325,953 | 16,364/30,212 | 3,517/5,000 | 89,406/158,867 |

## F.2 SOURCE DATA AND SEMANTIC ANCHORS

CoRBen is constructed from the 5,000-image test partition of the standard MS COCO Karpathy split (Lin et al., 2014; Karpathy & Fei-Fei, 2015). Each image is associated with multiple independently

Table 5: CoRBen perturbation populations. "Raw" denotes the number of initially available or requested non-anchor texts; "evaluated" denotes variants retained after the applicable normalization and deduplication rules.

| Group | Raw variants | Evaluated variants |
|---|---|---|
| Character level | 20,000 | 20,000 |
| Word level | 30,000 | 30,000 |
| Sentence level | 25,000 | 17,827 |
| Recaptions | 20,010 | 20,010 |
| FSP | 45,000 | 42,967 |
| Total | 140,010 | 130,804 |

written human captions. In our annotation file, 4,990 images have five captions and 10 images have six, giving 25,010 captions in total. These captions are original COCO annotations, not outputs of a recaptioning model. We nevertheless retain the term *recaptions* for consistency with the naming of our experimental groups.

For every source image $a$, we designate `caption[0]` as its unique semantic anchor. All synthetic perturbations, including FSP, are generated from this caption only. The remaining human captions form the recaptions group. The concepts extracted from `caption[0]` are frozen and inherited by all of its character-, word-, sentence-, recaption-, and FSP variants.

### F.3 Legacy perturbation families

CoRBen retains a controlled subset of the text transformations introduced by Qiu et al. (2022). The retained character-level transformations are random character insertion, substitution, swapping, and deletion, each at severity 1. The retained word-level transformations are word insertion, word swapping, and punctuation insertion at severity 1, together with word deletion at severities 1, 4, and 7. The sentence-level group contains formalization, casualization, conversion to passive voice, conversion to active voice, and back-translation. Importantly, this selection is intended to limit semantic drift but does not guarantee semantic equivalence, especially for high-severity word deletion.

As the original generated perturbation files were not included in the public repository, we regenerated the retained transformations following the protocol of Qiu et al. (2022).

Before evaluation, texts are normalized using Unicode NFKC normalization, whitespace normalization, and case folding. Within each anchor and perturbation family, unchanged strings and duplicate variants are removed. Duplicate texts across different perturbation families are allowed.

Table 5 summarizes the resulting populations. Counts correspond to non-anchor prompts. The sentence annotations contain 25,000 raw outputs before normalization; 17,827 distinct memberships remain in the generation and evaluation plan.

### F.4 Fixed-Semantics Perturbations (FSP)

Legacy text corruptions remain useful for measuring general prompt sensitivity, but they only partially represent the variation encountered in practical T2I systems. Character corruption can often be repaired before generation, while word deletion, sentence transformation, and independently written COCO captions can modify object identity, color, count, or other visually relevant details, preventing us from treating these variants as exact descriptions of the same scene or expecting identical generations from them. We therefore introduce Fixed-Semantics Perturbations (FSP), which are explicitly constrained to change linguistic realization while preserving the intended visual scene.

**Generation model.** We generate FSP with Qwen3-30B-A3B (Yang et al., 2025), served locally through vLLM. The model is used in non-thinking mode. We apply JSON-schema-guided decoding with temperature $0.3$, top-$p = 0.9$, repetition penalty $1.05$, maximum output length 2,048 tokens, and random seed 42. Generation is retried at most five times when the output cannot be parsed or does not satisfy the required schema.

Table 6: Retained FSP variants after unchanged-output and within-anchor duplicate filtering.

| Transformation | Count | Transformation | Count |
|---|---|---|---|
| Lexical | 4,977 | Syntactic | 4,996 |
| Structural | 5,000 | Compressed | 4,997 |
| Expanded | 4,973 | Style-shifted | 4,991 |
| Determiner shift | 4,390 | Attribute order | 3,764 |
| Micro-paraphrase | 4,879 | Total | 42,967 |

For every anchor, the model must return the original prompt and nine transformations:

1. `lexical`: synonym or near-synonym substitution;
2. `syntactic`: changes in grammar, word order, or voice;
3. `structural`: changes in sentence architecture;
4. `compressed`: removal of non-essential linguistic material;
5. `expanded`: addition of neutral grammatical material;
6. `style_shifted`: change of linguistic register only;
7. `determiner_shift`: changes to articles and determiners;
8. `attribute_order`: reordering of attributes and modifiers;
9. `micro_paraphrase`: small local equivalent reformulations.

The instruction explicitly requires preservation of all main objects, actions, relations, object counts, core attributes such as color, size, and position, and any stated scene context. It prohibits adding or removing meaningful entities, changing actions or counts, and introducing photographic or artistic styles. Transformation-specific examples are included in the instruction.

**Validation.** The returned `base` field is overwritten with the exact source caption, irrespective of the model output. Candidate variants are compared after Unicode NFKC normalization, whitespace normalization, and case folding. A variant is discarded if it is equal to the anchor or duplicates another FSP variant of the same anchor. Deduplication is restricted to the nine FSP candidates of one anchor; an FSP variant may coincide with a sentence-level or other legacy variant.

Of 45,000 requested FSP transformations, 42,967 are retained and 2,033 are removed as unchanged or within-anchor duplicates. The per-transformation counts are reported in Table 6. We additionally inspect a fixed random sample of 100 anchors together with all their retained variants as a qualitative audit. This audit is used to examine common failure modes but is not treated as a formal guarantee of semantic equivalence for every output.

**Generation instruction.** The complete instruction supplied to Qwen3-30B-A3B is reproduced in Listing 3 of Appendix G.

### F.5 Prompt-derived atomic concepts

**Limitations of the original references.** The object-evaluation protocol of Qiu et al. (2022) constructs the GLIP query from the ground-truth COCO instance categories associated with the source image. These categories describe the complete image rather than the selected conditioning caption. Consequently, *a generated image may be rewarded or penalized for an object that the prompt never requests*.

Our audit additionally revealed a systematic category-name error in the released `cat_per_img.txt` artifact. Sparse COCO category identifiers were used as positions in a dense 80-class name list, shifting category names after gaps in the COCO ID space. For example, *stop sign* was represented as *parking meter*, while *bottle* was represented as *knife*. Correcting this mapping would not address the more fundamental mismatch between image annotations and prompt semantics. We therefore discard all previous GLIP references, detections, and derived object scores.

**Extraction model and instruction.** We extract a new concept contract independently for each of the 5,000 clean anchors using Qwen3-30B-A3B. We use vLLM with non-thinking mode and JSON-schema-guided decoding. Extraction is deterministic: temperature 0, top-$p = 1$, repetition penalty 1, maximum output length 768 tokens, and random seed 42.

The extraction instruction requests only concrete and visually groundable entities explicitly mentioned in the prompt. It includes people, animals, vehicles, clothing, tools, furniture, food, and localizable scene regions such as roads, beaches, fields, water, and sky. It excludes attributes, actions, relations, artistic styles, and entities inferred from background knowledge.

Every extracted record contains:

- `surface`: the noun phrase appearing in the prompt;
- `canonical`: a normalized singular category;
- `glip_query`: a short grounding query;
- `kind`: either `countable_entity` or `scene_region`;
- `quantity.value`: the caption-derived required quantity;
- `quantity.exact`: whether that quantity is stated exactly.

Repeated mentions of the same canonical category are merged into one stored record with a corresponding quantity. Explicit numerals are preserved. Unspecified plurals receive the smallest quantity justified by the caption, normally two, and are marked as inexact. Scene regions are represented once and are evaluated by presence. Meaningful compound categories such as *traffic light*, *food truck*, and *tennis racket* are preserved, while attributes are removed from the GLIP query.

The initial automatic extraction produced structurally valid outputs for all 5,000 anchors. Deterministic validation identified 12 anchors with duplicate canonical forms, duplicate GLIP queries, or inconsistent scene-region quantities. These cases were manually adjudicated and revalidated. The final release contains 13,717 atomic concept records: 10,319 countable-entity records and 3,398 scene-region records.

The complete extraction instruction is reproduced in Listing 4 of Appendix G.

### F.6 GLIP EVALUATION AND DETECTOR SUPPORT

We use GLIP-L (Li et al., 2022) with confidence thresholds $\tau \in \{0.5, 0.7\}$. For every generated image, GLIP receives the ordered `glip_query` strings extracted from the corresponding clean anchor. It never receives the old COCO-category query or any previous GLIP prediction.

Open-vocabulary detectors cannot reliably localize every valid noun phrase. To avoid treating detector blind spots as generation failures, we first apply the same concept queries to the corresponding real COCO image. A concept record is eligible at threshold $\tau$ only if GLIP-L detects at least one box for it in the real image above the same threshold. This is a detector-capability gate, not an additional source-image target: the required quantity is still obtained exclusively from the prompt. The gate tests presence only, so one supported real-image detection retains all quantity slots requested by the anchor.

The concepts of the clean anchor are reused for all its perturbations. In particular, recaptions do not receive separately extracted concepts.

### F.7 OBJECT RECALL NOTATION

This section makes the compact definition in Equation (6) precise. Index $c$ denotes an evaluated prompt-generated-image pair. Although the concepts are inherited from its clean anchor, we repeat the corresponding evaluation targets for every generated prompt $c$.

The dataset stores one record for each canonical concept together with its required quantity $q$. For evaluation, a record with quantity $q$ is expanded into $q$ concept-instance slots:

$$\mathcal{S}_c = \{s_c^1, \ldots, s_c^{N_c}\},$$

where $N_c$ is the total caption-derived quantity across the anchor's concept records. A scene region contributes one slot.

Let $\mathcal{S}_c^\tau \subseteq \mathcal{S}_c$ be the slots whose concept record passes the real-image GLIP support gate at threshold $\tau$. In Equation (6), the notation $\{s_c^i\}_{c,i}$ refers implicitly to these evaluated, support-eligible slots.

For a canonical concept with required quantity $q$, let

$$r_c^{(1)} \geq r_c^{(2)} \geq \cdots$$

be the confidence scores of its GLIP boxes in the generated image, sorted in descending order. We assign the $k$-th required slot the confidence

$$p_c^k = \begin{cases} r_c^{(k)}, & \text{if a } k\text{-th detection exists,} \\ 0, & \text{otherwise.} \end{cases}$$

Consequently,

$$\sum_{k=1}^{q} \mathbb{I}\big[p_c^k > \tau\big] = \min(D_c^\tau, q)\,,$$

where $D_c^\tau$ is the number of GLIP boxes for the category above threshold. Thus, the compact OR definition in Equation (6) is equivalent to the quantity-capped implementation: duplicate detections beyond the requested quantity receive no additional credit. OR is micro-averaged over all support-eligible concept-instance slots.

### F.8 FID AND CLIP-FID PROTOCOL

In the protocol of Qiu et al. (2022), FID was computed using all perturbations available for each source prompt. This design could introduce an additional effect unrelated to generation quality, because multiple correlated variants of the same anchor entered the distributional estimate. To obtain a cleaner and more balanced comparison, we instead sample exactly one perturbation per source anchor within each family.

In more detail, for every perturbation family and source anchor, we select exactly one condition using deterministic random sampling with seed 42. The complete prompt sets are frozen, and the same selection is used for SD1.5 and SD1.5.LA$_3$ when computing both FID and CLIP-FID. Every family-generation-method pair therefore contains 5,000 generated images, except for sentence-level perturbations, where 4,921 anchors admit a valid three-prompt LA set. Each pair is compared with the same number of real COCO images from the corresponding source anchors. FID is computed using Inception-v3 features, while CLIP-FID uses CLIP ViT-B/32 features.

All generations use Stable Diffusion 1.5, 30 denoising steps, resolution $512 \times 512$, CFG scale 7.5, and seed 42. The same source-image-dependent initial latent is shared across perturbations and generation methods.

# G PROMPT INSTRUCTIONS

## G.1 SEVEN-FORMULATION PROMPT VARIATIONS

Listing 1: Few-shot instruction used to generate seven-formulation semantics-preserving prompt groups.

```
TASK: PROMPT TRANSFORMATION (FEW-SHOT INSTRUCTION)

You are given a base prompt describing a simple visual scene.

Your task is to generate multiple valid variations of this prompt according to transformation
     rules.

----------------------------------------
OUTPUT FORMAT (STRICT TSV)

Return ONLY a TSV table with columns:

prompt_id group_id type_id prompt

Where:
- prompt_id = global incremental id starting from 0
- group_id = id grouping all variations derived from the same Base prompt
- type_id:
   0 = Base (original prompt, unchanged)
   1 = Lexical
   2 = Syntactic
   3 = Structural
   4 = Compressed
   5 = Expanded
   6 = Style-shifted

Each Base prompt MUST generate a full group of transformations.

----------------------------------------
CORE TRANSFORMATION RULES

You MUST preserve:
- main object(s)
- key action
- relationships between objects
- number of objects
- core attributes (color, size, position)
- scene context if explicitly present

You MUST NOT:
- add new meaningful entities
- remove key entities
- change action into a different action
- change scene meaning
- change number of objects
- introduce artistic/image styles (e.g. anime, watercolor, cinematic)

----------------------------------------
TRANSFORMATION TYPES

1. LEXICAL
- Only synonym / near-synonym substitution
- No structural changes
Example:
cat sitting on railing →feline sitting on banister

2. SYNTACTIC
- Change grammar / word order / voice
- Same words mostly preserved
Example:
cat sitting on railing →on railing sits a cat

3. STRUCTURAL
- Change sentence architecture (phrase ↔sentence, clause restructuring)
Example:
cat on railing →there is a cat positioned on railing

4. COMPRESSED
- Remove non-essential words
- Keep only core meaning
Example:
a cat sitting on a stair railing →cat on stair railing
```

```
5. EXPANDED
- Add neutral modifiers (manner, soft adverbs, grammar glue)
- MUST NOT add new visual entities or style
Example:
cat sitting on railing →a cat calmly sitting on the railing

6. STYLE-SHIFTED
- Change linguistic style only (formal, colloquial, terse, narrative)
- MUST NOT add artistic style
Example:
cat sitting on railing →a cat is seated upon the railing

----------------------------------------
FEW-SHOT EXAMPLES

INPUT BASE PROMPTS:

Example 1:
a cat sitting on a stair railing

Example 2:
a red car parked on the street

----------------------------------------
EXPECTED OUTPUT (FORMAT EXAMPLE ONLY):

prompt_id group_id type_id prompt
0 0 0 a cat sitting on a stair railing
1 0 1 a feline sitting on a stair banister
2 0 2 on a stair railing sits a cat
3 0 3 there is a cat positioned on a stair railing
4 0 4 cat on stair railing
5 0 5 a cat calmly sitting on the stair railing
6 0 6 a cat is seated upon the stair railing
7 1 0 a red car parked on the street
8 1 1 a crimson automobile parked on roadway
9 1 2 on the street is parked a red car
10 1 3 there is a red car parked on the street
11 1 4 red car street parked
12 1 5 a red car neatly parked on the street
13 1 6 a red car is parked on the street

----------------------------------------
FINAL RULE
Input TSV table is @data/processed/samples.tsv.
Return ONLY the TSV into @data/processed. No explanations. No headings. No markdown.
```

### G.2 FOURTEEN-FORMULATION PROMPT VARIATIONS

Listing 2: Rewritten few-shot instruction used to expand each prompt group to 14 formulations for the ablation over $N$.

```
TASK: PROMPT TRANSFORMATION (FEW-SHOT INSTRUCTION)

You are given a base prompt describing a simple visual scene.
Your task is to generate multiple valid variations of this prompt according to transformation
     rules.

----------------------------------------

OUTPUT FORMAT (STRICT TSV)

Return ONLY a TSV table with columns:

prompt_id group_id type_id prompt

Where:
- prompt_id = global incremental id starting from 0
- group_id = id grouping all variations derived from the same Base prompt
- type_id:
   0 = Base (original prompt, unchanged)
   1 = Lexical
   2 = Syntactic
   3 = Structural
   4 = Compressed
   5 = Expanded
   6 = Style-shifted
   7 = Relative-clause
```

```
   8 = Nominalized
   9 = Focus/Cleft
   10 = Clause decomposition
   11 = Grammatical realization
   12 = Caption framing
   13 = Instructional framing
Each Base prompt MUST generate a full group of transformations.
----------------------------------------
CORE TRANSFORMATION RULES
You MUST preserve:
- main object(s)
- key action
- relationships between objects
- number of objects
- core attributes (color, size, position)
- scene context if explicitly present
You MUST NOT:
- add new meaningful entities
- remove key entities
- change action into a different action
- change scene meaning
- change number of objects
- introduce artistic/image styles (e.g. anime, watercolor, cinematic)
----------------------------------------
TRANSFORMATION TYPES
1. LEXICAL
  - Only synonym / near-synonym substitution
  - No structural changes
    Example:
    cat sitting on railing →feline sitting on banister
2. SYNTACTIC
  - Change grammar / word order / voice
  - Same words mostly preserved
    Example:
    cat sitting on railing →on railing sits a cat
3. STRUCTURAL
  - Change sentence architecture (phrase ↔sentence, clause restructuring)
    Example:
    cat on railing →there is a cat positioned on railing
4. COMPRESSED
  - Remove non-essential words
  - Keep only core meaning
    Example:
    a cat sitting on a stair railing →cat on stair railing
5. EXPANDED
  - Add neutral modifiers (manner, soft adverbs, grammar glue)
  - MUST NOT add new visual entities or style
    Example:
    cat sitting on railing →a cat calmly sitting on the railing
6. STYLE-SHIFTED
  - Change linguistic style only (formal, colloquial, terse, narrative)
  - MUST NOT add artistic style
    Example:
    cat sitting on railing →a cat is seated upon the railing
7. RELATIVE-CLAUSE
  - Express an action, attribute, or relationship using a relative clause
  - Preserve all entities, actions, relationships, and attributes
    Example:
    a panda eating bamboo →a panda that is eating bamboo
8. NOMINALIZED
  - Express an action or state as a noun phrase
  - Preserve the original visual meaning
  - MUST NOT introduce abstract visual content or new entities
    Example:
    a horse reading a book →the reading of a book by a horse
9. FOCUS/CLEFT
  - Use a cleft or focus construction to emphasize part of the description
  - Do not add or remove visual information
    Example:
    a panda eating bamboo →it is a panda that is eating bamboo
10. CLAUSE DECOMPOSITION
   - Split one compact description into multiple clauses or sentences
   - Preserve all relationships using explicit nouns or unambiguous references
   - MUST NOT add new entities
     Example:
     a panda eating bamboo →a panda is present. The panda is eating bamboo.
11. GRAMMATICAL REALIZATION
   - Change only the grammatical realization of the prompt
   - Convert a phrase into a complete finite clause when appropriate
   - Do not use synonym substitution or stylistic embellishment
     Example:
```

```
    a red car parked on the street →a red car is parked on the street
12. CAPTION FRAMING
   - Present the same scene as a neutral image caption or visual report
   - Use neutral framing words such as "the image depicts" or "the scene shows"
   - The framing words MUST NOT add visual entities, actions, attributes, or styles
    Example:
    a panda eating bamboo →the image depicts a panda eating bamboo
13. INSTRUCTIONAL FRAMING
   - Present the same scene as a concise image-generation instruction
   - Use neutral imperative framing such as "depict" or "show"
   - MUST NOT add camera instructions, artistic styles, lighting, composition, or new visual
       content
    Example:
    a panda eating bamboo →depict a panda eating bamboo
----------------------------------------
FINAL RULE
Return the original base prompt as type 0 followed by exactly 13 transformed prompts. No
    explanations. No headings. No markdown.
----------------------------------------
FEW-SHOT EXAMPLES
INPUT:
a cat sitting on a stair railing
a red car parked on the street
OUTPUT:
prompt_id  group_id  type_id  prompt
0      0     0     a cat sitting on a stair railing
1      0     1     a feline sitting on a stair banister
2      0     2     on a stair railing sits a cat
3      0     3     there is a cat positioned on a stair railing
4      0     4     cat on stair railing
5      0     5     a cat calmly sitting on the stair railing
6      0     6     a cat is seated upon the stair railing
7      1     0     a red car parked on the street
8      1     1     a crimson automobile parked on roadway
9      1     2     on the street is parked a red car
10     1     3     there is a red car parked on the street
11     1     4     red car street parked
12     1     5     a red car neatly parked on the street
13     1     6     a red car is parked on the street
14      0     7     a cat that is sitting on a stair railing
15      0     8     the sitting of a cat on a stair railing
16      0     9     it is a cat that is sitting on a stair railing
17      0     10    a cat is present. The cat is sitting on a stair railing.
18      0     11    a cat is sitting on a stair railing
19      0     12    the image depicts a cat sitting on a stair railing
20      0     13    depict a cat sitting on a stair railing
21      1     7     a red car that is parked on the street
22      1     8     the parking of a red car on the street
23      1     9     it is a red car that is parked on the street
24      1     10    a red car is present. The car is parked on the street.
25      1     11    a red car is parked on the street
26      1     12    the image depicts a red car parked on the street
27      1     13    depict a red car parked on the street
```

### G.3 FIXED-SEMANTICS PERTURBATIONS

Listing 3: Instruction used to generate Fixed-Semantics Perturbations.

```
TASK: PROMPT TRANSFORMATION

You are given one base prompt describing a simple visual scene.

Generate semantic-preserving variations of this prompt according to the
transformation rules below.

OUTPUT FORMAT

Return ONLY a JSON object with exactly these string keys:

- "base": the original prompt, unchanged
- "lexical": synonym / near-synonym substitution only
- "syntactic": grammar, word order, or voice change; same words mostly preserved
- "structural": sentence architecture change, such as phrase to sentence or clause
    restructuring
- "compressed": remove non-essential words while keeping core meaning
- "expanded": add neutral modifiers or grammar glue without adding visual entities or style
- "style_shifted": change linguistic style only; do not add artistic/image style
- "determiner_shift": add, remove, or replace articles and determiners only where meaning
    remains unchanged
```

```
- "attribute_order": reorder attributes, modifiers, or descriptive phrases while preserving
    meaning
- "micro_paraphrase": small local paraphrases using equivalent expressions

CORE TRANSFORMATION RULES

You MUST preserve:
- main object(s)
- key action
- relationships between objects
- number of objects
- core attributes such as color, size, and position
- scene context if explicitly present

You MUST NOT:
- add new meaningful entities
- remove key entities
- change the action into a different action
- change scene meaning
- change number of objects
- introduce artistic or image styles such as anime, watercolor, or cinematic

TRANSFORMATION TYPE DETAILS

1. LEXICAL
- Only synonym / near-synonym substitution
- No structural changes
Example: cat sitting on railing -> feline sitting on banister

2. SYNTACTIC
- Change grammar / word order / voice
- Same words mostly preserved
Example: cat sitting on railing -> on railing sits a cat

3. STRUCTURAL
- Change sentence architecture
Example: cat on railing -> there is a cat positioned on railing

4. COMPRESSED
- Remove non-essential words
- Keep only core meaning
Example: a cat sitting on a stair railing -> cat on stair railing

5. EXPANDED
- Add neutral modifiers such as manner, soft adverbs, or grammar glue
- MUST NOT add new visual entities or style
Example: cat sitting on railing -> a cat calmly sitting on the railing

6. STYLE-SHIFTED
- Change linguistic style only, such as formal, colloquial, terse, or narrative
- MUST NOT add artistic style
Example: cat sitting on railing -> a cat is seated upon the railing

7. DETERMINER-SHIFT
- Add, remove, or replace articles and determiners where meaning remains unchanged
- MUST preserve object count and scene meaning
- No other substantial wording changes
Example: a cat sitting on a stair railing -> cat sitting on stair railing

8. ATTRIBUTE-ORDER
- Reorder attributes, modifiers, or descriptive phrases
- Preserve all attributes and their meanings
- No entity additions or removals
Example: a small red car parked on the street -> a red small car parked on the street

9. MICRO-PARAPHRASE
- Apply small local paraphrases using equivalent expressions
- May combine limited synonym substitution and minor grammatical changes
- Must preserve all visual content and scene semantics
Example: a red car parked on the street -> a red car parked along the street

FINAL RULE
Return ONLY the JSON object. No explanations, headings, markdown, or extra text.
```

### G.4 ATOMIC CONCEPT EXTRACTION

Listing 4: Instruction used to extract prompt-derived atomic concepts.

```
TASK: EXTRACT ATOMIC VISUAL ENTITIES FROM A CLEAN IMAGE CAPTION
```

```
You are given one English caption used as a clean text-to-image prompt.

Extract only concrete, visually groundable entities explicitly requested by
the caption. Do not extract attributes or relations. Do not use outside
knowledge and do not infer objects that are merely typical for the scene.

Return ONLY one JSON object in this exact form:

{
  "atomic_entities": [
    {
      "surface": "string",
      "canonical": "string",
      "glip_query": "string",
      "kind": "countable_entity or scene_region",
      "quantity": {"value": 1, "exact": true}
    }
  ]
}

WHAT TO EXTRACT

Include:
- people and animals;
- vehicles, clothing, tools, furniture, food, and other visible objects;
- explicitly mentioned scene regions or surfaces that can plausibly be
  localized, such as road, highway, beach, field, water, or sky.

Do not extract:
- colors, sizes, materials, shapes, patterns, or other attributes;
- actions such as riding, holding, standing, or eating;
- spatial or semantic relations;
- photographic or artistic style;
- implied, hidden, or merely likely objects.

ENTITY RULES

- `surface` is the shortest relevant noun phrase from the caption.
- `canonical` is a concise lowercase singular category.
- `glip_query` is a short English noun phrase suitable for object grounding.
- `kind` is `countable_entity` for object instances and `scene_region` for
  uncountable scene areas.
- Preserve meaningful compound categories: `traffic light`, `food truck`, and
  `tennis racket` must remain compounds.
- Do not include attributes in `canonical` or `glip_query`: `black motorcycle`
  becomes `motorcycle`.
- Preserve order of first mention.

DUPLICATE AND QUANTITY RULES

- Return at most one entry for each canonical category.
- Merge repeated mentions of the same category into that one entry.
- Explicit singular: `value = 1`, `exact = true`.
- Explicit number: use that number and `exact = true`.
- Two separately mentioned members of the same canonical category, such as
  `a man and a woman`, become one `person` entry with `value = 2`,
  `exact = true`.
- Unspecified plural: use the smallest guaranteed number, normally 2, and
  `exact = false`.
- Scene region: always `value = 1`, `exact = true`.

Empty `atomic_entities` is allowed. Use English only. Return JSON only, without
markdown or explanations.

EXAMPLE

Input:
A person quickly rides a black motorcycle on a highway while wearing a yellow helmet.

Output:
{
  "atomic_entities": [
    {
      "surface": "person",
      "canonical": "person",
      "glip_query": "person",
      "kind": "countable_entity",
      "quantity": {"value": 1, "exact": true}
    },
    {
      "surface": "motorcycle",
```

```
      "canonical": "motorcycle",
      "glip_query": "motorcycle",
      "kind": "countable_entity",
      "quantity": {"value": 1, "exact": true}
    },
    {
      "surface": "highway",
      "canonical": "highway",
      "glip_query": "highway",
      "kind": "scene_region",
      "quantity": {"value": 1, "exact": true}
    },
    {
      "surface": "helmet",
      "canonical": "helmet",
      "glip_query": "helmet",
      "kind": "countable_entity",
      "quantity": {"value": 1, "exact": true}
    }
  ]
}
```