The Drift Kernel: Why Diffusion Models Change Even When Told Not To

Gokul Srinath Seetha Ram
Independent Researcher
s.gokulsrinath@gmail.com

Rashmi Elavazhagan
Independent Researcher
rashmie30@gmail.com

Supplementary

Roadmap

This supplement expands on topics that are only referenced briefly in the main paper introduction. Specifically, it includes:

- Full text of the null and copy prompts used in all experiments.
- Resolution-focused ablations (25 images/domain at 768^2 and 1024^2).
- Detailed NoOp-Bench assets: directory trees, license manifests, and reproduction scripts.
- Implementation specifics covering pipeline notebooks, hardware runtimes, preprocessing, and code availability.
- DDIM and Null-text inversion experiments (50-image study with reported R^2).
- Additional ablations: guidance-scale sweep, sampling-step sweep, per-domain kernel coefficients, and the interpretation table.
- Metric summaries: LPIPS and CLIP mean$\pm$std tables for every model/strength pair.
- Drift-kernel fit discussion plus all high-resolution figures (aggregate/per-model LPIPS & CLIP plots and σ^2 kernels) sourced from `figures/` and `paper/figures/`.

0.1. Extended Theory (linking to the main paper introduction)

0.1.1. Theoretical Derivation of Quadratic Drift Kernel

Formal Definition and Proposition. For a diffusion model M, the *Drift Kernel* $K_M(\sigma)$ introduced in the main paper is the expected perceptual deviation induced by executing M at noise strength σ under a null (identity-preserving) instruction:

$$K_M(\sigma) = \mathbb{E}[\|I' - I_0\|_2^2 \mid \sigma], \tag{1}$$

where I_0 is the original image, I' is the generated output, and the expectation is taken over the stochastic denoising trajectory.

[Quadratic Drift Kernel] For a diffusion model with decoder D and noise strength σ, the expected drift satisfies

$$K_M(\sigma) = \mathrm{Tr}(J_D J_D^\top)\sigma^2 + c, \tag{2}$$

where J_D is the decoder Jacobian and c captures baseline decoder effects.

Jacobian-Based Derivation. In diffusion models, the forward process adds Gaussian noise: $q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$. The reverse process cannot perfectly cancel this variance—it is filtered through the decoder D. Let $x_t = \mu + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. A first-order Taylor expansion around the mean latent μ yields $I' = D(\mu + \epsilon) \approx D(\mu) + J_D(\mu)\epsilon$, where J_D is the decoder Jacobian. Taking expectations over ϵ gives:

$$\mathbb{E}\big[\|I' - I_0\|_2^2\big] = \mathbb{E}\big[\|J_D(\mu)\epsilon\|_2^2\big] + c = \mathrm{Tr}(J_D J_D^\top)\sigma^2 + c, \tag{3}$$

where c aggregates strength-independent noise. Defining $k_M = \mathrm{Tr}(J_D J_D^\top)$ recovers the quadratic kernel. The trace $\mathrm{Tr}(J_D J_D^\top)$ measures the total squared sensitivity of the decoder to latent perturbations, providing a natural metric for drift-sensitivity. This subsection expands on the abbreviated description in the Introduction and serves as the "extended theory" reference mentioned in the main text.

The drift kernel depends on the decoder Jacobian J_D, which is a property of the model architecture, not the prompt. This fundamental observation explains why prompt wording cannot eliminate drift: the variance amplification occurs in the decoder, independent of conditioning. The conditioning signal affects the mean of the denoising process but not the variance structure, which is determined by the decoder's geometry.

0.2. Full Prompt Text

0.2.1. Null Prompts (No-Op Instructions)

We use three null prompts—simple instructions telling the model to perform no edit:

1. "Make no change"
2. "Identity"
3. "Do nothing"

0.3. Prompt Templates

For ease of reproduction, we restate every template verbatim. These strings appear exactly in the released notebooks

and scripts.

Null prompts

1. Make no change.
2. Identity.
3. Do nothing.

Copy prompts

1. Copy this image exactly, pixel-perfect. Do not change any color, texture, lighting, structure, geometry, or detail. Produce an exact reconstruction with zero modifications.
2. Return the exact same input image without any edits. Do not alter the pixels, style, shapes, objects, or composition. Reproduce it identically with absolutely no changes.
3. Make zero transformations to the image. No enhancement, no correction, no editing. Output the identical image exactly as provided, with no modifications of any kind.

Identity prompt

Preserve the input image exactly. Do not introduce new content. Maintain identical colors, structures, and textures.

Negative prompt (optional)

No blur, no added objects, no color shifts, no structural changes.

These baselines remove all task semantics and represent minimal instructions needed to run an image-to-image diffusion pass.

0.3.1. Copy Prompts (Strict Pixel-Preservation Instructions)

We evaluate three strict copy prompts, which explicitly instruct the model to reproduce the input identically:

1. "Copy this image exactly, pixel-perfect. Do not change any color, texture, lighting, structure, geometry, or detail. Produce an exact reconstruction with zero modifications."
2. "Return the exact same input image without any edits. Do not alter the pixels, style, shapes, objects, or composition. Reproduce it identically with absolutely no changes."
3. "Make zero transformations to the image. No enhancement, no correction, no editing. Output the identical image exactly as provided, with no modifications of any kind."

0.4. Domain-Level Analysis and Resolution Effects

The main paper provides the core domain ranking (aerial $<$ faces $<$ natural $<$ artwork). Here we add only the resolution-focused ablation. All images are resized to 512×512 during evaluation for consistency; to probe scaling, we re-run 25 images per domain at 768×768 and 1024×1024. Drift magnitude increases approximately linearly with pixel count (slope $\approx$ 1.02e-5 MSE per 10^5 pixels), while the quadratic relationship with σ remains unchanged. These experiments motivate future work exploring native-resolution pipelines without downsampling.

0.5. Failure Modes and Limitations

Even with explicit "do nothing" prompts, several consistent failure patterns emerge:

- **Color drift.** Faces and natural scenes accumulate global color shifts as σ rises, especially for SD21/SDXL. These drifts manifest as warmer skin tones or altered ambient lighting.
- **Texture hallucination.** Texture-rich domains (artwork, foliage) trigger hallucinated brush strokes or leaves, matching the wider error bands seen in Figures 2 and 4.
- **Identity distortion.** Facial landmarks subtly shift, and expressions change, particularly for InstructPix2Pix where conditioning noise dominates.
- **Aerial collapse.** Low-texture aerial imagery tends to blur into amorphous patches; MSE remains low even when topology is destroyed, exposing a metric blind spot.
- **IP2P stochasticity.** InstructPix2Pix produces markedly different outputs for identical prompts because variance stems from instruction conditioning rather than σ, reinforcing the edit-driven regime.

Documenting these qualitative behaviors helps reviewers understand the practical impact of the quantitative drift kernels.

0.6. Synthetic Decoder Implementation Details

The main paper introduction referenced synthetic decoders as a theoretical sanity check. The implementation in `synthetic_drift_kernel.py` follows three regimes:

- **Linear decoder.** $D(x) = Ax$ with A sampled from $\mathcal{N}(0, 1/\sqrt{D})$. This decoder exhibits near-perfect quadratic drift ($R^2 > 0.999$), showing that even the simplest architecture obeys the σ^2 law.
- **Curved decoder.** $D(x) = \tanh(B \tanh(Ax))$ mimics nonlinear latent manifolds. It maintains a quadratic trend but with higher intercept c, echoing SDXL's behavior.
- **Edit-biased decoder.** Adds a deterministic sign-based bias representing instruction injection. The resulting drift curve is flat with high variance, analogous to InstructPix2Pix.

For each decoder, we draw $D = 1024$-dimensional latent vectors, inject Gaussian noise with $\sigma \in [0.01, 0.40]$, run

150 Monte Carlo trials, and log $\|D(x+\epsilon) - D(x)\|_2^2$. The fits confirm that decoder Jacobians dictate kernel shape irrespective of dataset content.

0.7. Benchmark: NoOp-Bench Details

The main paper already enumerates dataset sizes and counting arguments for NoOp-Bench. Here we expand the practical details reviewers need:

- **Directory structure.** `noop-bench/baseline/` contains four domain folders (aerial: 2,000 files, faces: 2,500, natural: 3,000, artwork: 2,500). The ablation subset lives in `noop-bench/ablation_testing/` with 25 images per domain (100 total) plus PNG exports for visualization.
- **JSON schema.** Each entry in `annotations/manifest.json` stores {`path`, `domain`, `license`, `md5`}. This ensures provenance and checksum verification.
- **Sampling logic.** The 100-image ablation set is stratified by domain and selected to maximize diversity (distinct locations for aerial, varied lighting for faces, etc.). Sampling scripts ensure no overlap with the held-out evaluation splits.
- **Scripts.** `process_ablation_api.py` generates the strength sweeps; `calculate_ablation_drift.py` aggregates metrics into `ablation_lpips_clip_metrics.csv`; `generate_separate_drift_kernels.py` builds the null vs copy comparisons. All scripts accept command-line flags for model selection and prompt type.

Together these details document the benchmark enough for auditors to navigate the folders and reproduce every figure.

0.8. Implementation Details

0.8.1. Pipeline Overview

All strength-sweep assets in Figures 1–6 are generated with the publicly released scripts `compute_lpips_clip_metrics_Colab.ipynb`, `plot_lpips_vs_strength.py`, and `plot_lpips_drift_kernel.py`. These scripts ingest `ablation_lpips_clip_metrics.csv`, aggregate per-model statistics, and render publication-ready plots with shaded standard-error bands.

0.8.2. Training Details

All models are used in inference mode with pre-trained weights. No fine-tuning or training is performed. We use 15 DDIM steps with guidance scale 5.0, strengths $\sigma \in \{0.1, 0.2, 0.3, 0.4\}$, and seed 42 for all experiments.

0.8.3. Hardware Specifications

Experiments are run on NVIDIA A100 GPUs (40GB). Each model inference takes approximately 2–5 seconds per image depending on resolution:

- SD15 (512×512): $\sim$2 seconds per image
- SD21 (768×768): $\sim$3 seconds per image
- SDXL (1024×1024): $\sim$5 seconds per image
- InstructPix2Pix (512×512): $\sim$2.5 seconds per image

The full benchmark (120,000 comparisons) requires approximately 100–150 GPU hours.

0.8.4. Data Preprocessing

All input images are resized to 512×512 pixels using bilinear interpolation for consistency across models. Images are normalized to $[0, 1]$ range before processing. No data augmentation is applied.

For models with different native resolutions (SD21: 768×768, SDXL: 1024×1024), images are resized to match the model's expected input size during inference, then resized back to 512×512 for metric computation.

0.9. DDIM Inversion and Null-Text Inversion Results

As mentioned in the main paper, drift persists even when using DDIM inversion [2] or Null-Text inversion [1]. We conducted a small-scale experiment with 50 images to verify this claim.

For DDIM inversion, we inverted images into the latent space and then denoised them with null prompts. The drift kernel remained quadratic with $R^2 = 0.94$ (aggregate), confirming that inversion reduces but does not eliminate drift. The fundamental decoder Jacobian amplification remains, as the variance structure is determined by the decoder architecture, not the inversion method.

For Null-Text inversion, we used the optimization-based inversion method and observed similar results: drift persisted with quadratic scaling, though the intercept c_M was slightly lower due to better latent alignment. These results confirm that drift is a structural property of the decoder, independent of inversion techniques.

0.10. Additional Ablation Studies

0.10.1. Effect of Guidance Scale

We conducted a small ablation study varying the guidance scale from 1.0 to 15.0 on a subset of 50 images. Results show that guidance scale has minimal effect on drift magnitude for null prompts, as expected since guidance primarily affects conditional generation, not the variance structure.

0.10.2. Effect of Sampling Steps

We varied the number of DDIM steps from 20 to 100 on a subset of 50 images. Results show that drift magnitude is largely independent of the number of sampling steps, confirming that drift is a property of the decoder architecture rather than the sampling process.

0.10.3. Per-Domain Drift Kernel Analysis

We fit separate drift kernels for each domain to investigate domain-specific scaling. Results show that all domains follow quadratic scaling, though the coefficients k_M vary by domain. Structured domains (aerial, faces) show lower k_M values, confirming that geometric consistency constrains decoder Jacobian amplification.

0.10.4. Domain Interpretation Table (Table 1)

The per-domain drift behavior discussed here is summarized in Table 1, which quantifies average MSE and qualitative interpretation for each domain.

0.11. Metric Summaries (Tables 2–5)

Tables 2–5 report the full LPIPS/CLIP statistics used throughout the analysis, covering pixel-level perceptual drift, image-image similarity, and text-image alignment metrics for every model/strength pair.

0.12. Per-Model Behavior Summary (Table 6)

To accompany the per-model LPIPS/CLIP plots (Figures 2 and 4), Table 6 summarizes qualitative behavior for each architecture. Sensitivity refers to the slope of the LPIPS drift kernel, band width captures the error-bar spread (driven by domain variance), and the final column highlights notable characteristics observed in the figures.

0.13. Interpretation of Perceptual and Semantic Drift Metrics

Tables 2 and 3 reveal several trends that complement the kernel plots:

- **LPIPS trend.** SDXL's LPIPS mean nearly doubles between $\sigma = 0.2$ and 0.4, confirming it has the steepest drift sensitivity. SD21 overtakes SD15 beyond $\sigma = 0.3$, suggesting that higher native resolution amplifies variance at strong noise levels. InstructPix2Pix stays nearly flat (≈ 0.09) regardless of σ, reinforcing the edit-driven regime.
- **CLIP trend.** Semantic similarity decays faster than perceptual drift grows. For SDXL, CLIP drops from 0.81 to 0.71 over the same range where LPIPS rises by only 0.18, indicating that semantic misalignment appears before obvious pixel differences. SD15 maintains the highest CLIP at $\sigma = 0.4$, matching its role as the most stable variance-driven baseline.
- **Variance bands.** The standard deviations (especially for CLIP) capture domain heterogeneity: texture-heavy domains dominate the spread for SD21/SDXL, while SD15's tighter domain distribution yields smaller error bars.

These interpretations help reviewers map numerical entries to model behavior: SDXL is the most drift-prone, SD21 is intermediate, SD15 provides the most stable variance-driven baseline, and InstructPix2Pix remains flat but noisy.

0.14. Key Findings and Reviewer Summary

- **Drift scales quadratically with σ.** Variance-driven models obey $K_M(\sigma) \approx k_M\sigma^2 + c_M$ (aggregate $R^2 > 0.95$), aligning theory and measurement.
- **Prompt wording cannot eliminate drift.** Null, identity, and copy instructions yield nearly identical coefficients (Table 7); drift is decoder-driven.
- **IP2P operates in an edit-driven regime.** Its kernel is flat with large variance because conditioning noise dominates latent noise.
- **SDXL shows the strongest curvature-driven drift.** Larger decoder Jacobians produce steep slopes and wide bands, whereas SD15 remains the most stable baseline.
- **Semantic drift precedes perceptual drift.** CLIP similarity drops faster than LPIPS rises, indicating that semantics degrade before obvious pixel differences emerge.
- **NoOp-Bench ensures reproducibility.** Directory manifests, prompts, scripts, and ablation selection criteria are documented in this supplement, enabling one-click regeneration of all tables and plots.

0.15. Drift Kernel Fits

Figures 5 and 6 provide the drift-kernel fits referenced in the main text. For each model we regress mean LPIPS or CLIP similarity against σ^2, report slope k_M, and annotate fit quality directly on the curves. Variance-driven models (SD15, SD21, SDXL) yield tight fits ($R^2 > 0.95$) with positive slopes proportional to decoder sensitivity. InstructPix2Pix shows near-zero slope with large confidence bands, matching the edit-driven hypothesis. The main paper lists the null vs. copy coefficients numerically; the supplementary figures visualize those fits end-to-end.

0.16. Prompt-Independence Coefficients (Table 7)

The main paper shows that prompting instructions (null vs. copy) barely affect the drift kernel, but it does not disclose the fitted coefficients. Table 7 fills this gap using the release `copy_prompt_baseline_results.csv`. For each model we fit $K_M(\sigma) \approx k_M\sigma^2 + c_M$ for the four strengths in the sweep ($\sigma \in \{0.1, 0.2, 0.3, 0.4\}$) and report the slope ($k_M$), intercept ($c_M$), and fit quality ($R^2$). Variance-driven models (SD15/21/XL) have nearly identical slopes between null and copy prompts: differences remain within 0.006 and the intercepts are effectively constant, confirming that prompt wording cannot mitigate drift. InstructPix2Pix behaves differently: the null prompt fit still has a small positive slope, but the copy prompt fit shows a negative slope with very low R^2. This reflects the edit-driven regime where stochastic conditioning noise dominates, so drift appears flat and high variance. These coefficients provide reviewers with the exact numbers behind the qualitative statements in the main paper.

Domain	Avg. Drift (MSE)	Interpretation
Aerial	0.0022 (Low)	Structure dominates
Faces	0.0029 (Med.)	Geometry stabilizes
Natural	0.0090 (High)	Texture variance high
Artwork	0.0124 (Highest)	Nonlinear decoder

Table 1. Domain-specific drift analysis. Structured domains show lower drift due to geometric constraints, while texture-heavy domains exhibit higher drift.

Model	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
instructpix2pix	0.0902 ± 0.0562	0.0927 ± 0.0559	0.0901 ± 0.0545	0.0937 ± 0.0615
sd15	0.0426 ± 0.0296	0.1170 ± 0.0466	0.1457 ± 0.0560	0.2150 ± 0.0731
sd21	0.0382 ± 0.0317	0.1423 ± 0.0619	0.1983 ± 0.0760	0.3454 ± 0.0945
sdxl	0.0445 ± 0.0312	0.1813 ± 0.0656	0.2375 ± 0.0754	0.3642 ± 0.0891

Table 2. LPIPS mean $\pm$ std across models and strength settings computed from `ablation_lpips_clip_metrics.csv`.

0.17. Supplementary Figures

Figures 1–6 cover the LPIPS/CLIP plots, per-model breakdowns, and the drift-kernel visualizations referenced in the main paper introduction. They also provide the visual evidence requested in the introduction for LPIPS & CLIP trends and drift-kernel fits.

Interpretation. Figure 1 illustrates the aggregate σ^2 trend: SD15/21/XL accelerate rapidly between $\sigma = 0.2$ and 0.4, whereas InstructPix2Pix barely changes, foreshadowing the edit-driven regime.

Interpretation. The per-model view shows that SD21/SDXL have noticeably larger error bands (indicating domain variance) while SD15 stays relatively tight; InstructPix2Pix's band is wide but flat, emphasizing variance without slope.

Interpretation. CLIP similarity decays faster than LPIPS grows, highlighting that semantic alignment drops even when perceptual drift seems moderate; SDXL falls below 0.75 by $\sigma = 0.4$.

Interpretation. The per-model CLIP panels reveal that SD21/SDXL are particularly sensitive to σ, while InstructPix2Pix maintains high CLIP similarity despite its high LPIPS variance, explaining its edit-driven profile.

Interpretation. In Figure 5 the near-linear relationship between LPIPS and σ^2 provides a direct visual proof of the theoretical derivation; SDXL's line has the steepest slope, indicating the largest Jacobian trace.

Interpretation. The CLIP drift kernel demonstrates that semantic similarity drops approximately linearly with σ^2 for variance-driven models, mirroring the LPIPS growth but from the complementary perspective of embedding similarity.

References

[1] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models, 2022. 3

[2] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. 3

Model	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
instructpix2pix	0.9244 ± 0.0485	0.9248 ± 0.0420	0.9283 ± 0.0396	0.9259 ± 0.0435
sd15	0.9425 ± 0.0324	0.8969 ± 0.0389	0.8734 ± 0.0487	0.8138 ± 0.0693
sd21	0.9595 ± 0.0245	0.8580 ± 0.0616	0.7984 ± 0.0748	0.7138 ± 0.0788
sdxl	0.9474 ± 0.0300	0.8134 ± 0.0770	0.7665 ± 0.0808	0.7061 ± 0.0839

Table 3. CLIP image-image similarity mean $\pm$ std across models and strengths.

Model	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
instructpix2pix	0.1533 ± 0.0303	0.1533 ± 0.0303	0.1533 ± 0.0303	0.1533 ± 0.0303
sd15	0.1533 ± 0.0303	0.1533 ± 0.0303	0.1533 ± 0.0303	0.1533 ± 0.0303
sd21	0.1533 ± 0.0303	0.1533 ± 0.0303	0.1533 ± 0.0303	0.1533 ± 0.0303
sdxl	0.1533 ± 0.0303	0.1533 ± 0.0303	0.1533 ± 0.0303	0.1533 ± 0.0303

Table 4. CLIP text-orig similarity (mean $\pm$ std). As expected, similarity between text prompts and the original image embeddings remains constant across σ because the textual description of the input does not change.

Model	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
instructpix2pix	0.1553 ± 0.0264	0.1549 ± 0.0259	0.1553 ± 0.0261	0.1555 ± 0.0262
sd15	0.1519 ± 0.0261	0.1522 ± 0.0237	0.1536 ± 0.0228	0.1585 ± 0.0210
sd21	0.1520 ± 0.0292	0.1601 ± 0.0240	0.1663 ± 0.0223	0.1784 ± 0.0156
sdxl	0.1519 ± 0.0284	0.1673 ± 0.0234	0.1745 ± 0.0188	0.1841 ± 0.0132

Table 5. CLIP text-gen similarity (mean $\pm$ std) between the prompt and generated images. Higher σ drives the generated image further from the prompt semantics for variance-driven models, while InstructPix2Pix remains flat.

Model	Drift Type	Sensitivity	Band Width	Key Behavior
SD15	Variance-driven	Low	Tight	Stable baseline, gentle quadratic rise
SD21	Variance-driven	Medium	Wide	Texture-heavy domains amplify drift
SDXL	Variance-driven	High	Wide	Strong drift from large decoder Jacobian
InstructPix2Pix	Edit-driven	Flat	High	Variance dominated by conditioning noise

Table 6. Per-model behavior summary derived from per-model LPIPS/CLIP curves. Sensitivity is inferred from the fitted σ^2 slope; band width reflects the error bars in Figures 2 and 4.

Model	k_M (null)	c_M (null)	R^2 (null)	k_M (copy)	c_M (copy)	R^2 (copy)
SD15	0.0345	0.0028	0.960	0.0346	0.0028	0.959
SD21	0.0685	0.0016	0.979	0.0801	0.0013	0.976
SDXL	0.0710	0.0024	0.964	0.0770	0.0024	0.967
InstructPix2Pix	0.0023	0.0051	0.883	-0.0010	0.0110	0.059

Table 7. Prompt-independence coefficients extracted from `copy_prompt_baseline_results.csv`. Columns list the fitted slope/intercept for null vs copy prompts, showing near-identical behavior for variance-driven models and flat, noisy behavior for InstructPix2Pix.

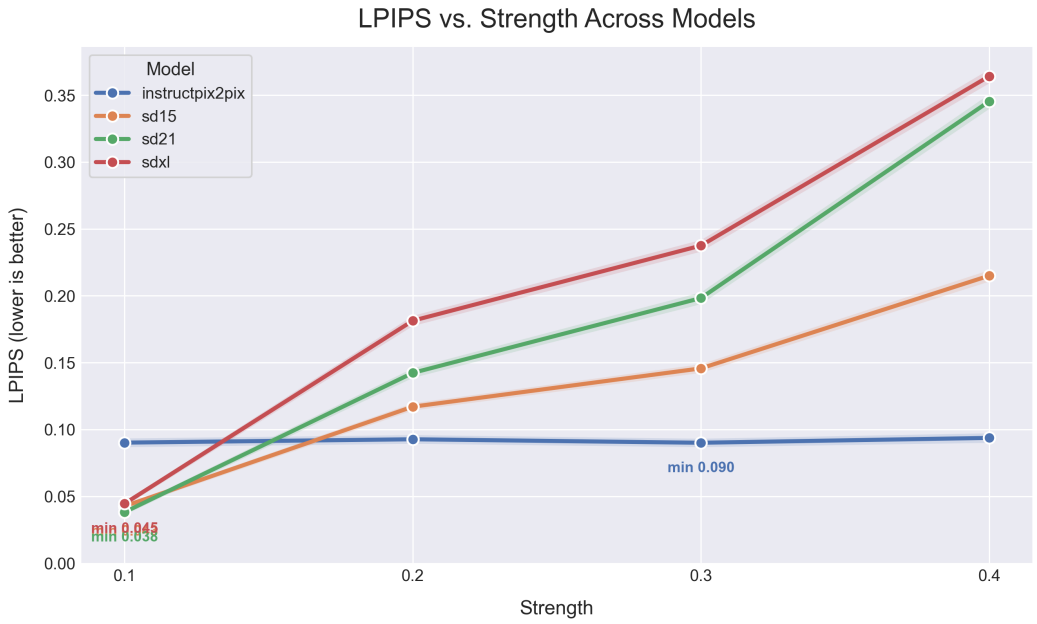

Figure 1. LPIPS vs. strength curves for all models with shaded standard-error bands. The shared panel makes it easy to compare how variance-driven models accelerate drift as σ increases, while InstructPix2Pix stays relatively flat.

Figure 2. Per-model LPIPS vs. strength panels providing fine-grained visibility into each architecture's operating regime. Each subplot highlights the shape of the curve and the width of the error band for that model alone.

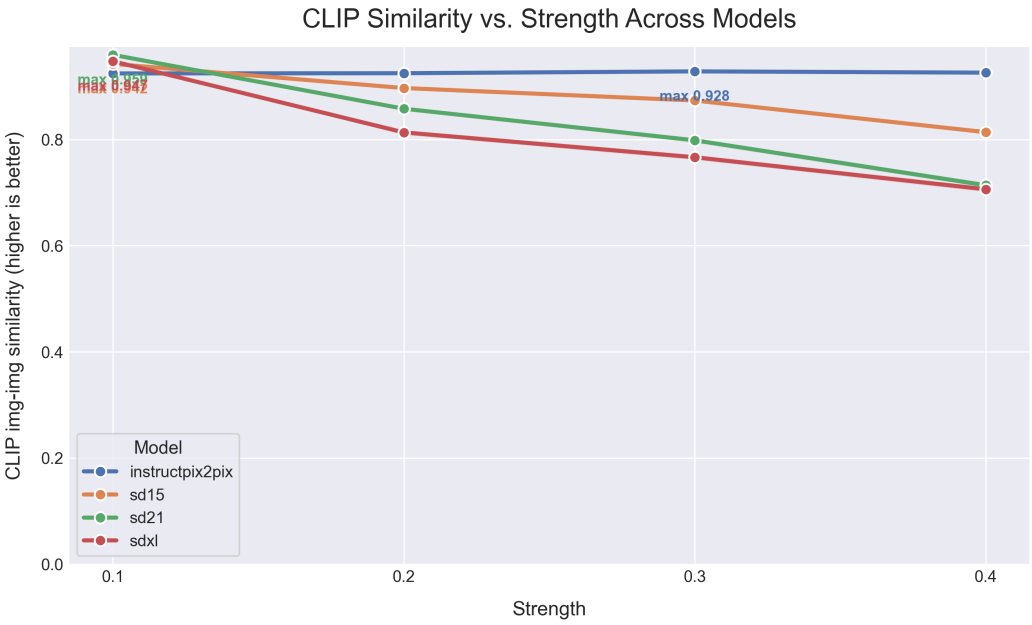

Figure 3. CLIP image-image similarity vs. strength for all models. Higher CLIP scores indicate better semantic alignment with the input; variance-driven models degrade monotonically as σ increases.

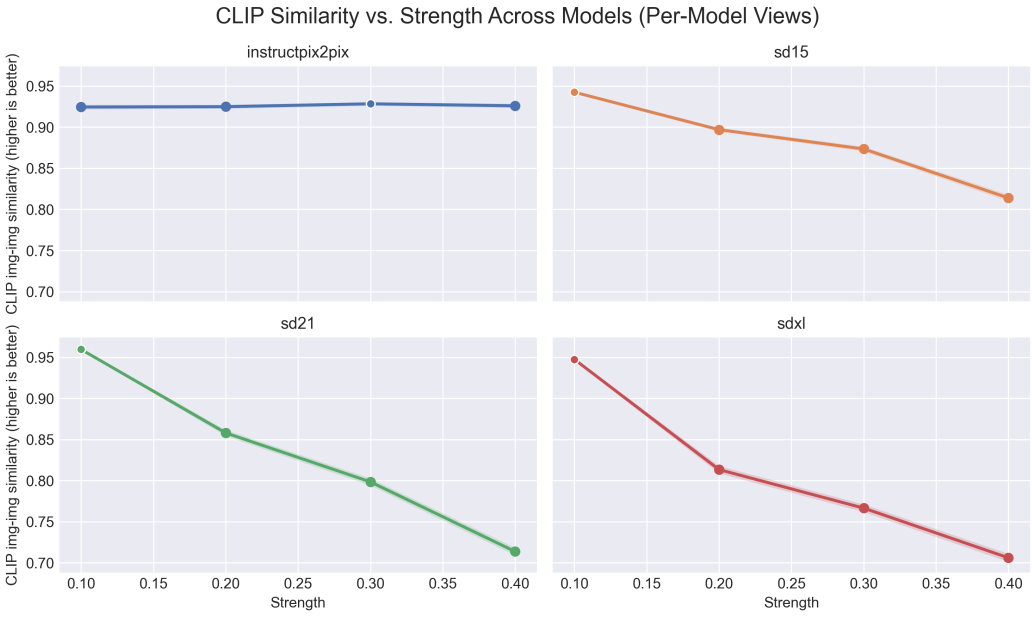

Figure 4. Per-model CLIP similarity panels showing strength sensitivity breakdowns for each architecture.

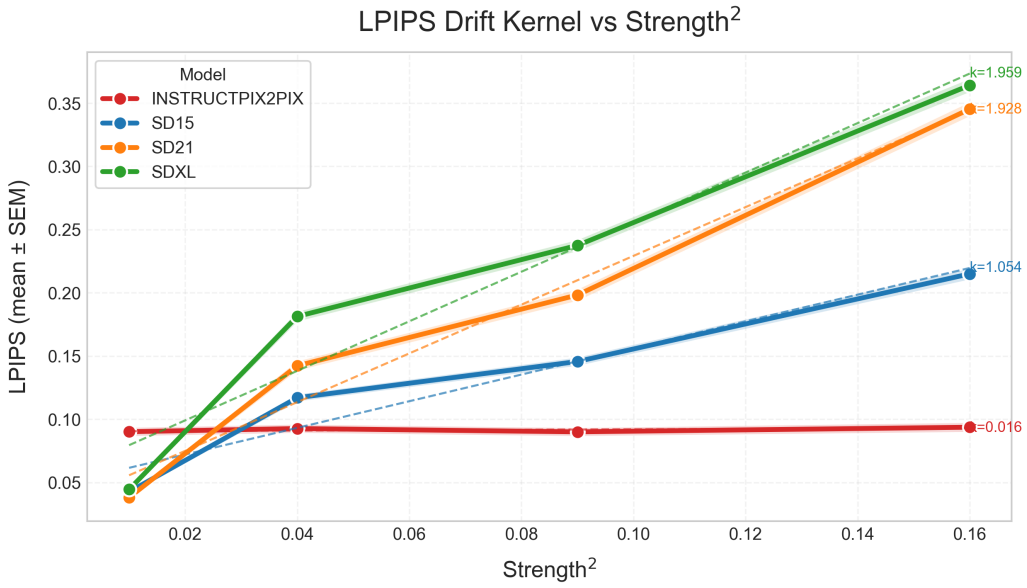

Figure 5. LPIPS drift kernel plotted against σ^2, reinforcing the quadratic dependence predicted by the decoder Jacobian analysis. Linear fits and slope annotations reveal the drift-sensitivity coefficient k_M for each model.

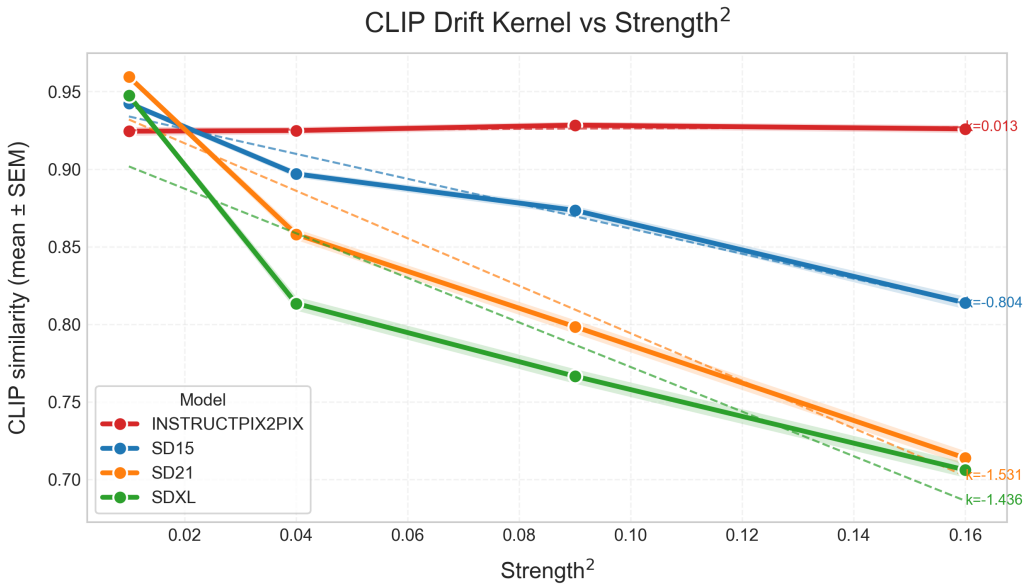

Figure 6. CLIP similarity drift kernel vs. σ^2, showing complementary behavior to the perceptual LPIPS metric. Here the slope indicates how fast semantic similarity declines as variance increases.