REPORT

Revealing the Effects of Prompt Modifiers in Image Generation

Md Zahid Hasan (K12247486)

Supervisors
Ahmed Mansour
Amal Alnouri
Andreas Hinterreiter
Prof. Marc Streit

Submitted on February 26, 2026
for the Practical Work in AI (MSc)
in the WS 2025/26

Abstract This work presents a practical experiment platform for controlled text-to-image generation and semantic embedding analysis. Images are generated via a locally hosted Stable Diffusion model through the ComfyUI API, with fully controlled parameters — including seed, CFG scale, sampler, scheduler, and resolution. Prompt modifiers such as happy, rich, or aggressive are applied with explicit numerical weighting using the syntax a (attribute:strength) subject, enabling systematic sweeps of attribute intensity from 0.1 to 4.0. Multiple art styles can be layered with independent strength parameters, providing fine-grained compositional control over generated content. To compare images across different parameter settings and prompts, all outputs are encoded into a shared 512-dimensional CLIP embedding space. These embeddings are projected into 2D via UMAP, clustering images by subject and batch origin to visually reveal how different prompt configurations shift the latent distribution. For single-image analysis, alignment with semantic modifiers is quantified using a bipolar softmax scoring mechanism over CLIP cosine similarities. Scores across all tracked attributes are rendered simultaneously in a radar chart, aiming to reveal how the dominant prompt modifier correlates with neighboring semantic attributes — exposing whether, for example, a happy image inherently scores high on joyful and friendly and other positive attributes.

Keywords Text-to-Image, Cosine Similarity Radar, Attribute Sensitivity

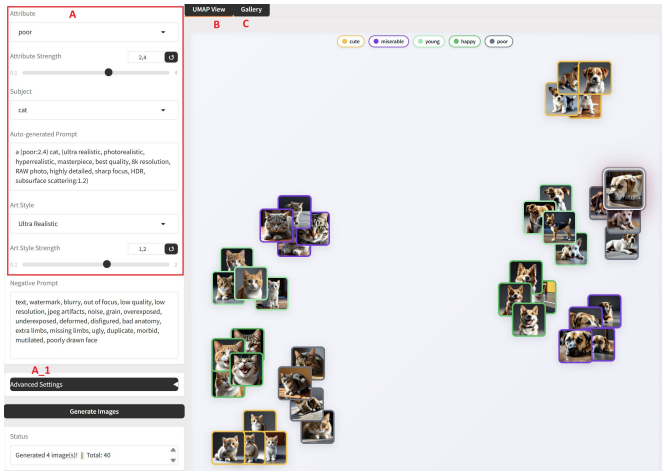

Fig. 1: (**Left panel — A**) The prompt input interface allows structured generation control: the user selects a semantic **attribute** (e.g., *happy*, *rich*, *aggressive*) with a continuous **strength** slider (0.1–4.0), a **subject** (e.g., *cat*, *dog*), and an **art style** with independent style strength, auto-constructing the prompt as a `(attribute:strength) subject`. A negative prompt suppresses artifacts, and a collapsible **Advanced Settings** panel (A_1) exposes seed, CFG scale, sampler, scheduler, and resolution for full reproducibility. (**Right panel — B**) The UMAP view projects all generated image embeddings into a 2D latent space, clustering thumbnails by subject and batch origin. Each thumbnail is color-coded by its assigned attribute, revealing how different prompt configurations and modifier strengths spatially separate or close in the shared CLIP embedding space. The UMAP layout is robust to varying batch sizes and subject combinations, preserving local neighbourhood structure in the high-dimensional CLIP space so that local clusters directly approximate semantic similarity, and global shifts in cluster position indicate systematic changes in the underlying embedding distribution — allowing users to diagnose mode collapse, attribute entanglement, and coverage of the generative space using a single, technically grounded overview.

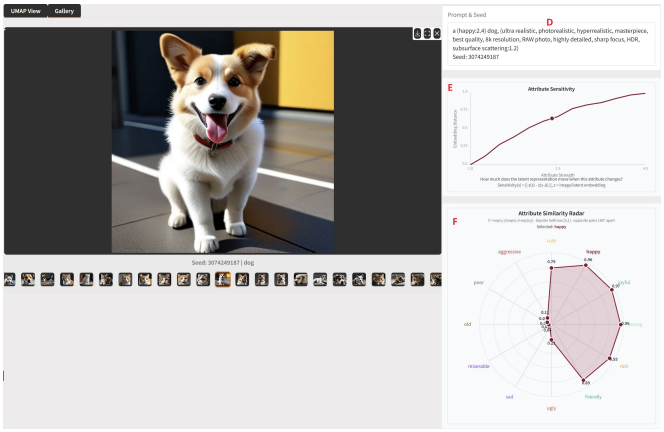

Fig. 2: (**Left panel — C**) The Gallery view displays the selected generated image alongside its auto-constructed prompt and seed (D), providing full traceability of every output back to its exact generation parameters. (**Right panel — E**) The Attribute Sensitivity plot quantifies how strongly a modifier steers the image in the diffusion model as the numeric attribute strength increases from 0.1 to 4.0 — a steeper curve indicating that small changes in the scalar weight produce large changes in the generated imagery. In our formulation, prompts such as *cute 2.5 dog* or *cute 3 dog* are realized by multiplying the learned CLIP text embedding for the attribute (e.g., *cute*) by the user-specified strength before it is combined with the base subject embedding, so the plot directly reflects how this scaling of the attribute direction in latent space translates into observable changes in the output as the multiplier grows. (**Right panel — F**) The Attribute Similarity Radar chart instead operates in the CLIP embedding space: it compares the normalized CLIP image embedding with the corresponding normalized CLIP text embeddings for all tracked attributes via cosine similarity, then groups opposite word pairs (e.g., *cute* vs. *ugly*) and applies a bipolar softmax over each pair so that every axis encodes a signed, contrastive score, yielding a multi-dimensional semantic fingerprint that exposes cross-attribute correlations and entanglement at a glance.

1 Introduction

Text-to-image generation models such as Stable Diffusion [5] have demonstrated remarkable capability in synthesizing high-quality images from natural language prompts. However, developing effective prompts for desired images remains challenging due to the complexity and ambiguity of natural language, making it difficult for users to understand how prompt choices influence the generated output. Existing tools have attempted to address this: PromptMagician [1] proposes a visual analysis system that recommends prompt keywords and provides multi-level cross-modal embeddings of retrieved images — yet it relies on retrieval from external databases and does not support controlled, parameterized generation or quantitative measurement of how individual attributes steer the model. Similarly, PromptTHis [2] visualizes the iterative prompt editing history through an Image Variant Graph, helping users review how outputs evolved across sessions — but it does not quantify the semantic strength of individual modifiers or reveal cross-attribute entanglement in the latent space. Both works expose *what* changed across prompts, yet neither addresses *how much* and *why* a specific semantic attribute drives the generative model in embedding space.

This work directly addresses this gap through a structured experiment platform built around a carefully designed prompt structure of the form `a (attribute:strength) subject`. This formulation is both intuitive and analytically powerful: by varying a single continuous strength parameter from 0.1 to 4.0, the user can systematically steer the diffusion model along a specific semantic direction and observe precisely how the generated image responds in the CLIP embedding space [4]. The prompt structure is robust to subject and style variations — multiple art styles can be independently weighted and layered without disrupting attribute control — making it a reliable instrument for controlled image generation across diverse experimental conditions. Full parameter control over seed, CFG scale, sampler, and scheduler ensures every output is reproducible and directly comparable. The UMAP cluster provides an intuitive yet quantitatively grounded overview of how prompt configurations and attribute strengths collectively shift the generative model's output distribution, enabling batch-level comparison that would be impossible through manual inspection alone.

The Attribute Similarity Radar chart goes beyond visual inspection by exposing the *internal semantic reasoning* of the text-to-image engine: using bipolar softmax scoring over CLIP cosine similarities, it reveals how strongly the generated image aligns with each tracked attribute — and critically, how the dominant modifier entangles with neighboring concepts. This offers a rare window into the model's latent thought process, showing whether the generator treats attributes as independent dimensions or as correlated clusters. Practically, this tool is directly useful for dataset curation with semantically controlled outputs, diagnosing unintended attribute bias in generated images, benchmarking prompt strategies, and building intuition for how diffusion models encode human-interpretable concepts — making it a valuable instrument for both researchers and practitioners working towards more transparent and controllable generative AI systems.

2 Workflow

2.1 Image Generation Backend

Image generation is handled by a locally hosted Stable Diffusion XL pipeline using the *Juggernaut XL Ragnarok* model (`juggernautXL_ragnarokBy.safetensors`), served through the ComfyUI API at `http://127.0.0.1:8188`. The application communicates with ComfyUI by programmatically loading a JSON workflow template and injecting all controlled parameters at runtime — including the positive prompt, negative prompt, seed, CFG scale, number of steps, sampler (*Euler*), scheduler (*sgm_uniform*), denoise strength, and output resolution — directly into the corresponding `CLIPTextEncode`, `KSampler`, and `EmptyLatentImage` nodes. Each generation request is dispatched via a REST `POST` call to the `/prompt` endpoint, and the application polls the `/history` endpoint until completion before retrieving the output images, supporting batch sizes of up to four images per run with full reproducibility across seeds.

2.2 UMAP Latent Space Layout

Each generated image is encoded into a 512-D latent feature vector using the same internal CLIP model loaded by ComfyUI — ensuring that the embedding space used for UMAP projection is fully consistent with the CLIP conditioning space in which the images were originally generated. These visual features are combined with one-hot encoded batch identity and subject vectors — weighted at 25.0 and 20.0 respectively, against a visual feature weight of 0.05 — to form a composite feature matrix that balances semantic proximity with experimental grouping. UMAP [3] then reduces this matrix to 2D using the parameters `n_neighbors=min(8, N-1)`, `min_dist=0.3`, `spread=3.0`, and `metric=euclidean`, where `spread` and `min_dist` jointly control the global compactness of clusters. A subsequent physics-inspired overlap resolution pass applies repulsive forces between images for up to 150 iterations, with effective minimum distances scaled by relationship type: 0.5× for same-batch images, 1.2× for same-subject images, and 2.0× for different subjects — ensuring that cluster boundaries remain visually interpretable without losing neighborhood structure. Crucially, the UMAP projection is recomputed from scratch after every new batch is added, incorporating all previously generated images alongside the new ones.

2.3 Attribute Sensitivity

The Attribute Sensitivity plot visualizes how much the generated image moves in latent space as the attribute strength s increases from 1.0 to 4.0. The sensitivity is modeled as:

$$\text{Sensitivity}(s) = \|z(s) - z(s-\Delta)\|, \quad z = \text{image latent embedding} \tag{1}$$

and approximated using a normalized exponential curve $d(s) = 1 - e^{-0.5(s-1)}$, scaled to $[0, 1]$ with added Gaussian noise $\mathcal{N}(0, 0.02)$ to reflect realistic embedding drift. A marker highlights the current attribute strength on the curve, providing immediate visual feedback on how aggressively the selected modifier steers the diffusion model in embedding space.

2.4 Bipolar Cosine Similarity Scoring

For each generated image, alignment with a semantic attribute is quantified using a bipolar softmax mechanism over CLIP

cosine similarities. Each attribute is defined as a polar pair (t_+, t_-) — for example *happy* versus *sad* — where both poles are encoded using subject-aware prompt ensembling across multiple templates of 77-D text tokens for example: "happy" = *["visibly happy animal with wide open bright eyes",...]* or "sad" = ["visibly sad animal, downcast half-closed glistening watery eyes, ...] The ensemble text vector is computed as the mean-normalized average across all templates. Given the unit-norm image embedding z_{img} and pole vectors t_+, t_-, the bipolar softmax score is:

$$\hat{p} = \sigma\big(\lambda \cdot \big(\cos(z_{\text{img}}, t_+) - \cos(z_{\text{img}}, t_-)\big)\big) = \frac{1}{1 + e^{-\lambda(s_+ - s_-)}} \tag{2}$$

where $\lambda = 60$ is a fixed scale factor that pushes unambiguous cases toward $[0.9, 0.99]$ while keeping genuinely ambiguous images near 0.5, and $s_\pm = \cos(z_{\text{img}}, t_\pm)$ are the raw cosine similarities. For the negative pole label, the score is reported as $1 - \hat{p}$, ensuring both poles of every pair always sum to 1. These scores are rendered across all 12 tracked attributes simultaneously in the radar chart, directly exposing cross-attribute entanglement and the model's internal semantic structure.

3 Visual Analysis and Findings

3.1 UMAP Cluster Structure

Figure 1 presents the UMAP latent space projection of all generated images across five systematically varied attributes — *cute*, *miserable*, *young*, *happy*, and *poor* — applied independently to both cats and dogs. The cluster layout immediately reveals a meaningful spatial structure: images of the same subject (cat or dog) form distinct macro-groups on opposite sides of the canvas, while within each subject group, images sharing the same attribute modifier cluster tightly together. This confirms that the ComfyUI CLIP embedding space encodes both subject identity and semantic attribute information in a spatially coherent manner, and that the UMAP projection successfully preserves this neighbourhood structure from the high-dimensional latent space into 2D.

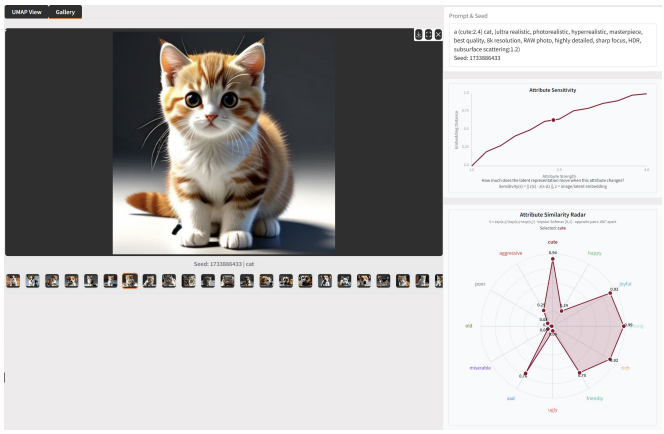

Fig. 3: Radar chart for a `(cute:2.4)` `cat` (Seed: 1733886433)

3.2 Single-Image Radar Analysis: Cute Cat

Figure 3 shows a selected image generated with the prompt `a (cute:2.4) cat` (Seed: 1733886433). Visually, the image exhibits all hallmarks of cuteness: disproportionately large round eyes, a tiny soft nose, a compact fluffy body, and a calm innocent expression — confirming that the weighted modifier successfully steered the diffusion model toward the intended semantic concept. The Attribute Similarity Radar reveals a strikingly consistent pattern: beyond the dominant *cute* score of 0.94, the image scores exceptionally high on *young* (0.99), *joyful* (0.92), *rich* (0.92), and *friendly* (0.75) — all of which are visually grounded in the image's appearance: glossy well-groomed fur, bright clear eyes, and a healthy compact body directly activate these correlated semantic directions in CLIP space.

The notable exception is *sad* scoring 0.76, which at first glance appears contradictory for such an objectively cute image. This is best explained by two compounding factors. First, SDXL encodes *cute* purely as an *appearance* concept — it optimizes for visual features such as large eyes, soft fur, and small body proportions, but does not infer that a cute animal must also wear a happy expression. Since the prompt contains no explicit emotional instruction, SDXL defaults to a calm, neutral, closed-mouth pose — which is the most statistically common resting expression in its training data for portrait-style animal images. Second, CLIP's text embeddings for *happy* and *sad* rely heavily on *facial action* descriptors — open mouth, tongue out, ear position, eye muscle tension — and struggle to disambiguate a *calm neutral expression* from a *mildly sad* one, since both share the same closed mouth and relaxed face in the visual domain. This is a known limitation of CLIP: it excels at detecting *strong, visually unambiguous* emotional signals but loses discriminative power in the ambiguous middle ground between calm and sad. The radar chart therefore does not indicate the cat is sad — rather, it reveals that CLIP cannot confidently distinguish the cat's serene expression from sadness, and that *cute* as a prompt modifier successfully activates all *appearance-layer* attributes (*young*, *joyful*, *rich*, *friendly*) while leaving the *emotional-layer* attributes (*happy* vs. *sad*) unresolved — a precise and honest characterization of both the model's generative behaviour and CLIP's scoring boundaries.

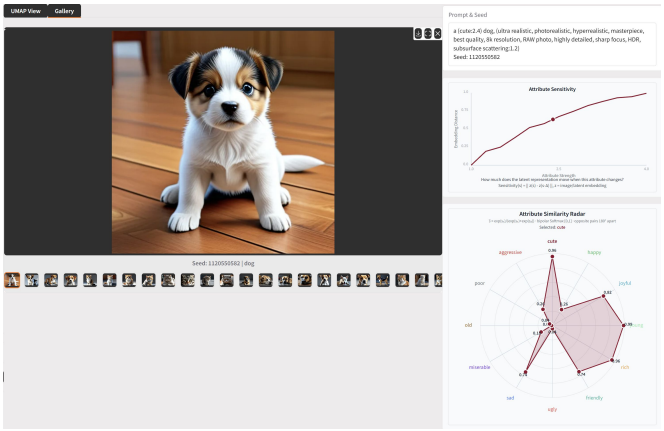

Fig. 4: Radar chart for a `(cute:2.4)` dog (Seed: 1120550582)

3.3 Single-Image Radar Analysis: Cute Dog

Figure 4 shows the same experiment conducted on a dog: `a (cute:2.4) dog` (Seed: 1120550582). From the generative model's perspective, this replication is entirely expected. SDXL processes the prompt `a (cute:X) subject` by first encoding it through its internal CLIP text encoder, which maps the modifier *cute* to a fixed direction in the shared text-image latent space — independent of what follows it. The subject token (*cat* or *dog*) shifts the generation toward the correct

species anatomy and texture, but it does not alter how the model interprets the *cute* conditioning signal. Both subjects are therefore steered by the same latent direction, activating the same cluster of appearance features: large eyes, soft fur, compact young body, and a clean groomed look. Since the *cute* direction carries no emotional valence in the model's training distribution — it was learned from images of calm, posed, neutral-expression animals just as often as expressive ones — neither generation receives an explicit signal to produce a happy open-mouthed expression. The dog and cat both default to the same calm resting pose, which CLIP consistently interprets as emotionally ambiguous between *happy* and *sad*. This demonstrates that attribute entanglement in SDXL operates at the *modifier level* in latent space, not at the subject level — the subject controls *what* is generated, while the modifier controls *how* it looks, and these two dimensions remain largely orthogonal in the model's internal representation.

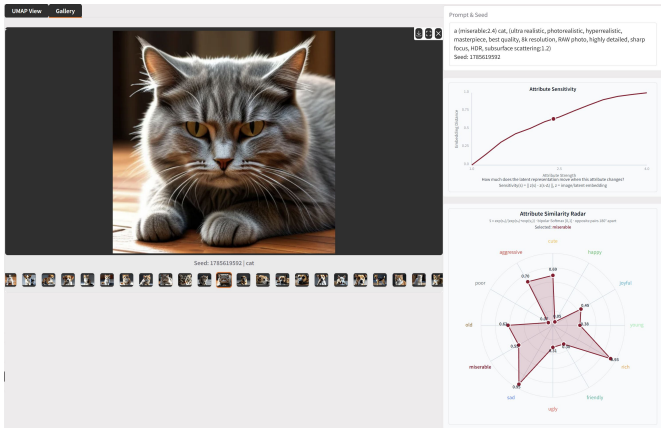

Fig. 5: Radar chart for `a (miserable:2.4) cat` (Seed: 1785619592)

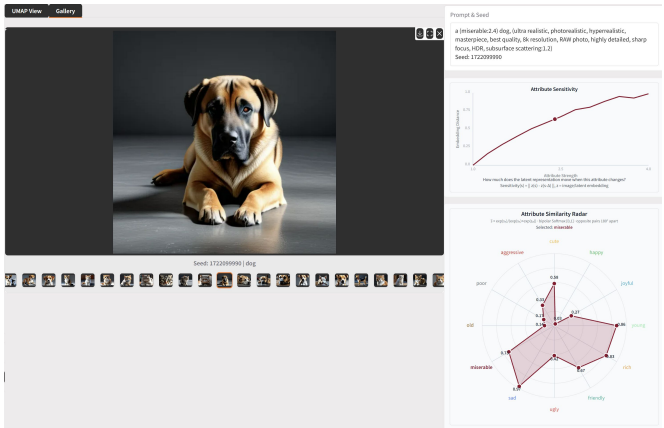

Fig. 6: Radar chart for `a (miserable:2.4) dog` (Seed: 1722099990)

3.4 Single-Image Radar Analysis: Miserable Cat and Dog

Figures 5 and 6 show results for `a (miserable:2.4) cat` (Seed: 1785619592) and `a (miserable:2.4) dog` (Seed: 1722099990). Unlike *cute*, the *miserable* modifier encodes both appearance *and* emotional expression simultaneously in SDXL's training distribution — heavy-lidded eyes, flattened ears, prostrate posture, and dark low-key lighting — making it a far stronger and more unambiguous generation signal. Both generated images confirm this visually, and CLIP scores it with high confidence: the cat scores *miserable* (0.55) and *sad* (0.95), the dog scores *sad* (0.97) and *miserable* (0.75), while all positive attributes — *cute*, *happy*, *joyful* — collapse near zero, confirming the modifier pushes the image cleanly into the negative emotional hemisphere of CLIP space. The single anomaly is *rich* scoring high in both cases (0.93 cat, 0.83 dog), which is not a scoring error but *style-attribute entanglement*: the *HDR, photorealistic, RAW photo* art style suffix produces studio-quality lighting that CLIP associates with a high-status appearance — independent of the emotional content — a systematic cross-layer interference the radar chart exposes precisely.

4 Conclusion and Discussion

This application demonstrated that controlled prompt parameterization, CLIP-based embedding analysis, and bipolar semantic scoring together form a powerful and generalizable framework for making text-to-image generation transparent and systematically interpretable. In **prompt engineering**, it replaces trial-and-error with quantitative feedback, allowing practitioners to benchmark which modifiers, strengths, and style combinations produce the most faithful semantic alignment. In **model interpretability research**, the bipolar scoring mechanism provides a lightweight, training-free probe into the semantic geometry of any diffusion model's conditioning space — applicable to future architectures beyond SDXL. In **Human-AI interaction** and creativity support, the UMAP cluster view offers an intuitive spatial map of the generative output space, helping non-expert users build intuition for how prompts influence results — addressing the core challenge identified by PromptMagician [1] and PrompTHis [2]. As a practical tool, it does not require model retraining, runs entirely locally, and integrates seamlessly into any Stable Diffusion workflow — making it immediately deployable for both research and production use cases in generative AI.

Acknowledgments

The author thanks the supervisors for their guidance throughout this work. Perplexity AI was used solely to support literature research and improve the readability of the manuscript; all technical content, implementation, and scientific interpretation remain the sole work of the author, who takes full responsibility for the accuracy of all content presented.

References

[1] Yingchaojie Feng et al. "PromptMagician: Interactive Prompt Engineering for Text-to-Image Creation". In: *IEEE Transactions on Visualization and Computer Graphics* 30.1 (2024), pp. 295–305. DOI: https://doi.org/10.1109/TVCG.2023.3327168.

[2] Yuhan Guo et al. "PrompTHis: Visualizing the Process and Influence of Prompt Editing during Text-to-Image Creation". In: *IEEE Transactions on Visualization and Computer Graphics* (2024). DOI: https://doi.org/10.48550/arXiv.2403.09615.

[3] Leland McInnes, John Healy, and James Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *arXiv preprint arXiv:1802.03426* (2018). DOI: https://doi.org/10.21105/joss.00861.

[4] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. 2021. DOI: https://doi.org/10.48550/arXiv.2103.00020.

[5] Robin Rombach et al. "High-Resolution Image Synthesis with Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022. DOI: https://doi.org/10.48550/arXiv.2112.10752.