# Report: Analogy-Based Image Generation Evaluation with CLIP Margin

The goal of this project was to test whether an analogy-style transformation can reliably steer a text-to-image model from a source concept AAA to a target concept TTT. A typical example is changing “a red car” into “a blue car.” The core question was straightforward: after applying the analogy method, do the generated images become closer to the target prompt than to the source prompt?

A major part of the project was designing an evaluation that would be meaningful, not anecdotal. Instead of judging a handful of examples, I wanted enough samples to make statistically defensible claims and also to see whether performance changes across different types of transformations. That is why I created **8 semantic categories** and used **50 word pairs per category**, which results in **400 word pairs** in total. This number was chosen deliberately: it is large enough to support statistical testing of whether the method works “in general,” while also allowing **category-wise analysis** (50 items per category) to reveal how the method behaves in different domains. In other words, the dataset size was not arbitrary — it was intended to enable **statistical significance checks** and to compare behavior across categories rather than reporting a single averaged score that could hide failures.

For generation, I used **segmind/SSD-1B**, a smaller SDXL-style model. This was mainly a practical decision based on my hardware: everything ran on an **RTX 4060 with 12GB VRAM**, and a smaller model allowed me to run the full experimental grid efficiently. For each word pair, I generated outputs with **three random seeds** and tested **three alpha values** that control blending strength: **0.5, 0.8, and 1.0**. That yields 400×3×3=3600400 \times 3 \times 3 = 3600400×3×3=3600 evaluated analogy images.

To evaluate whether the transformation actually moved outputs toward the target concept, I used a CLIP-based metric. For each generated image xxx, I computed a **CLIP margin** m=sim(x,T)−sim(x,A)m = \text{sim}(x, T) - \text{sim}(x, A)m=sim(x,T)−sim(x,A), where sim\text{sim}sim is cosine similarity between L2-normalized CLIP embeddings (image embedding compared to text embedding). This margin is easy to interpret: a positive margin means the image is closer to the target prompt than the source prompt according to CLIP, while a negative margin means it stayed closer to the source. Because the three seeds are repeated trials of the same underlying item rather than independent samples, I treated the unit of analysis as a **(pair, alpha)** item by averaging the margin over the three seeds before computing overall statistics.

Since the goal was not just “does it look okay?” but “does it work in a measurable way?”, I included an explicit statistical requirement. To claim that the method works overall, the mean margin should be **significantly greater than zero**, which corresponds to testing H0:μm=0H_0:\mu_m = 0H0:μm=0 versus H1:μm>0H_1:\mu_m > 0H1:μm>0. I reported the mean margin with a **95% bootstrap confidence interval** and also ran a one-sided one-sample t-test. If the confidence interval crosses zero (or the p-value is above 0.05), then there is no statistically strong evidence that the method consistently moves images toward the target concept.

The evaluation was efficient: scoring the 3600 images took about **54 seconds** on the RTX 4060, which is roughly **66.7 images per second**.

The overall results show that the method does **not** reliably succeed across the full dataset. The mean margin per (pair, alpha) item was approximately **−0.00014**, and the 95% bootstrap confidence interval included zero [−0.00198,0.00173][−0.00198, 0.00173][−0.00198,0.00173]. The one-sided t-test for μm>0\mu_m > 0μm>0 was not significant (p ≈ **0.56**). This means the method does not meet the statistical significance requirement for claiming overall success. The success rate also supports this: only about **40.6%** of items achieved positive margin, so outputs were more often closer to the source than to the target.

At the same time, alpha clearly behaves like a meaningful control knob. Increasing alpha produced a consistent improvement: alpha 0.5 tended to be negative, alpha 0.8 moved closer to zero, and alpha 1.0 performed best overall. This suggests the blending strength is pushing in the intended direction, even though the method still fails to generalize reliably across all cases.

The most important insight came from the category-wise breakdown, which was a major reason for using 8×50 items in the first place. Performance is not uniform across domains. At alpha 1.0, **category 8** achieved a mean margin of about **0.069** with a success rate of **1.0**, and **category 5** also performed strongly (mean margin **0.0466**, success rate **1.0**). Categories 3 and 4 were also clearly positive with high success rates (about **0.83** and **0.77**). In contrast, categories 1, 6, and 7 remained negative even at alpha 1.0, with low success rates around **0.24–0.31**, meaning the analogy transformation often failed to move images toward the target prompt in those domains. Category 2 was borderline, with mean margin near zero and success rate close to chance.

In summary, this project shows that analogy-based steering can work very well in some semantic categories, but it is not robust across a diverse dataset. The full-sample result is not statistically significant, but the category-level analysis reveals that the method is not random — it is **condition-dependent**. This outcome justifies the original design choice of using **400 word pairs** across **8 categories**: it made it possible to test significance at scale and to identify where the method succeeds versus where it systematically breaks down.