# PROBING CROSS-BRANCH COMPATIBILITY IN UNIFIED MULTIMODAL MODELS THROUGH SEMANTIC STEERING

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

## ABSTRACT

Unified multimodal models (UMMs) aim to integrate understanding and generation within a single architecture, yet these two capabilities are still largely evaluated in isolation. Strong performance on both tasks therefore tells us little about how their internal semantic representations relate to each other. A fundamental question is whether the semantic structures underlying understanding and generation are compatible across branches. To study this question, we introduce *cross-branch semantic steering*, an intervention-based framework that extracts semantic directions from one branch and applies them to the other. We show that steering vectors learned from the understanding branch can transfer to generation, enabling controllable image synthesis and improved semantic faithfulness. In contrast, the reverse direction consistently shows limited effectiveness. Our analysis suggests that this asymmetry may be related to a practical representational mismatch: understanding-derived vectors capture transferable, object-centric semantics, while generation-derived vectors primarily encode low-level appearance features. Our results reveal that architectural unification does not guarantee semantic alignment, and establish cross-branch steering as a practical tool for probing multimodal representations.

## 1 INTRODUCTION

Unified Multimodal Models (UMMs) integrate multimodal understanding and generation within a single architecture, and have emerged as a pivotal step toward artificial general intelligence (Deng et al., 2025; Chen et al., 2025b; Wu et al., 2025a; Wang et al., 2025c; Zhang et al., 2025a; Lin et al., 2025; Xie et al., 2025b; Zhao et al., 2026b; Qu et al., 2025; Xie et al., 2025a; Wu et al., 2025d;c; Luo et al., 2026; Su et al., 2026; Yan et al., 2026; Sun et al., 2026; Liu et al., 2025b; Wu et al., 2026; Jiao et al., 2025; Zheng et al., 2025). These models maintain two functional branches through a shared backbone: an *understanding branch* that encodes visual and textual inputs into semantic representations for tasks such as visual question answering and captioning, and a *generation branch* that synthesizes images from latent representations conditioned on text. While recent UMMs have demonstrated strong performance across both understanding and generation tasks, these capabilities are typically evaluated independently (Deng et al., 2025; Chen et al., 2025b; Xie et al., 2025b; Luo et al., 2026). Existing evaluations establish that a single model can perform both capabilities well, but leave a more fundamental representation-level question unanswered: *Does architectural unification also give rise to compatible semantic structure across the understanding and generation branches?*

The question is non-trivial to answer, as the two branches, despite sharing the same backbone, are optimized for different objectives and place different demands on their internal representations (Song et al., 2026; Wang et al., 2026). Understanding primarily relies on high-level semantic representations that capture objects, attributes, and relations, whereas generation additionally requires fine-grained visual information, such as appearance, texture, and spatial details, to synthesize images faithfully (Song et al., 2026; Wang et al., 2026). These different representational demands may lead the two branches to organize semantic information differently, even within the same model.

In this paper, we introduce cross-branch semantic steering as a diagnostic framework to probe this question. Rather than comparing representations directly, we employ an intervention-based test: extract a steering vector encoding a target semantic direction (e.g., "red → pink") from one branch



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

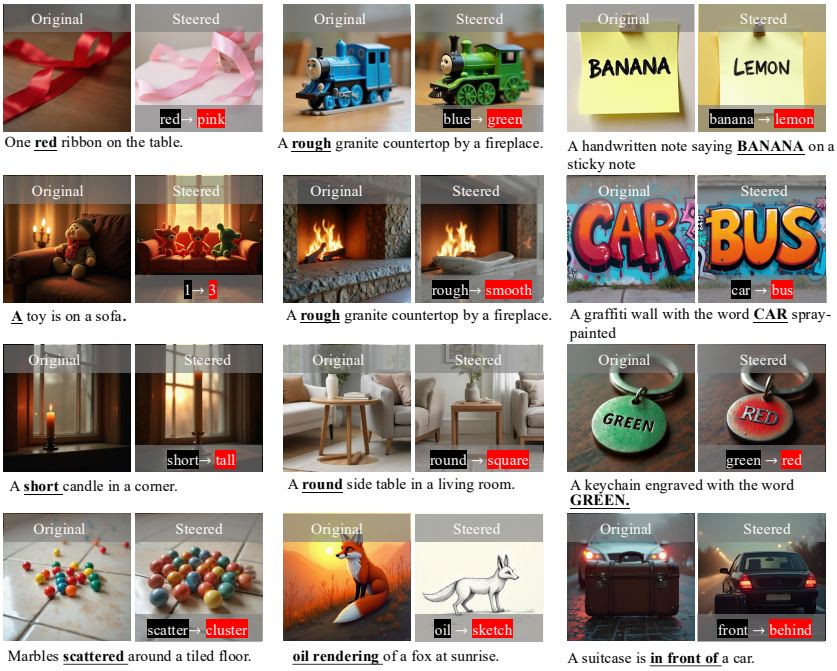


Figure 1: Qualitative results of understanding → generation steering across semantic types. Each pair shows the original generation (left) and the steered output (right). Steering modifies the target attribute while preserving object identity. More results are provided in the appendix D.1.

and inject it into the other during inference. If the vector learned from the understanding branch causes the generation branch to produce a pink object instead of a red one, this provides causal evidence that the extracted semantic direction from understanding is functionally compatible with the generation branch. While steering has been widely used to probe LLM internals (Panickssery et al., 2024; Zou et al., 2025; Li et al., 2024), prior work operates within a single modality. Whether steering vectors can transfer *across* modality branches—where token types, training signals, and representational structures differ—is an open question.

First, we show that steering vectors learned from the understanding branch can transfer to the generation branch. We construct UMMSteer, a visually grounded contrastive dataset comprising 51 diverse concept pairs spanning color, text, counting, position, style, appearance, and persona. For each concept pair, we first extract a semantic steering vector from the understanding branch using text-based QA samples, and then apply this vector to the generation branch to test whether it can causally shift image generation. As illustrated in Fig. 1, with a vector encoding "red → pink", the UMM model produces a pink ribbon when prompted with "One red ribbon on the table." Notably, although the vector is learned from textual representations, its effect manifests in visual generation, demonstrating that understanding-derived semantic directions can remain effective across branches.

Meanwhile, we uncover a striking asymmetry: while understanding-to-generation transfer is feasible, the reverse direction consistently fails. Steering vectors derived from the generation branch do not meaningfully influence the understanding branch's predictions. Our analysis suggests that this asymmetry may stem from a representational mismatch: understanding-derived vectors tend to capture compositional, object-centric high-level semantics that align well with both branches, while generation-derived vectors primarily capture low-level appearance features—background tone, illumination, global color statistics—that show weaker alignment with the understanding branch's semantic structure. Cross-branch transfer succeeds only when the steering direction resides within a shared subspace defined by understanding-level abstractions. To verify the generalizability of our findings, we evaluate cross-branch steering on other state-of-the-art UMMs, including Janus-Pro (Chen et al., 2025b) and UniPic-1 (Wang et al., 2025c). Our analysis suggests that strong model performance does not necessarily imply aligned semantic representations across branches. Instead, such alignment depends on the underlying architectural design and modality integration. We summarize our main contributions as follows:

1. We introduce *cross-branch semantic steering*, the first framework to transfer steering vectors across understanding and generation branches in UMMs. We construct a visually grounded contrastive steering dataset for this purpose, which will be publicly released.
2. We uncover a *fundamental asymmetry in cross-branch transfer*: semantic directions learned from the understanding branch can control generation, while the reverse direction consistently fails.
3. We trace this asymmetry to a *representational mismatch between branches*, where understanding-derived vectors capture higher-level, object-centric semantics that align well across branches, and generation-derived vectors primarily reflect lower-level appearance features.

## 2 Related Work

**Unified Multimodal Models.** Unified multimodal models (UMMs) aim to build a single architecture capable of both understanding and generating data across multiple modalities (Lin et al., 2025; Li et al., 2025; Qin et al., 2026; Xiao et al., 2025; Chen et al., 2025a; Tong et al., 2024; Zhou et al., 2025; Sun et al., 2024). These models typically maintain two functional branches: an *understanding branch* that encodes visual and textual inputs into semantic representations for tasks such as visual question answering and captioning, and a *generation branch* that synthesizes images from latent representations conditioned on text. Although the two branches differ in their input encoders and output mechanisms, they typically share a common backbone. Existing UMMs fall into three broad paradigms. Pure autoregressive models serialize visual and textual tokens into a unified sequence and model them via next-token prediction, as in Janus-Pro (Chen et al., 2025b) and UniPic-1 (Wang et al., 2025c). Pure diffusion-based approaches perform both understanding and generation within a unified diffusion framework, as in UniModel (Zhang et al., 2025a). Hybrid models combine autoregressive text generation with diffusion- or flow-based image synthesis, using language modeling for semantic control and diffusion for high-fidelity visual generation, as in BAGEL (Deng et al., 2025) and Show-o2 (Xie et al., 2025b). While these models demonstrate strong performance on standard benchmarks, it remains unclear whether their unified architectures give rise to unified representations, and whether the understanding and generation branches share a common semantic space. Our work directly probes this important question through the lens of cross-branch steering.

**Steering and Representation Intervention in LLMs.** Steering methods modify model behavior at inference time by adding learned directions to hidden states. A central family of approaches derives steering vectors from contrastive representations: mean differences (CAA) (Panickssery et al., 2024), principal components of difference vectors (Zou et al., 2025), linear classifiers trained on hidden states (Li et al., 2024), and paired contrastive prompts (Turner et al., 2024). Extensions include input-adaptive scaling (Lee et al., 2025) and layer-specific multi-behavior control (van der Weij et al., 2024). A broader line of work on representation probing and intervention (Xu et al., 2025; Wang et al., 2025a; Zhao et al., 2026a; Fang et al., 2026; Sinii et al., 2025; Zhao et al., 2025b; Park et al., 2025; Patel et al., 2024; Li et al., 2026) has established steering as a reliable tool for analyzing and controlling LLM internals. However, all existing steering work operates within a single modality—typically text-only LLMs or, more recently, multimodal language models in their understanding capacity (Wang et al., 2025b; Parekh et al., 2025; Wu et al., 2025b; Gan et al., 2025; Liu et al., 2025a; Zhao et al., 2025a; Zhang et al., 2025b). No prior work has attempted to transfer steering vectors across understanding and generation branches in UMMs. Our work is the first to do so, repurposing steering not as a control mechanism but as a diagnostic probe for cross-modal semantic alignment in UMMs.

## 3 Cross-Branch Steering as a Diagnostic Framework

We aim to determine whether the understanding and generation branches of UMMs share a unified semantic space. Directly comparing representations across branches is difficult, as they differ in both token types and optimization objectives. To sidestep the challenge, we propose an intervention-based perspective: if a semantic direction extracted from one branch can reliably influence the behavior of the other, this constitutes evidence that both branches encode compatible semantic structure.

Steering provides a natural yet unexplored lens for this purpose. In LLMs, steering vectors have been shown to capture interpretable semantic directions in hidden space, enabling controlled shifts in model behavior (Panickssery et al., 2024; Zou et al., 2025; Li et al., 2024). While prior steering work operates within a single modality, the cross-branch setting in UMMs poses a distinct challenge: the two branches process different token types, serve different objectives, and may organize semantics at different levels of abstraction. *Whether steering vectors can cross this boundary is an open question*. We emphasize that steering here serves as a diagnostic probe for semantic alignment, not as a proposed method for controlling generation.

This section describes the setup: the contrastive dataset UMMSteer we construct (Sec. 3.1), and how steering vectors are extracted based on the dataset (Sec. 3.2). The results of applying this framework will be presented in Sec. 4.

### 3.1 UMMSteer: Contrastive Steering Dataset

A key challenge in operationalizing cross-branch steering is the lack of suitable data. Existing steering datasets are not directly applicable, since steering in UMMs requires concepts that are both semantically meaningful for the understanding branch and visually verifiable in the generation branch. We therefore begin by constructing UMMSteer, a contrastive dataset tailored for UMMs.

**Semantic Concept for Steering.** To comprehensively cover the semantic space, we define 7 semantic types spanning 3 levels of increasing complexity, yielding **51 pairs of semantic concepts** in total. At the object level, we include intrinsic properties of individual objects: `color` (e.g., red vs. blue), `text` (e.g., "23" vs. "10"), and `counting` (e.g., 1 vs. 4). At the relational level, we include spatial relationships between objects: `position` (e.g., left/right, above/below, inside/outside). At the scene level, we include global appearance and holistic properties: `style` (e.g., realistic vs. cartoon), `appearance` (e.g., clean vs. dirty, wooden vs. metal), and `persona` (e.g., neurotic vs. calm). We summarize the complete list of semantic concepts in Appendix A.2 (Tab. 3).

**Definition 1 (Cross-Branch Steering Dataset).** *For a target concept $k$, we construct a dataset of $N$ contrastive pairs $\mathcal{D}^k = \{(x_i^+, x_i^-)\}_{i=1}^N$, where $x_i^+ = (q_i^+, y_i^+)$ and $x_i^- = (q_i^-, y_i^-)$ denote the positive and negative instances, respectively. Here, $q_i$ is the input context (a question for understanding, or a text prompt for generation) and $y_i$ is the expected output (an answer token or a generated image). Each pair differs only in the target semantic attribute while all other contextual factors remain fixed.*

Each pair is constructed in two branch-specific formats. We provide representative examples in Appendix (Fig. 7). Specifically, for the understanding branch, we construct contrastive question–answer pairs where the questions describe identical scenes except for the target attribute, e.g.:

$x_i^+$ : "A **blue** car parked on a street. What color is this car?" $\rightarrow$ **blue**

$x_i^-$ : "A **red** car parked on a street. What color is this car?" $\rightarrow$ **red**

For the generation branch, we construct contrastive text prompts that differ only in the target attribute (e.g., "A blue car parked on a street" vs. "A red car parked on a street"). For each binary concept, we construct 50 pairs per format for learning and 50 held-out samples for evaluation.

### 3.2 Learning Steering Vectors for Cross-Branch Transfer

A steering vector represents an additive direction in the hidden-state space that can bias a model toward a target semantic concept (Panickssery et al., 2024; Zou et al., 2025; Li et al., 2024). In our setting, the goal is to learn such semantic directions from one branch of a UMM and examine whether applying them to the other branch can steer its behavior toward the same target concept. Here, we briefly describe how we learn steering vectors using representative LLM steering methods and how we apply them for cross-branch transfer.

As shown in Fig. 2 (a), for a target concept $k$, we feed contrastive pairs from $\mathcal{D}^k$ into the source branch and extract residual stream activations at the answer token position (for understanding) or all VAE token positions (for generation), yielding pairs $\{\mathbf{h}_{+,i}^{(l)}, \mathbf{h}_{-,i}^{(l)}\}$. We estimate $\mathbf{v}^{(l)}$ using three common representative methods: CAA (Panickssery et al., 2024) (mean difference between positive

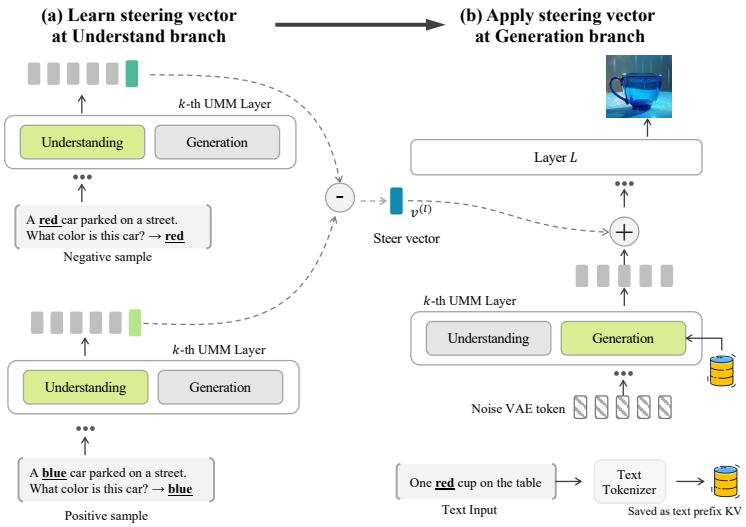


Figure 2: **Cross-branch steering pipeline (Und → Gen).** (**a**) Contrastive text pairs (e.g., `'blue'` vs. `'red'`) are fed through the understanding branch. At each layer, we extract residual stream activations from the answer token and compute steering vectors. (**b**) At inference time, the input text prompt (e.g., `'one red cup'`) is encoded into a prefix KV cache. During denoising, steering vectors are added to VAE token representations at each layer, biasing generation toward the target attribute.

and negative activations), RepE (Zou et al., 2025) (first principal component of the difference vectors), and ITI (Li et al., 2024) (normal vector of a learned decision boundary). All three recover a semantic direction along which model behavior can be shifted. Formal definitions are in Appendix A.4.

## 4 Do the Branches Share a Unified Semantic Space?

We now apply the cross-branch steering diagnostic framework to answer our central question. We first test whether steering vectors from the understanding branch can control generation (Sec. 4.1), then examine the reverse direction (Sec. 4.2), and finally analyze why transfer is asymmetric (Sec. 5).

### 4.1 Understanding → Generation

**Applying learned steer vectors to the generation branch.** After obtaining layer-wise steer vectors from the understanding branch, we transfer them to the generation branch at inference time to bias the denoising trajectory toward the desired semantic attribute, as shown in Fig 2(b). Given an input prompt such as "`one red cup on the table`", the text tokens are first encoded by the text tokenizer and processed by the model to produce a text prefix key-value cache. This prefix KV cache serves as the textual conditioning context for image generation. The generation process starts from noisy VAE latent tokens, which are iteratively updated through the transformer and finally decoded by the VAE decoder into an image.

During generation, at each transformer layer $l$, the VAE latent tokens attend to the text prefix KV cache through multimodal self-attention. We intervene on the hidden representation of the VAE tokens by adding the pre-computed steer vector $\mathbf{v}^{(l)}$:

$$\tilde{\mathbf{H}}_t^{(l)} = \mathbf{H}_t^{(l)} + \alpha \mathbf{v}^{(l)}, \tag{1}$$

where $\mathbf{H}_t^{(l)}$ denotes the hidden states of $t$-th VAE token at layer $l$, $\mathbf{v}^{(l)}$ is the steer vector learned from the understanding branch, and $\alpha$ is the steering strength. In practice, the same steer vector is added to all VAE token representations at the selected layer, biasing the denoising trajectory toward the target semantic direction.

Table 1: **Steering success rate (SSR) across diverse semantic categories** in BAGEL. SSR measures the fraction of generated images that both preserve the target object and correctly realize the steered attribute. Values are mean ± std across concept pairs within each category.

| | Object-level | | | Relational | Scene-level | | |
|---|---|---|---|---|---|---|---|
| Method | Color | Text | Counting | Position | Style | Appearance | Persona |
| CAA | $96.20_{\pm 1.46}$ | $\mathbf{43.46}_{\pm 28.79}$ | $\mathbf{68.75}_{\pm 23.96}$ | $78.60_{\pm 16.57}$ | $\mathbf{82.71}_{\pm 24.57}$ | $\mathbf{79.88}_{\pm 21.06}$ | $24.37_{\pm 31.25}$ |
| RepE | $52.95_{\pm 40.63}$ | $21.13_{\pm 22.50}$ | $6.87_{\pm 7.79}$ | $59.83_{\pm 21.74}$ | $29.29_{\pm 26.61}$ | $44.25_{\pm 23.95}$ | $\mathbf{34.00}_{\pm 33.94}$ |
| ITI | $\mathbf{96.48}_{\pm 1.66}$ | $43.09_{\pm 29.06}$ | $34.90_{\pm 28.35}$ | $\mathbf{78.96}_{\pm 17.32}$ | $80.75_{\pm 21.70}$ | $78.86_{\pm 21.80}$ | $29.38_{\pm 30.58}$ |

**Evaluation Setup.** To evaluate steering effectiveness in the generation branch, we construct 50 held-out text prompts for each semantic concept. The model performs text-to-image generation under steering, and the resulting images are evaluated for semantic correctness. We adopt a question-based evaluation protocol using Qwen-VL-3 30B (Bai et al., 2025) as the evaluator (Niu et al., 2025; Ghosh et al., 2023a). The evaluation consists of two stages. First, we verify whether the generated image preserves the target object, ensuring that steering does not alter object identity. Second, we assess whether the desired semantic attribute (e.g., spatial relation or color) is correctly enforced. A sample is considered successfully steered only if both conditions are satisfied. We report the *steering success rate* (SSR) as the fraction of samples that meet both criteria. More details are provided in Appendix A.5. We further validate this automatic evaluation protocol through a human agreement study in Appendix B.1, where Qwen-VL-3 30B achieves high agreement with human annotations, with an overall Cohen's $\kappa$ of 0.894.




Figure 3: **The steering success rates (SSR) across multiple semantic categories.** The effectiveness of CAA-based steering varies within object-level, relational-level, and scene-level semantics. Full per-concept results are provided in Appendix B.2.

**Steering Effectiveness.** We present the steering success rates across all semantic categories based on BAGEL (Deng et al., 2025) in Tab. 1. Across all three steering methods, understanding-derived vectors can control the generation. However, the effectiveness is not uniform across concepts. We observe substantial within-category variance in Tab. 1, and therefore further present the SSR of representative concepts using CAA (Panickssery et al., 2024) in Fig. 3. As shown in Fig. 3, effectiveness varies within categories and correlates with generation difficulty: simpler concepts (e.g., `one→two` in counting category) achieve higher success rates than harder ones (e.g., `one→four`), suggesting that steering is constrained by the model's baseline generative capability.

Qualitative examples in Fig. 1 further illustrate successful steering across diverse concepts, including color, counting, text, appearance, and spatial relations. We provide additional qualitative examples in Appendix D.1 and representative failure cases in Appendix D.2.

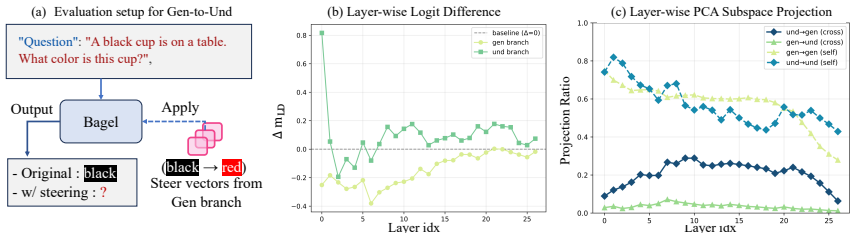


Figure 4: **Analysis of cross-branch (Gen → Und) steering failure. (a) Setup.** Steering vectors learned from the generation branch (e.g., `blue` vs. `red`) are applied to the understanding branch, and we evaluate whether QA predictions can be shifted. **(b) Layer-wise logit difference.** Here, $\Delta m_{\mathrm{LD}}$ quantifies the effect of steering vectors on model output logits. Understanding-derived vectors consistently shift model preference toward the target attribute, whereas generation-derived vectors show negligible or unstable effects. **(c) Layer-wise subspace alignment analysis.** We measure alignment between steering vectors and the branch's semantic subspace using the projection ratio (PSP), where higher values indicate better alignment. Generation-derived vectors exhibit low alignment with the understanding semantic subspace under gen → und, indicating a semantic mismatch. Here, $A \to B$ denotes learning on $A$ and applying to $B$ (self if $A = B$, cross otherwise).

### 4.2 Generation → Understanding

In Sec. 4.1, we showed that steering vectors learned from the understanding branch can be transferred to the generation branch, enabling controllable image synthesis. In this section, we show that the reverse direction—from generation to understanding—does not work.

Similar to Sec. 3, we use semantic concepts constructed from our dataset UMMSteer to study cross-branch transfer from generation branch into understand branch. In this setting, as shown in Fig. 4(a), contrastive pairs are formed using text prompts, *e.g.*, $x_i^+$ : "`A blue car parked on a street.`" and $x_i^-$ : "`A red car parked on a street.`". Using these contrastive pairs, we learn steering vectors from the generation branch based on VAE token representations and apply them to the understanding branch in a QA setting. More details are provided in Appendix A.3.

**Generation Cannot Steer Understanding.** We first directly evaluate whether these vectors can change the model's textual predictions across multiple semantic concepts. As shown in Appendix B.3, the model's outputs remain unchanged under generation-derived steering, indicating that these vectors fail to induce meaningful changes in the model's text outputs. To further examine this failure at a finer-grained logit level, we adopt the logit-difference metric $m_{\mathrm{LD}}$, which measures the model's preference between a target answer and its contrastive alternative (Tan et al., 2024). Specifically,

$$m_{\mathrm{LD}} = \mathrm{Logit}(\text{"}\texttt{positive}\text{"}) - \mathrm{Logit}(\text{"}\texttt{negative}\text{"}), \tag{2}$$

where "`positive`" and "`negative`" tokens correspond to the target and contrastive concepts, respectively. Then, we compare the change in $m_{\mathrm{LD}}$ with the steered model against the baseline (unsteered) model to quantify the effect of steering:

$$\Delta m_{\mathrm{LD}} = m_{\mathrm{LD}}^{\mathrm{steered}} - m_{\mathrm{LD}}^{\mathrm{baseline}}. \tag{3}$$

A positive $\Delta m_{\mathrm{LD}}$ indicates that steering shifts the model toward the target concept, while values near zero suggest negligible behavioral change. A negative $\Delta m_{\mathrm{LD}}$ indicates that steering shifts the model toward the contrastive (opposite) concept.

In Fig. 4(b), we report the layer-wise $\Delta m_{\mathrm{LD}}$ on the '`red`' vs. '`blue`' semantic concept, comparing steering vectors learned from the understanding branch and the generation branch under the same training data and steering scale. We observe that steering vectors from understanding branch consistently increase $\Delta m_{\mathrm{LD}}$, indicating effective control over the model's output preference. In contrast, vectors from the generation branch yield near-zero or even negative $\Delta m_{\mathrm{LD}}$ across layers and exhibit unstable behavior, suggesting a failure to induce meaningful changes in the output logits. This indicates that generation-derived steering cannot reliably shift the model toward the desired semantic direction in the understanding branch. To further verify that this phenomenon is not limited to a single semantic concept, we provide additional results across more semantic concepts in Appendix C.3.

## 5 WHY IS TRANSFER ASYMMETRIC?

The results above reveal a clear asymmetry: steering succeeds from understanding to generation, but fails in the reverse direction. We now investigate *why* this asymmetry arises through two complementary analyses.

**Subspace Alignment Analysis.** We hypothesize that semantic directions learned in the generation branch may not align with the semantic structure used by the understanding branch. To verify this, we introduce a PCA Subspace Projection (PSP) metric that measures how well a steering vector aligns with a branch's dominant semantic directions (see Appendix C.4 for details). Intuitively, a higher PSP indicates that the steering vector follows the model's natural semantic structure, while a lower value suggests that it lies outside the space of meaningful semantics for that branch. We compute the PSP for the `red–blue` semantic direction in both self-branch and cross-branch settings. As shown in Fig. 4(c), steering vectors from the understanding branch maintain high alignment when transferred to the generation branch, suggesting they capture generalized semantic directions. Conversely, generative steering vectors align poorly with the understanding branch, indicating their directions are not recognized as meaningful semantic variations. This discrepancy accounts for the observed asymmetry in cross-branch transferability.

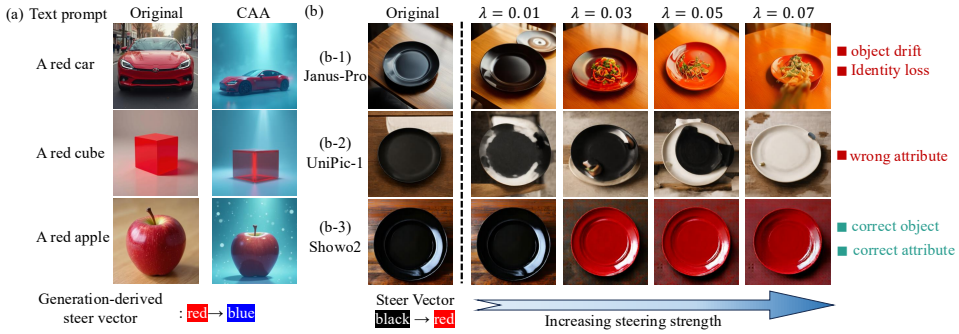


Figure 5: **(a) Generation-derived vectors fail even in self-branch steering.** Steering vectors learned from the generation branch (red → blue) are applied back to the generation branch. Vectors instead alter global appearance (background, illumination) without changing the object. **(b) Generalization across UMM architectures.** Cross-branch steering fails in pure autoregressive structures (Janus-Pro (Chen et al., 2025b), UniPic-1 (Wang et al., 2025c)), but succeeds in hybrid AR+diffusion models (Show-o2 (Xie et al., 2025b)), highlighting the role of architecture.

**What do generation-derived vectors encode?** To examine what information is encoded in steering vectors learned from the generation branch, we apply them back to the generation process in a self-branch setting. As shown in Fig. 5(a), the target semantic corresponds to object color (red → blue). While the intervention induces noticeable visual changes, it fails to change the object color to the target attribute. Instead, it primarily affects global appearance properties, such as background tone, illumination, and overall color distribution. This phenomenon suggests that generation-derived steering vectors mainly capture low-level, appearance-oriented features rather than fine-grained, object-level semantics. Consequently, they are unable to reliably steer the image toward the desired semantic target, which also explains their ineffectiveness in cross-branch transfer.

## 6 FURTHER DISCUSSION

**Implication on UMM Architecture Design.** We further examine whether cross-branch steering generalizes across UMM architectures. Specifically, we evaluate two representative pure autoregressive (AR) UMMs, Janus-Pro (Chen et al., 2025b) and UniPic-1 (Wang et al., 2025c), together with a hybrid AR+diffusion model, Show-o2 (Xie et al., 2025b). For all models, we use CAA (Panickssery et al., 2024) to extract steering vectors from the understanding branch and follow the same evaluation protocol as for BAGEL (Deng et al., 2025). For reference, we additionally include generation (GenEval (Ghosh et al., 2023b), DPG-Bench (Hu et al., 2024)) and understanding (MMMU (Yue et al., 2024), MM-Vet (Yu et al., 2024)) benchmark results reported in the original papers.

Table 2: Und→Gen steering success rate (SSR, %) across UMM architectures, alongside standard generation and understanding benchmarks reported in the original papers. Stronger standalone generation or understanding performance does not necessarily imply stronger cross-branch steerability. Some benchmark results are omitted as they were not reported in the original papers.

| | | Und→Gen Steering SSR ↑ (%) | | | | | Generation↑ | | Understanding↑ | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **Architecture** | **Color** | **Count.** | **Pos.** | **Style** | **Avg.** | **GenEval** | **DPG** | **MMMU** | **MM-Vet** |
| BAGEL | Hybrid AR+diffusion | **96.20** | **68.75** | **78.60** | **82.71** | **81.57** | 0.82 | 85.1 | 55.3 | 67.2 |
| Show-o2 | Hybrid AR+diffusion | 87.13 | 43.07 | 32.20 | 64.20 | 56.65 | 0.73 | **86.1** | 37.1 | – |
| Janus-Pro | Pure AR | 48.75 | 20.57 | 8.67 | 47.80 | 31.45 | 0.80 | 84.2 | 36.3 | 39.8 |
| UniPic-1 | Pure AR | 2.30 | 13.07 | 13.40 | 37.80 | 16.64 | **0.86** | 85.5 | – | – |

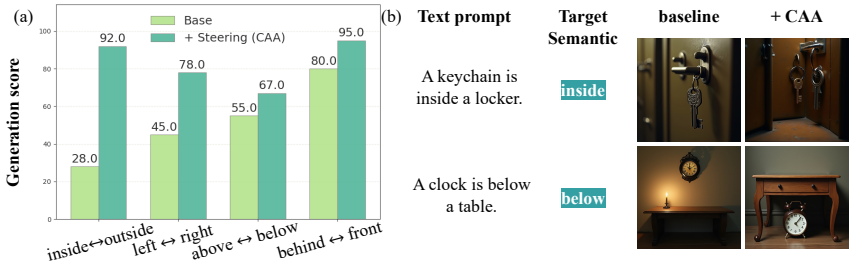


Figure 6: **Understanding-derived steering reduces generation hallucination.** (a) Quantitative comparison shows that steering significantly improves generation on relational semantics. (b) Qualitative examples show corrected spatial configurations (e.g., inside, below) that the baseline model hallucinates.

The quantitative results are reported in Table 2. BAGEL achieves the highest SSR across all four semantic categories, while Show-o2 also exhibits substantial Und→Gen steering effects. In contrast, Janus-Pro and UniPic-1 achieve substantially lower SSR, indicating limited cross-branch steering under our evaluation setup. Importantly, this trend does not consistently track the models' standalone generation or understanding performance.

To further rule out sensitivity to the choice of steering strength, we sweep a dense range of intervention strengths and visualize representative examples in Fig. 5(b). Increasing the steering strength in Janus-Pro primarily causes object drift and identity degradation, while UniPic-1 likewise fails to reliably induce the target attribute. In contrast, Show-o2 exhibits the desired semantic change over a broad range of steering strengths. Together, these results suggest that cross-branch steerability depends strongly on the underlying UMM architecture and the coupling between understanding and generation representations, rather than simply on standalone model performance.

**Application: Reducing Generation Hallucination in UMMs.** Here, we also explore whether understanding-derived steering vectors can mitigate image generation hallucinations, particularly on complex relational semantics where errors are most prevalent. We evaluate cross-branch steering on relational concepts (e.g., inside/outside, above/below) and report the *generation score*, defined as the fraction of generated images that both contain the target object and satisfy the specified spatial relation. As shown in Fig. 6(a), CAA-based steering consistently improves generation scores across all relational concepts. Qualitative results in Fig. 6(b) further illustrate that steering enforces the desired relations, correcting common errors like incorrect spatial configurations.

## 7 Conclusion

In this work, we investigate the representational compatibility between multimodal understanding and generation in UMMs through cross-branch semantic steering. We find that semantic directions learned from the understanding branch transfer effectively to generation, enabling controllable and faithful image synthesis. In contrast, generation-derived directions fail to influence understanding, revealing a fundamental asymmetry. Our analysis traces this asymmetry to a representational mismatch: understanding encodes object-centric semantics, while generation primarily captures low-level appearance features. These findings suggest that semantic alignment in UMMs depends critically on representation structure rather than model capability alone. Limitations are discussed in Appendix F.

## REFERENCES


Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset, 2025a. URL https://arxiv.org/abs/2505.09568.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025b.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025. URL https://arxiv.org/abs/2505.14683.

Yi Fang, Wenjie Wang, Mingfeng Xue, Boyi Deng, Fengli Xu, Dayiheng Liu, and Fuli Feng. Controllable llm reasoning via sparse autoencoder-based steering. *arXiv preprint arXiv:2601.03595*, 2026.

Woody Haosheng Gan, Deqing Fu, Julian Asilis, Ollie Liu, Dani Yogatama, Vatsal Sharan, Robin Jia, and Willie Neiswanger. Textual steering vectors can improve visual understanding in multimodal large language models. *arXiv preprint arXiv:2505.14071*, 2025.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment, 2023a. URL https://arxiv.org/abs/2310.11513.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment, 2023b. URL https://arxiv.org/abs/2310.11513.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment, 2024. URL https://arxiv.org/abs/2403.05135.

Yang Jiao, Haibo Qiu, Zequn Jie, Shaoxiang Chen, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. Unitoken: Harmonizing multimodal understanding and generation through unified visual encoding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3600–3610, 2025.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering, 2025. URL https://arxiv.org/abs/2409.05907.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model, 2024. URL https://arxiv.org/abs/2306.03341.

Teng Li, Quanfeng Lu, Lirui Zhao, Hao Li, Xizhou Zhu, Yu Qiao, Jun Zhang, and Wenqi Shao. Unifork: Exploring modality alignment for unified multimodal understanding and generation. *arXiv preprint arXiv:2506.17202*, 2025.

Yawei Li, Benjamin Bergner, Yinghan Zhao, Vihang Prakash Patil, Bei Chen, and Cheng Wang. Steering large reasoning models towards concise reasoning via flow matching, 2026. URL https://arxiv.org/abs/2602.05539.

Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, Yatian Pang, and Li Yuan. Uniworld-v1: High-resolution semantic encoders for unified visual understanding and generation, 2025. URL https://arxiv.org/abs/2506.03147.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

Zheyuan Liu, Zhangchen Xu, Guangyao Dou, Xiangchi Yuan, Zhaoxuan Tan, Radha Poovendran, and Meng Jiang. Steering multimodal large language models decoding for context-aware safety. *arXiv preprint arXiv:2509.19212*, 2025a.

Zhiheng Liu, Weiming Ren, Haozhe Liu, Zijian Zhou, Shoufa Chen, Haonan Qiu, Xiaoke Huang, Zhaochong An, Fanny Yang, Aditya Patel, Viktar Atliha, Tony Ng, Xiao Han, Chuyan Zhu, Chenyang Zhang, Ding Liu, Juan-Manuel Perez-Rua, Sen He, Jürgen Schmidhuber, Wenhu Chen, Ping Luo, Wei Liu, Tao Xiang, Jonas Schult, and Yuren Cong. Tuna: Taming unified visual representations for native unified multimodal models, 2025b. URL https://arxiv.org/abs/2512.02014.

Yinyi Luo, Wenwen Wang, Hayes Bai, Hongyu Zhu, Hao Chen, Pan He, Marios Savvides, Sharon Li, and Jindong Wang. Torchumm: A unified multimodal model codebase for evaluation, analysis, and post-training, 2026. URL https://arxiv.org/abs/2604.10784.

Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Kunpeng Ning, Chaoran Feng, Bin Zhu, and Li Yuan. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition, 2024. URL https://arxiv.org/abs/2312.06681.

Jayneel Parekh, Pegah Khayatan, Mustafa Shukor, Arnaud Dapogny, Alasdair Newson, and Matthieu Cord. Learning to steer: Input-dependent steering for multimodal llms. *arXiv preprint arXiv:2508.12815*, 2025.

Sunghyun Park, Seokeon Choi, Hyoungwoo Park, and Sungrack Yun. Steering guidance for personalized text-to-image diffusion models, 2025. URL https://arxiv.org/abs/2508.00319.

Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. Steering rectified flow models in the vector field for controlled image generation. *arXiv preprint arXiv:2412.00100*, 2024.

Luozheng Qin, Jia Gong, Yuqing Sun, Tianjiao Li, Mengping Yang, Xiaomeng Yang, Chao Qu, Zhiyu Tan, and Hao Li. Uni-cot: Towards unified chain-of-thought reasoning across text and vision, 2026. URL https://arxiv.org/abs/2508.05606.

Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2545–2555, 2025.

Viacheslav Sinii, Alexey Gorbatovski, Artem Cherepanov, Boris Shaposhnikov, Nikita Balagansky, and Daniil Gavrilov. Steering llm reasoning through bias-only adaptation, 2025. URL https://arxiv.org/abs/2505.18706.

Wei Song, Yuran Wang, Zijia Song, Yadong Li, Zenan Zhou, Long Chen, Jianhua Xu, Jiaqi Wang, and Kaicheng Yu. Dualtoken: Towards unifying visual understanding and generation with dual visual vocabularies, 2026. URL https://arxiv.org/abs/2503.14324.

Zhaolong Su, Wang Lu, Hao Chen, Sharon Li, and Jindong Wang. Unigame: Turning a unified multimodal model into its own adversary, 2026. URL https://arxiv.org/abs/2511.19413.

Peng Sun, Jun Xie, and Tao Lin. Rethinking umm visual generation: Masked modeling for efficient image-only pre-training, 2026. URL https://arxiv.org/abs/2603.16139.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners, 2024. URL https://arxiv.org/abs/2312.13286.

Daniel Tan, David Chanin, Aengus Lynch, Dimitrios Kanoulas, Brooks Paige, Adria Garriga-Alonso, and Robert Kirk. Analyzing the generalization and reliability of steering vectors, 2024. URL https://arxiv.org/abs/2407.12404.

Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning, 2024. URL https://arxiv.org/abs/2412.14164.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248.

Teun van der Weij, Massimo Poesio, and Nandi Schoots. Extending activation steering to broad skills and multiple behaviours, 2024. URL https://arxiv.org/abs/2403.05767.

Anyi Wang, Xuansheng Wu, Dong Shu, Yunpu Ma, and Ninghao Liu. Enhancing llm steering through sparse autoencoder-based vector refinement. *arXiv preprint arXiv:2509.23799*, 2025a.

Chenlong Wang, Yuhang Chen, Zhihan Hu, Dongping Chen, Wenhu Chen, Sarah Wiegreffe, and Tianyi Zhou. Quantifying the gap between understanding and generation within unified multimodal models. *arXiv preprint arXiv:2602.02140*, 2026.

Han Wang, Gang Wang, and Huan Zhang. Steering away from harm: An adaptive approach to defending vision language model against jailbreaks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29947–29957, 2025b.

Peiyu Wang, Yi Peng, Yimeng Gan, Liang Hu, Tianyidan Xie, Xiaokun Wang, Yichen Wei, Chuanxin Tang, Bo Zhu, Changshi Li, Hongyang Wei, Eric Li, Xuchen Song, Yang Liu, and Yahui Zhou. Skywork unipic: Unified autoregressive modeling for visual understanding and generation, 2025c. URL https://arxiv.org/abs/2508.03320.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025a. URL https://arxiv.org/abs/2508.02324.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Towards instruction-aligned multimodal generation, 2026. URL https://arxiv.org/abs/2506.18871.

Lyucheng Wu, Mengru Wang, Ziwen Xu, Tri Cao, Nay Oo, Bryan Hooi, and Shumin Deng. Automating steering for safe multimodal large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 792–814, 2025b.

Size Wu, Zhonghua Wu, Zerui Gong, Qingyi Tao, Sheng Jin, Qinyue Li, Wei Li, and Chen Change Loy. Openuni: A simple baseline for unified multimodal understanding and generation, 2025c. URL https://arxiv.org/abs/2505.23661.

Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Zhonghua Wu, Qingyi Tao, Wentao Liu, Wei Li, and Chen Change Loy. Harmonizing visual representations for unified multimodal understanding and generation, 2025d. URL https://arxiv.org/abs/2503.21979.

Xindi Wu, Dingli Yu, Yangsibo Huang, Olga Russakovsky, and Sanjeev Arora. Conceptmix: A compositional image generation benchmark with controllable difficulty, 2024. URL https://arxiv.org/abs/2408.14339.

Yicheng Xiao, Lin Song, Yukang Chen, Yingmin Luo, Yuxin Chen, Yukang Gan, Wei Huang, Xiu Li, Xiaojuan Qi, and Ying Shan. Mindomni: Unleashing reasoning generation in vision language models with rgpo, 2025. URL https://arxiv.org/abs/2505.13031.

Ji Xie, Trevor Darrell, Luke Zettlemoyer, and XuDong Wang. Reconstruction alignment improves unified multimodal models, 2025a. URL https://arxiv.org/abs/2509.07295.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models, 2025b. URL https://arxiv.org/abs/2506.15564.

Haolei Xu, Xinyu Mei, Yuchen Yan, Rui Zhou, Wenqi Zhang, Weiming Lu, Yueting Zhuang, and Yongliang Shen. Easysteer: A unified framework for high-performance and extensible llm steering. *arXiv preprint arXiv:2509.25175*, 2025.

Zhiyuan Yan, Kaiqing Lin, Zongjian Li, Junyan Ye, Hui Han, Haochen Wang, Zhendong Wang, Bin Lin, Hao Li, Xinyan Xiao, Jingdong Wang, Haifeng Wang, and Li Yuan. Unified multimodal models as auto-encoders, 2026. URL https://arxiv.org/abs/2509.09666.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities, 2024. URL https://arxiv.org/abs/2308.02490.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024. URL https://arxiv.org/abs/2311.16502.

Chi Zhang, Jiepeng Wang, Youming Wang, Yuanzhi Liang, Xiaoyan Yang, Zuoxin Li, Haibin Huang, and Xuelong Li. Unimodel: A visual-only framework for unified multimodal understanding and generation, 2025a. URL https://arxiv.org/abs/2511.16917.

Yu Zhang, Jinlong Ma, Yongshuai Hou, Xuefeng Bai, Kehai Chen, Yang Xiang, Jun Yu, and Min Zhang. Evaluating and steering modality preferences in multimodal large language model. *arXiv preprint arXiv:2505.20977*, 2025b.

Hongjue Zhao, Yizhuo Chen, Yuchen Wang, Hairong Qi, Lui Sha, Tarek Abdelzaher, and Huajie Shao. Understanding the theoretical foundations of deep neural networks through differential equations. *arXiv preprint arXiv:2603.18331*, 2026a.

Rui Zhao, Qirui Yuan, Jinyu Li, Haofeng Hu, Yun Li, Zhenhai Gao, and Fei Gao. Sce2drivex: A generalized mllm framework for scene-to-drive learning. *IEEE Robotics and Automation Letters*, 2025a.

Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, and Kaifu Zhang. Unified multimodal understanding and generation models: Advances, challenges, and opportunities, 2026b. URL https://arxiv.org/abs/2505.02567.

Xiaoyan Zhao, Ming Yan, Yilun Qiu, Haoting Ni, Yang Zhang, Fuli Feng, Hong Cheng, and Tat-Seng Chua. Steerx: Disentangled steering for llm personalization, 2025b. URL https://arxiv.org/abs/2510.22256.

Dian Zheng, Manyuan Zhang, Hongyu Li, Kai Zou, Hongbo Liu, Ziyu Guo, Kaituo Feng, Yexin Liu, Ying Luo, Yan Feng, et al. Architecture decoupling is not all you need for unified multimodal model. *arXiv preprint arXiv:2511.22663*, 2025.

Xin Zhou, Dingkang Liang, Sifan Tu, Xiwu Chen, Yikang Ding, Dingyuan Zhang, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Hermes: A unified self-driving world model for simultaneous 3d scene understanding and generation, 2025. URL https://arxiv.org/abs/2501.14729.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL https://arxiv.org/abs/2310.01405.

# A Experiment Details

## A.1 Implementation Detail

All experiments are conducted on NVIDIA A100 GPUs with 80GB memory, using two GPUs for most runs. We use officially released checkpoints for all models, including BAGEL (Deng et al., 2025), Show-o2 (Xie et al., 2025b), Janus-Pro (Chen et al., 2025b), and Qwen-VL-3 30B (Bai et al., 2025), without additional modification or re-training. Steering methods, including CAA (Panickssery et al., 2024), RepE (Zou et al., 2025), and ITI (Li et al., 2024), are implemented in UMMs based on publicly available open-source implementations, following their standard formulations.

For BAGEL generation, we use the default inference configuration unless otherwise specified. Images are generated at a resolution of $1024$, with 50 generation steps and 50 diffusion timesteps. The text and image classifier-free guidance scales are both set to $1.0$. The CFG interval is set to $[0.4, 1.0]$, the CFG renormalization minimum is set to $0.0$, and the timestep shift is set to $3.0$. We use a random seed of 42 for all reported BAGEL experiments.

To ensure reproducibility, all experimental components rely on open-source methods and publicly accessible model checkpoints, and we provide detailed descriptions of dataset construction, steering procedures, and evaluation protocols in the main paper and appendix to facilitate independent verification.

## A.2 Dataset Detail

In this paper, we propose a contrastive paired dataset, UMMSteer, for cross-branch steering. Specifically, for each semantic type, we define multiple binary concepts, where each binary pair corresponds to a contrastive semantic direction, such as `red↔blue` and `above↔below`. The complete set of contrastive semantic concepts is listed in Tab. 3.

For clarity, we also present examples of training and evaluation samples in Fig. 7. For training, each binary concept is expressed in two branch-specific forms. For the understanding branch, we construct 50 contrastive question-answer pairs for each binary concept, where the two answers differ only in the target concept. For the generation branch, we construct 50 paired text prompts, where the prompts differ only in the target semantic attribute. These paired samples are used to extract branch-specific steering vectors. For evaluation, we prepare 50 held-out samples for each individual semantic concept. In U→G steering, the held-out text prompts are used for image generation and then evaluated with a VQA-based framework. In G→U steering, the held-out QA samples are used to measure whether steering changes the model's answer preference toward the target concept.

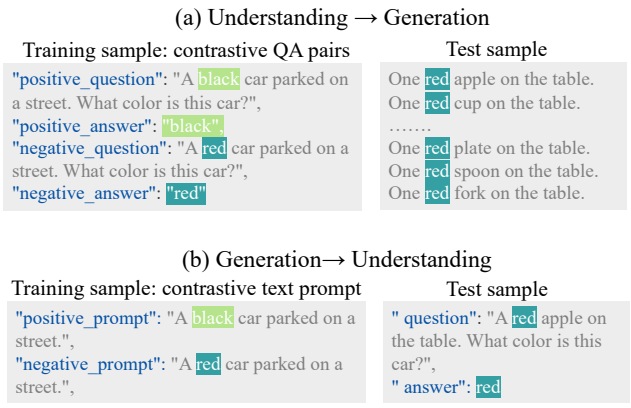


Figure 7: **Illustration of cross-branch steering data samples.** (a) Understanding-to-generation (U→G): steering vectors are learned from contrastive QA pairs in the understanding branch and applied to image generation, where output image are evaluated using a VQA-based framework. (b) Generation-to-understanding (G→U): steering vectors are learned from contrastive text prompt pairs in the generation branch and applied to the understanding task, where effectiveness is measured via logit-difference on target answers.

Table 3: **Hierarchical taxonomy of visually grounded contrastive concepts used for cross-branch steering.** The concepts are organized into three levels: object-level (intrinsic properties), relational (spatial and structural relations), and scene-level (global appearance and style). Each concept is instantiated as a binary contrastive pair to enable precise semantic control.

| Category | Family | Example Binary / Contrastive Concepts |
|---|---|---|
| **Object-level** | **Color** | black↔red, black↔white, blue↔green, pink↔red; red↔blue, red↔green, yellow↔blue, yellow↔green |
| | **Text** | apple↔orange, banana↔lemon, car↔bus, red↔blue, red↔green, sherry↔monica, three↔ten, twenty-three↔ten |
| | **Counting** | one↔two, one↔three, one↔four, one↔five; two↔three, two↔four, two↔five, three↔four, three↔five, four↔five |
| **Relational** | **Position** | left↔right, above↔below, above↔under, inside↔outside, front↔behind, near↔far, clustered↔scattered |
| **Scene-level** | **Appearance** | clean↔dirty, full↔empty, intact↔broken, new↔worn, round↔square, smooth↔rough, tall↔short, wooden↔metal |
| | **Style & Tone** | colorful↔monochrome, vivid↔dull, warm↔cool, realistic↔cartoon, sketch↔oil painting, happy↔unhappy, peaceful↔chaotic, romantic↔gloomy |
| | **Persona** | extraversion↔introversion, neuroticism↔ calmness |

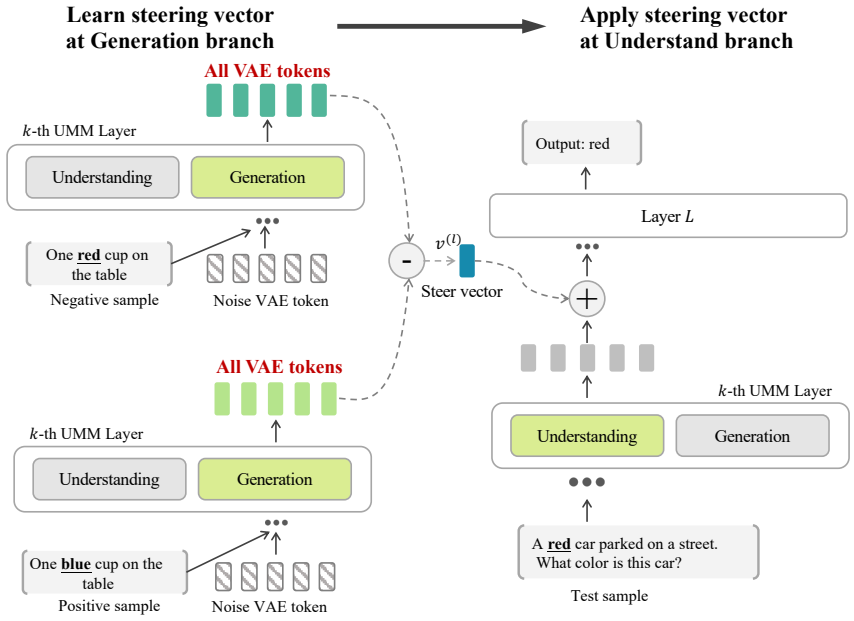


Figure 8: **Cross-branch steering pipeline (Gen → Und).** First, contrastive text prompt pairs (e.g., `'blue'` vs. `'red'`) are fed through the generation branch. At each layer, we extract residual-stream activations from the VAE-token hidden states and use them to compute a generation-derived steering vector. At inference time, a test sample is fed into the understanding branch, and the steering vector is added to the text-token hidden states at each layer to test whether generation-derived directions can bias the model's textual prediction.

### A.3 From Generation to Understanding Steering Process Detail

Fig. 8 illustrates the pipeline for transferring steering vectors from the generation branch to the understanding branch. This setting is the reverse direction of the Und→Gen transfer studied in the

main text. The goal is to examine whether semantic directions learned from the generation process can also influence the model's textual understanding behavior.

Given a pair of contrastive generation prompts that differ only in the target semantic concept, such as `'red'` versus `'blue'`, we feed each prompt into the generation branch and collect the hidden states of the VAE latent tokens at each UMM layer. Unlike the Und→Gen setting, where the target semantic concept is explicitly represented at the answer-token position, the generation branch does not provide a single token that directly corresponds to the target object or attribute. To obtain a semantically targeted representation, we adopt an object-specific extraction strategy. Specifically, we first localize the target object using its bounding box and map the localized image region to the corresponding VAE token positions. We then pool only the VAE tokens within this region, thereby excluding most background and scene-level information. Specifically, we obtain the bounding box using Grounding DINO (Liu et al., 2023). For attribute concepts such as `red`, we use object-specific grounding prompts, such as `red book`, to improve localization accuracy.

Specifically, for each contrastive prompt pair $x_i = (x_{+,i}, x_{-,i})$ at layer $l$, we compute the mean residual-stream activation over all VAE latent token positions:

$$\bar{\mathbf{h}}^{(l)}_{\text{gen},+}(x_i) = \frac{1}{|\mathcal{T}_{\text{vae}}|} \sum_{t \in \mathcal{T}_{\text{vae}}} \mathbf{h}^{(l)}_{t,+}, \quad \bar{\mathbf{h}}^{(l)}_{\text{gen},-}(x_i) = \frac{1}{|\mathcal{T}_{\text{vae}}|} \sum_{t \in \mathcal{T}_{\text{vae}}} \mathbf{h}^{(l)}_{t,-}, \tag{4}$$

where $\mathcal{T}_{\text{vae}}$ denotes the set of VAE latent token positions and $\mathbf{h}^{(l)}_{t,\pm}$ denotes the residual-stream hidden state of token $t$ at layer $l$ under the positive or negative generation prompt. For notational simplicity, we denote these averaged activations as $\mathbf{h}^{(l)}_{+,i} = \bar{\mathbf{h}}^{(l)}_{\text{gen},+}(x_{+,i})$ and $\mathbf{h}^{(l)}_{-,i} = \bar{\mathbf{h}}^{(l)}_{\text{gen},-}(x_{-,i})$. This yields a set of paired generation-branch activations $\{(\mathbf{h}^{(l)}_{+,i}, \mathbf{h}^{(l)}_{-,i})\}_{i=1}^{N}$, which are then used to compute layer-wise steering vectors with the methods described in Sec. A.4.

At inference time, we transfer the generation-derived vector to the understanding branch. Given an input question-answering sample, the model processes the text input through the understanding branch. At each layer, we add the generation-derived steering vector $\mathbf{v}^{(l)}$ to the hidden states of the text tokens:

$$\mathbf{h}^{(l)}_t \leftarrow \mathbf{h}^{(l)}_t + \alpha \mathbf{v}^{(l)}, \qquad t \in \mathcal{T}_{\text{text}}, \tag{5}$$

where $\alpha$ is the steering strength and $\mathcal{T}_{\text{text}}$ denotes the text-token positions in the understanding input. We then evaluate whether the intervention shifts the model's answer toward the target semantic concept.

### A.4 LLM Steering Methods Detail

In Sec. 3.2, we briefly introduced three representative methods for estimating steering directions: CAA (Panickssery et al., 2024), RepE (Zou et al., 2025), ITI (Li et al., 2024). Here, we provide their formal definitions in details.

For a target concept $k$, we construct contrastive pairs from $\mathcal{D}^k$ and feed them into the source branch. We extract residual stream activations at layer $l$, taken from the answer token position in the understanding branch, or from VAE token positions in the generation branch. This yields a set of paired activations $\{\mathbf{h}^{(l)}_{+,i}, \mathbf{h}^{(l)}_{-,i}\}_{i=1}^{N}$, where $+$ and $-$ denote the positive and negative instances, respectively. We define the difference vectors as $\Delta\mathbf{h}^{(l)}_i = \mathbf{h}^{(l)}_{+,i} - \mathbf{h}^{(l)}_{-,i}$.

**(1) Mean Difference (CAA (Panickssery et al., 2024)).** The steering direction is estimated as the average contrastive shift:

$$\mathbf{v}^{(l)}_{\text{CAA}} = \frac{1}{N} \sum_{i=1}^{N} \Delta\mathbf{h}^{(l)}_i. \tag{6}$$

**(2) First Principal Component of the Difference Vector (RepE (Zou et al., 2025)).** We compute the top principal component of the difference vectors:

$$\mathbf{v}^{(l)}_{\text{RepE}} = \text{TopPC}\left(\{\Delta\mathbf{h}^{(l)}_i\}_{i=1}^{N}\right), \tag{7}$$

where $\text{TopPC}(\cdot)$ returns the leading principal component (unit-normalized) corresponding to the largest singular value.

**Steering Effectiveness Evaluation Prompt Template**

You are judging whether an image satisfies a target semantic induced by steering. Use only visible evidence from the image. Return JSON only and do not reveal chain-of-thought.
**Target semantic:**

*[Target Semantic $s$, e.g., blue apple, two dogs, text "STOP", object above table]*

**Judgment mode:**
*[Judge Mode $m \in$ {item_attribute, count, text_content, persona, relation, global_attribute}]*

**Stage 1: Evidence / object-presence check**
Question ID: `q_gate`
Question Type: *[Gate Type]*
Question: *[Does the image clearly contain the target item / readable text / person / required relation objects / visual evidence?]*
Intent: Check whether the image contains sufficient visible evidence for evaluating the target semantic.
Positive signal: *[The required object, subject, text, or visual evidence is clearly visible.]*

**Stage 2: Target-semantic match check**
Question ID: `q_target`
Question Type: *[Target Type]*
Question: *[If the required evidence is present, does it match the target semantic $s$?]*
Intent: Check whether the generated image satisfies the desired steered semantic.
Positive signal: *[The visible evidence matches the target attribute/count/text/persona/relation/global semantic.]*

**Return JSON only:**

```
{
  "question_id": "...",
  "question_type": "...",
  "label": "yes" | "no" | "uncertain",
  "confidence": 0.0,
  "answer": "...",
  "evidence": "...",
  "short_rationale": "..."
}
```

Figure 9: Two-stage VQA evaluation prompt for measuring steering success. The judge first verifies whether the generated image contains sufficient visible evidence for evaluation, and then checks whether the target semantic is satisfied. A generation is counted as successful only when both the gate question and the target-semantic question are answered `yes`.

**(3) Normal Vector of a Learned Decision Boundary (ITI (Li et al., 2024)).** We train a linear classifier to distinguish positive and negative activations, and use the normal vector of the decision boundary as the steering direction:

$$\mathbf{v}_{\text{ITI}}^{(l)} = \text{Classify}\left(\{\mathbf{h}_{\pm,i}^{(l)}\}_{i=1}^{N}\right), \tag{8}$$

where $\text{Classify}(\cdot)$ returns the learned separating direction, normalized to match the standard deviation along that axis.

In summary, CAA (Panickssery et al., 2024) captures the average semantic shift between contrastive states, RepE (Zou et al., 2025) extracts the dominant variation direction, and ITI (Li et al., 2024) explicitly optimizes discriminative separation. Despite their different formulations, all three methods recover a semantic direction $\mathbf{v}^{(l)}$ in hidden space, which can be used to steer model behavior toward the target concept.

### A.5 Evaluation Metric

**Steering Success Rate (SSR).** Following Sec. 3, we adopt a question-answering (QVA) protocol with Qwen-VL-3 30B (Bai et al., 2025) as the evaluator to assess steering effectiveness in the

generation branch. The evaluation consists of two criteria. First, we verify whether the generated image preserves the target object, ensuring that steering does not alter object identity. Second, we assess whether the desired semantic attribute (e.g., spatial relation or color) is successfully enforced. A sample is considered successful only if both criteria are satisfied. We show the two-stage VQA evaluation prompt template in Fig. 9. Formally, we define the *steering success rate* (SSR) as:

$$\text{SSR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\big(y_i^{\text{obj}} = 1 \ \wedge \ y_i^{\text{attr}} = 1\big), \tag{9}$$

where $N$ is the total number of evaluation samples, $y_i^{\text{obj}} \in \{0, 1\}$ indicates whether the target object is preserved, and $y_i^{\text{attr}} \in \{0, 1\}$ indicates whether the desired semantic attribute is successfully enforced. $\mathbb{I}(\cdot)$ denotes the indicator function. Here, we use Qwen-VL-3 30B (Bai et al., 2025) as the evaluator due to its strong performance on multimodal reasoning benchmarks. In addition, our evaluation is based on simple, visually grounded verification (e.g., object presence and attribute matching), which reduces reliance on complex reasoning and improves robustness.

Here, we include qualitative examples of evaluator outputs in Fig. 10 for clarity.

**Generation Score.** Following GenEval (Ghosh et al., 2023a) and WISE (Niu et al., 2025), we adopt a VQA-based evaluation protocol using Qwen3-VL-30B (Bai et al., 2025) as the judge to assess text–image alignment. A generated image is considered correct only if it contains the target object and satisfies the target attribute or relation specified in the prompt. Formally, we define the *generation score* as:

$$\text{GenScore} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\big(y_i^{\text{obj}} = 1 \ \wedge \ y_i^{\text{attr}} = 1\big), \tag{10}$$

where $N$ is the total number of evaluation samples, $y_i^{\text{obj}} \in \{0, 1\}$ indicates whether the target object is present in the generated image, and $y_i^{\text{attr}} \in \{0, 1\}$ indicates whether the desired attribute or relation is satisfied. $\mathbb{I}(\cdot)$ denotes the indicator function. The evaluation prompt for Qwen3-VL-30B (Bai et al., 2025) follows the same two-stage structure as in SSR, consisting of object presence verification and attribute (or relation) matching. The key difference is that, in SSR, the target attribute is defined by the applied steering vector, whereas in the generation score it is specified by the input prompt.

# B Additional results

## B.1 Human Agreement Study

To validate the reliability of our question-answering-based visual assessment (QVA) protocol, we conduct a human agreement study on a randomly sampled subset of generated images. Specifically, we sample 50 images for each of the seven semantic types in UMMSteer: *color*, *text*, *counting*, *position*, *style*, *appearance*, and *persona*, resulting in 350 annotated images in total. For each image, a human annotator independently judges the same binary verification criteria used by the automatic evaluator: whether the target object is present and whether the target semantic attribute or relation is correctly realized. The annotator is shown the generated image, the prompt, the target object, and the target semantic attribute or relation.

We compute Cohen's $\kappa$ between the human annotations and Qwen-VL-3 30B (Bai et al., 2025) predictions to quantify chance-corrected human–model agreement. We also report raw agreement for completeness. For each sample, we evaluate overall correctness, where a generated image is considered correct only if both the target object is present and the target attribute or relation is correct. Cohen's $\kappa$ accounts for chance agreement and is therefore more conservative than raw agreement.

As shown in Tab. 4, Qwen-VL-3 30B achieves high agreement with human annotations across all seven semantic types. Across 350 annotated samples, the QVA-based evaluator obtains a raw agreement of 95.4% and a Cohen's $\kappa$ of 0.894, indicating strong chance-corrected agreement with human judgments. These results support the reliability of our automatic evaluation protocol.

"image_path": "1.png",
"target item": : "toy"
"target count": : "5"
#==========Step1: item_presence==========
"question": "Does the image clearly contain the target item 'toy'?",
"intent": "Check whether the target item needed for counting is visible.",
"positive_signal": "At least one clearly visible instance of 'toy' appears in the image."
"label": "yes",
"confidence": 0.98,
"answer": "yes",
"evidence": "The image shows five distinct plush toys (a monkey, a white bear, an orange cat, a green dragon, and a yellow lion) arranged on a couch.",
"short rationale": "Five plush toys are clearly visible on the couch.",
"raw response": "Got it, let's check the image. The problem is to see if there are 5 toys. Wait, the target semantic is \"5 toy\", but the question is about item presence. So the target item is 'toy', and we need to see if toys are present. The target item 'toy' is present. So the answer should be yes; \"item_presence\",\n \"label\": \"yes\",\n \"confidence\": 0.98,\n \"answer\": \"yes\",\n \"evidence\": \"The image shows five distinct plush toys (a monkey, a white bear, an orange cat, a green dragon, and a yellow lion) arranged on a couch.\",\n \"short_rationale\": \"Five plush toys are clearly visible on the couch"
},
#==========Step1: count_match==========
"question": "If 'toy' is present, is the number of visible 'toy' objects exactly 5?",
"intent": "Check whether the visible count matches the target count.",
"positive_signal": "The image shows exactly 5 visible 'toy' objects."
"label": "yes",
"confidence": 0.99,
"answer": "yes",
"evidence": "The image shows five distinct plush toys (a monkey, a white bear with red feet, an orange cat, a green dragon, and a small green lion) arranged on a couch.",
"short_rationale": "Five toys are visible on the couch.",
"raw response": "Got it, let's count the toys in the image. There are five plush toys on the couch: the monkey, the white bear with red feet, the orange cat, the green dragon, and the small green lion. So that's exactly 5. The question is asking if the number is exactly 5. So the answer should be \"yes\".\n"question_type\": \"count_match\",\n \"label\": \"yes\",\n \"confidence\": 0.99,\n \"answer\": \"yes\",\n \"evidence\": \"The image shows five distinct plush toys (a monkey, a white bear with red feet, an orange cat, a green dragon, and a small green lion) arranged on a couch.\",\n \"short_rationale\": \"Five toys are visible on the couch.\"\n}"


Figure 10: **Qualitative examples of evaluator outputs from Qwen3 VL-30B (Bai et al., 2025).**

Table 4: Human–model agreement for validating the QVA-based evaluator. We report raw agreement and Cohen's $\kappa$ between human annotations and Qwen-VL-3 30B predictions across semantic types.

| Semantic type | Agreement (%) | Cohen's $\kappa$ |
|---|---|---|
| Text | 98.0 | 0.658 |
| Position | 98.0 | 0.790 |
| Style | 100.0 | 1.000 |
| Persona | 90.0 | 0.800 |
| Appearance | 90.0 | 0.786 |
| Counting | 98.0 | 0.929 |
| Color | 94.0 | 0.540 |
| Overall | 95.4 | 0.894 |

### B.2 Full Steer Effectiveness Evaluation for Understanding-to-Generation Steering

We observe substantial within-category variance in Tab. 1, and therefore further present the full steering success rate (SSR) of all semantic concept pair from our UMMSteer in Fig. 11, Fig. 13 and Fig. 12. Here, each concept pair (`'a'` vs. `'b'`) is evaluated bidirectionally: (i) for `'a'`$\rightarrow$`'b'`, we use prompts containing attribute `'a'` and apply the `'a'`$\rightarrow$`'b'` steering vector to enforce attribute `'b'`; (ii) for `'b'`$\rightarrow$`'a'`, we perform the reverse. The final SSR is obtained by averaging the two directions.

First, we report the SSR for all **object-level** semantic concepts in Fig. 11. For *Color* semantics, both CAA (Panickssery et al., 2024) and ITI (Li et al., 2024) achieve consistently high performance

across all concept pairs, indicating that these methods can reliably capture and transfer color-related semantic directions. In contrast, RepE (Zou et al., 2025) exhibits unstable behavior, particularly failing on concepts involving `'green'`, `'yellow'`, and `'blue'`, leading to significantly degraded performance. For *Text* semantics, all methods perform well on short and simple tokens (e.g., `'apple'`, `'orange'`), suggesting that such semantics are relatively easy to encode and transfer. However, performance drops substantially for longer or more complex words (e.g., `'banana'`, `'lemon'`, `'Monica'`, `'Sherry'`), where all methods struggle, indicating limitations in capturing fine-grained textual semantics in generation. For *Counting* semantics, we observe a clear trend that steering difficulty increases with the target number. Specifically, concepts involving larger counts (e.g., 4 vs. 5) yield much lower SSR compared to smaller counts (e.g., 1 vs. 2), suggesting that precise numerical control in generation remains challenging, especially as the counting complexity increases.

Second, we report the full SSR for **relational** semantics in Fig. 12. Most spatial relations (e.g., `'above'` vs. `'below'`, `'inside'` vs. `'outside'`, `'left'` vs. `'right'`) can be reliably steered, with CAA (Panickssery et al., 2024) and ITI (Li et al., 2024) achieving consistently strong performance. However, more complex structural relations such as `'clustered'` vs. `'scattered'` are noticeably harder, especially for RepE (Zou et al., 2025), which shows large performance drops. Overall, relational semantics are easier to steer than counting and complex text, but still exhibit variability depending on spatial complexity.

Last, we report the full SSR for **scene-level** semantics in Fig. 13. For *Appearance*, CAA (Panickssery et al., 2024) and ITI (Li et al., 2024) achieve strong performance on most concepts (e.g., `'metal'`, `'wooden'`, `'round'`, `'square'`, `'short'`, `'tall'`), while some attributes such as `'clean'`, `'dirty'`, `'empty'` and `'full'` are relatively harder. RepE (Zou et al., 2025) shows unstable behavior and significantly underperforms across many pairs. For *Style*, CAA (Panickssery et al., 2024) and ITI (Li et al., 2024) perform well on visually distinctive styles such as `'cartoon'`, `'realistic'`, `'colorful'`, `'monochrome'`, and `'cool'`, `'warm'`. The lower SSR on `'oil'`, `'sketch'` is likely due to the difficulty of generating the oil-painting style itself. For *Persona*, all methods perform poorly (e.g., `'calmness'`, `'neuroticism'`), suggesting that such abstract and weakly visualizable semantics are difficult to steer.

### B.3 Full Steer Effectiveness Evaluation for Generation-to-Understanding Steering

We further evaluate whether steering vectors learned from the generation branch can be transferred to the understanding branch.

Specifically, for each contrastive semantic concept, such as `'blue'` vs. `'red'`, we extract steering vectors from the generation branch and apply them to the hidden states of the understanding branch during visual question answering. The goal is to test whether these generation-derived directions can shift the model's textual prediction toward the target semantic label, as shown in Fig. 4(a).

For evaluation, we use the target semantic label as the ground-truth answer and compute the prediction accuracy between the model output and the target semantic label. For example, when applying a `red`→`blue` steering direction, a successful transfer should make the model answer `blue` when queried about the corresponding visual attribute with an input `'a red car'`.

To rule out the possibility that the failure is caused by an inappropriate steering magnitude, we conduct a dense sweep over the steering scale $\alpha$ and evaluate the model under each scale value. However, even under this exhaustive scale search, we do not observe any scale value that consistently shifts the model output toward the target semantic label. Across all evaluated semantic concepts, **the prediction accuracy remains zero**, indicating that generation-derived steering vectors fail to induce meaningful changes in the understanding branch. This result shows that semantic directions learned from the generation branch are not directly usable for controlling text-side understanding behavior.

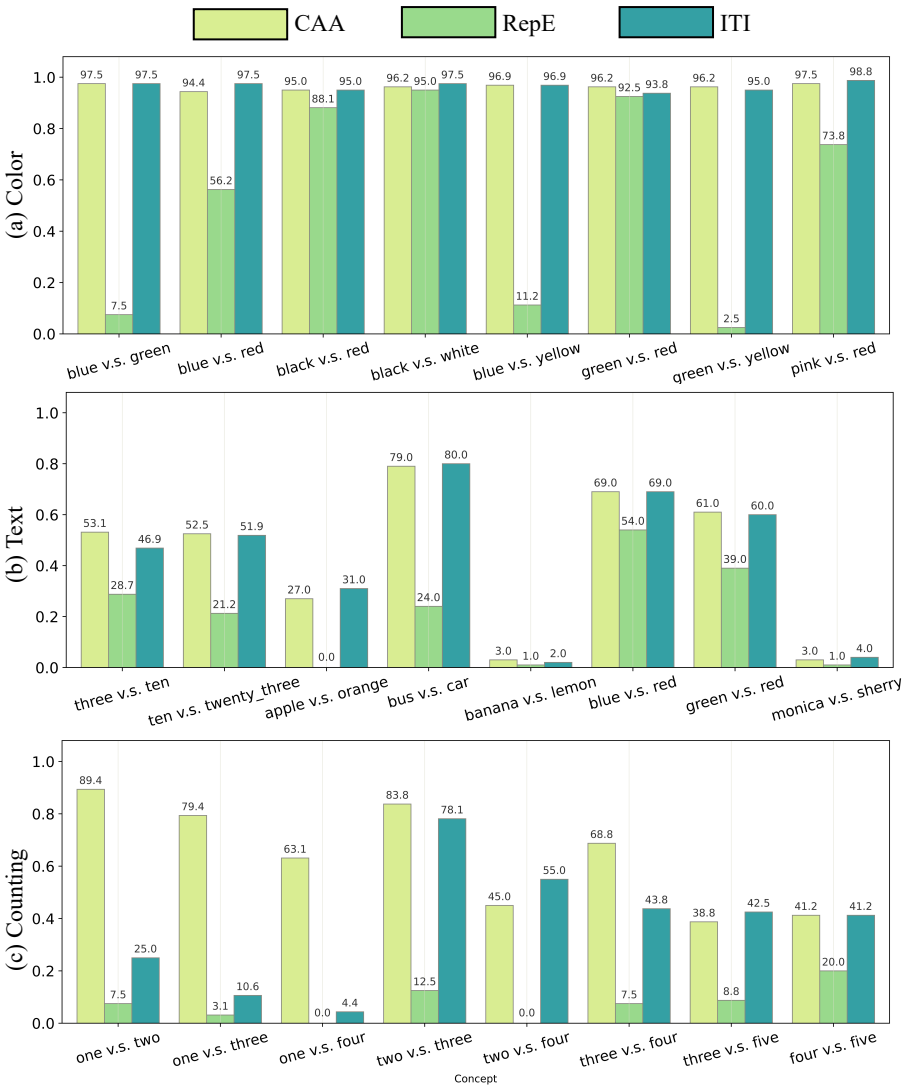


Figure 11: Steering success rates for each concept across object-level semantics.

# C Additional Ablation Study and Analysis

## C.1 Additional Baseline Experiments

To further examine whether successful cross-branch steering can be attributed to a generic perturbation rather than the extracted target direction, we introduce two additional baselines for understanding-to-generation (Und→Gen) steering on BAGEL (Deng et al., 2025):

- *Random-direction baseline.* At each layer, we sample a random direction and rescale it to match the $\ell_2$ norm of the corresponding understanding-derived CAA (Panickssery et al., 2024) vector. This baseline tests whether a perturbation with a comparable magnitude is sufficient to induce successful steering.
- *Unrelated-direction baseline.* We extract an understanding-derived CAA vector from a semantically unrelated persona concept, specifically `extraversion` versus `introversion`, and apply it to the evaluation of color, counting, position, and style. This baseline tests whether an unrelated but semantically meaningful direction can produce similar steering effects.

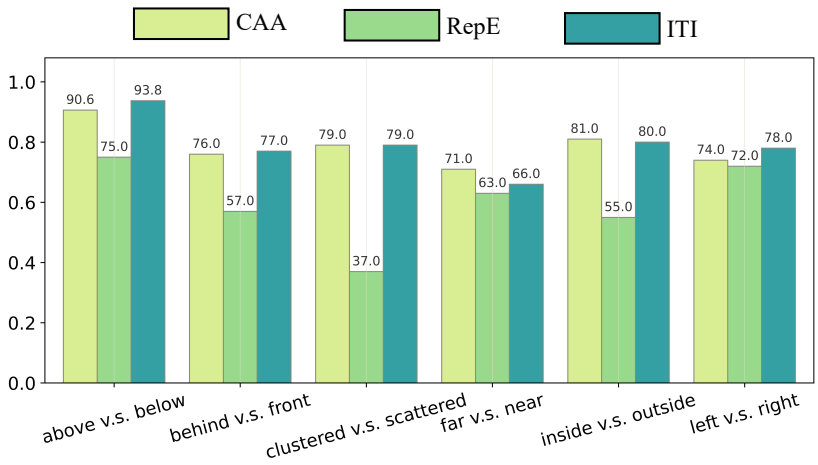


Figure 12: Steering success rates for each concept across relational semantics.

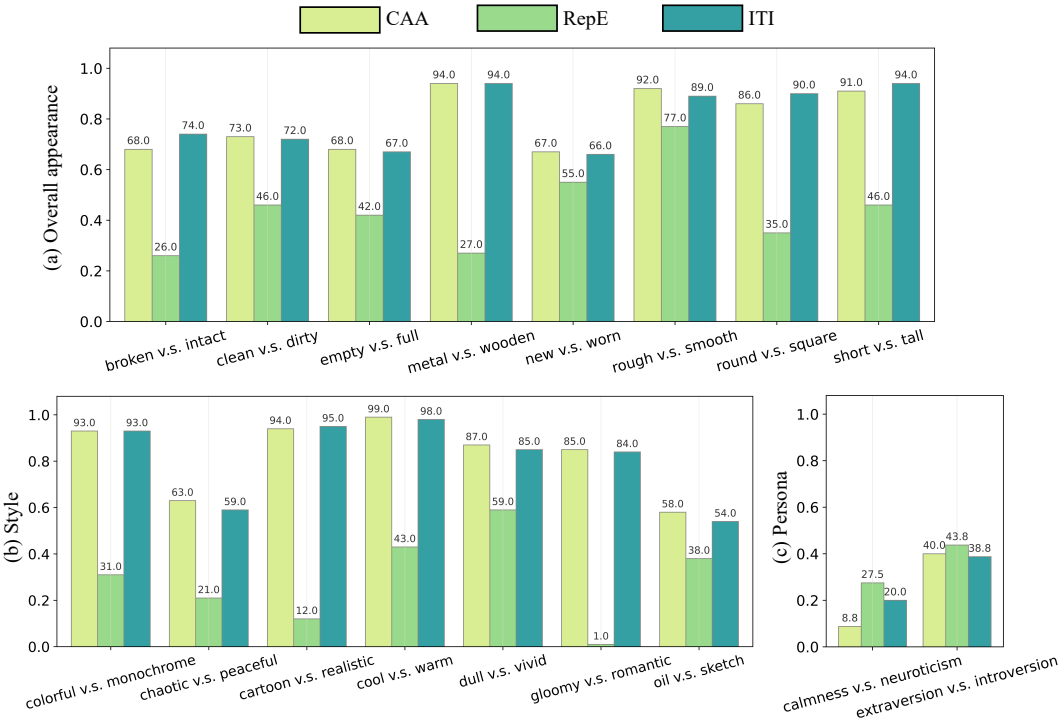


Figure 13: Steering success rates for each concept across scene-level semantics.

All directions are applied using the same intervention and evaluation protocol. We report the steering success rate (SSR) in Table 5.

As shown in Table 5, the target CAA vectors achieve high SSR across all four semantic categories, whereas both the norm-matched random directions and the unrelated semantic direction yield an SSR of 0.0%. These results show that successful Und→Gen steering cannot be explained solely by the magnitude of the intervention or by injecting an arbitrary semantic direction. Instead, the steering effect depends on the target-specific semantic information encoded in the extracted vector.

Table 5: Und→Gen steering success rate (SSR, %) on BAGEL using CAA and two additional baselines.

| Method | Color | Counting | Position | Style |
|---|---|---|---|---|
| CAA vector | **96.20** | **68.75** | **78.60** | **82.71** |
| Random direction | 0.00 | 0.00 | 0.00 | 0.00 |
| Unrelated direction | 0.00 | 0.00 | 0.00 | 0.00 |

Table 6: Gen→Und steering success rate (SSR, %) across different generation-branch extraction strategies. Object-aware pooling is our default extraction method, while global mean pooling and attention-weighted pooling serve as alternative strategies.

| Generation-Branch Extraction | Color | Counting | Style | Appearance |
|---|---|---|---|---|
| Object-aware pooling (default) | 0.0 | 0.0 | 0.0 | 0.0 |
| Global mean pooling | 0.0 | 0.0 | 0.0 | 0.0 |
| Attention-weighted pooling | 0.0 | 0.0 | 0.0 | 0.0 |

### C.2 Additional Steering Vector Extraction Strategies Comparison in the Generation Branch

As described in Sec. 4.2 and Sec. A.3, we adopt object-aware pooling as our default strategy for extracting semantic steering vectors from the generation branch. Unlike the understanding branch, where the target concept is explicitly represented at the answer-token position, the generation branch does not provide a single token that directly corresponds to the target semantic concept. We therefore extract object-specific representations by pooling the hidden states of VAE latent tokens associated with the target object.

Specifically, we first localize the target object using Grounding DINO (Liu et al., 2023) and map its bounding box to the corresponding VAE token positions. We then average the hidden states of the selected tokens to obtain an object-specific representation. For attribute concepts such as `red`, we use object-specific grounding prompts, such as `red book`, to improve localization accuracy. By restricting extraction to the target object region, this strategy reduces interference from unrelated background and scene-level information, providing a semantically targeted counterpart to answer-token extraction in the understanding branch.

To examine whether the observed generation-to-understanding (Gen→Und) transfer failure is sensitive to the choice of extraction strategy, we additionally evaluate two alternative pooling methods:

- *Global mean pooling.* We average the hidden states of all VAE latent tokens to obtain a global generation-branch representation. This architecture-agnostic strategy does not require external grounding modules or token-selection heuristics. However, it may entangle information about object identity, background, spatial layout, illumination, and global appearance, potentially diluting object-specific semantic information.
- *Attention-weighted pooling.* We weight each VAE token according to its attention association with the target concept token, such as `red`, and compute a weighted average of the corresponding VAE hidden states. This strategy assigns greater weight to tokens associated with the target concept without explicitly restricting extraction to a localized object region.

For each extraction strategy, we compute generation-derived steering vectors using CAA and apply them to the hidden states of the understanding branch during visual question answering. We evaluate cross-branch transfer using the steering success rate (SSR), which measures whether the intervention shifts the textual prediction toward the target semantic label. The results are reported in Table 6.

Across all three extraction strategies and four representative semantic categories, generation-derived vectors fail to induce successful Gen→Und steering, with SSR remaining at 0.0%. This result is also consistent across a broad range of intervention strengths. Notably, restricting extraction to object-specific VAE tokens does not improve cross-branch transfer compared with global mean pooling or attention-weighted pooling.

These findings indicate that the observed Gen→Und failure cannot be attributed solely to the spatial aggregation strategy. However, the effectiveness of generation-derived steering vectors may also depend on other extraction factors, such as the denoising timestep. Our results therefore establish limited Gen→Und transfer under the evaluated extraction and intervention protocols, while leaving the underlying representational mechanisms open to further investigation.

### C.3 Additional Layer-wise Logit Difference Analysis

In Sec. 4.2, to evaluate the effectiveness of steering when applied to the understanding branch, we adopt the logit-difference metric $m_{\mathrm{LD}}$ for the `red↔blue` semantic direction, which measures the model's preference between a target answer and its contrastive alternative (Tan et al., 2024).

Here, we provide additional layer-wise logit-difference results in Fig. 14 for other semantic concepts, including counting (*one↔two*) and appearance (*clean↔dirty*). Across these settings, steering vectors derived from the understanding branch consistently produce positive $\Delta m_{\mathrm{LD}}$ across layers, indicating a reliable shift toward the target semantic attribute. In contrast, steering vectors derived from the generation branch fail to induce meaningful changes in the model's predictions. Their corresponding $\Delta m_{\mathrm{LD}}$ values remain close to zero or even negative in early layers, suggesting that generation-derived directions are misaligned with the semantic representations used in the understanding branch.

These results are consistent with the main findings and further confirm that cross-branch transfer is asymmetric: only understanding-derived semantic directions can effectively influence the understanding process, while generation-derived directions lack the necessary alignment to modify the model's predictions.

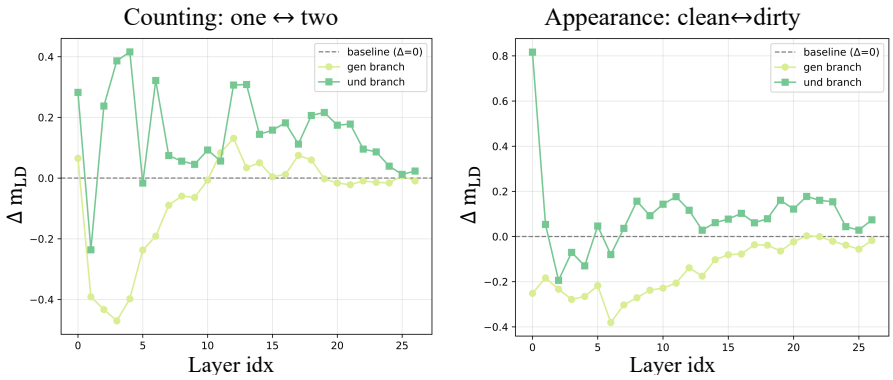


Figure 14: Additional Layer-wise Logit Difference Analysis.

### C.4 PCA Subspace Projection for Steering Analysis

In Sec. 5, we observe a clear asymmetry: steering fails from generation to understanding, while the reverse direction succeeds. To analyze this phenomenon, we hypothesize that semantic directions learned in the generation branch are not well aligned with the semantic structure used by the understanding branch.

To quantitatively verify this hypothesis, we introduce the *PCA Subspace Projection* (PSP) metric, which measures how well a steering vector aligns with a target semantic subspace induced by contrastive data. Intuitively, if a steering vector captures the intended semantic concept, it should lie within (or close to) the dominant subspace spanned by semantic variations of that concept.

**PCA Subspace Projection (PSP) Formulation.** First, we describe the computation pipeline of PSP as follows: *Step 1: Construct semantic difference vectors.* Given a set of contrastive pairs, we extract the hidden representations corresponding to the target tokens (e.g., answer tokens) and compute difference vectors:

$$\mathbf{d}_i = \mathbf{h}_{+,i} - \mathbf{h}_{-,i}, \tag{11}$$

Table 7: Und→Gen steering success rate (SSR, %) on out-of-template compositional prompts from ConceptMix. The compositional difficulty increases with $k$.

| Semantic Concept | $k = 2$ | $k = 3$ | $k = 4$ |
|---|---|---|---|
| Color | 46.25 | 21.43 | 20.45 |
| Counting | 42.22 | 33.33 | 17.28 |

where $\mathbf{h}_{+,i}$ and $\mathbf{h}_{-,i}$ denote the hidden states of the positive and negative samples, respectively. These difference vectors isolate the semantic variation of interest (e.g., *red* vs. *blue*) while largely canceling out shared contextual information.

*Step 2: Estimate the semantic subspace.* We perform Principal Component Analysis (PCA) on the set $\{\mathbf{d}_i\}_{i=1}^{N}$ to capture the dominant directions of semantic variation. Let

$$U = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathbb{R}^{d \times k} \tag{12}$$

denote the matrix of the top-$k$ principal components. In this paper, we set $k = 20$ for all analysis studies. These orthonormal vectors span a low-dimensional subspace that approximates the intrinsic semantic structure of the target concept.

*Step 3: Measure subspace alignment.* Given a steering vector $\mathbf{s}$, we measure its alignment with the semantic subspace via the projection ratio:

$$\mathrm{PSP}(\mathbf{s}) = \frac{|U^\top \mathbf{s}|^2}{|\mathbf{s}|^2}. \tag{13}$$

This quantity corresponds to the fraction of the energy of $\mathbf{s}$ that lies within the PCA subspace. A higher PSP value indicates stronger alignment with the semantic structure captured by the contrastive data, while a lower value suggests that the steering vector encodes directions orthogonal to the target semantics.

**Additional PCA Subspace Projection Analysis.** In Sec. 5, we measure the PCA Subspace Projection (PSP) for the `red↔blue` semantic direction under both self-branch and cross-branch settings. Additional results across different concepts are shown in Fig. 15, exhibiting consistent patterns. Steering vectors derived from the understanding branch show consistently high alignment in both settings, indicating that they capture generalizable and transferable semantic directions. In contrast, steering vectors from the generation branch exhibit low alignment when projected onto the understanding subspace, explaining their limited influence on downstream predictions in the understanding branch. Overall, these results suggest that cross-branch transfer is governed by subspace alignment: only directions that lie within a unified semantic subspace can transfer effectively across branches.

### C.5 Generalization to Out-of-Template Compositional Prompts

Our primary evaluation uses controlled and largely template-based examples to isolate individual semantic factors and reduce potential confounding effects. This controlled setup enables a more reliable analysis of cross-branch steering, but does not fully reflect the complexity of free-form, real-world text-to-image prompts.

To evaluate whether the extracted understanding-derived steering directions generalize beyond this controlled setting, we conduct an additional out-of-template evaluation using ConceptMix (Wu et al., 2024), a public compositional text-to-image benchmark containing free-form prompts that combine multiple visual concepts. We evaluate prompts at different compositional difficulty levels, denoted by $k$, where a larger $k$ indicates that more visual concepts are simultaneously specified in the prompt.

We consider two representative semantic categories, *color* and *counting*, and apply the corresponding understanding-derived steering vectors to the generation branch. We follow the same steering success rate (SSR) evaluation protocol as in the main experiments. The results are reported in Table 7.

As shown in Table 7, the understanding-derived directions retain non-zero steering effectiveness on free-form prompts containing multiple simultaneous visual concepts. This indicates that the extracted semantic directions are not limited to the controlled prompt templates used in our primary evaluation and can generalize, to some extent, to out-of-template compositional prompts.

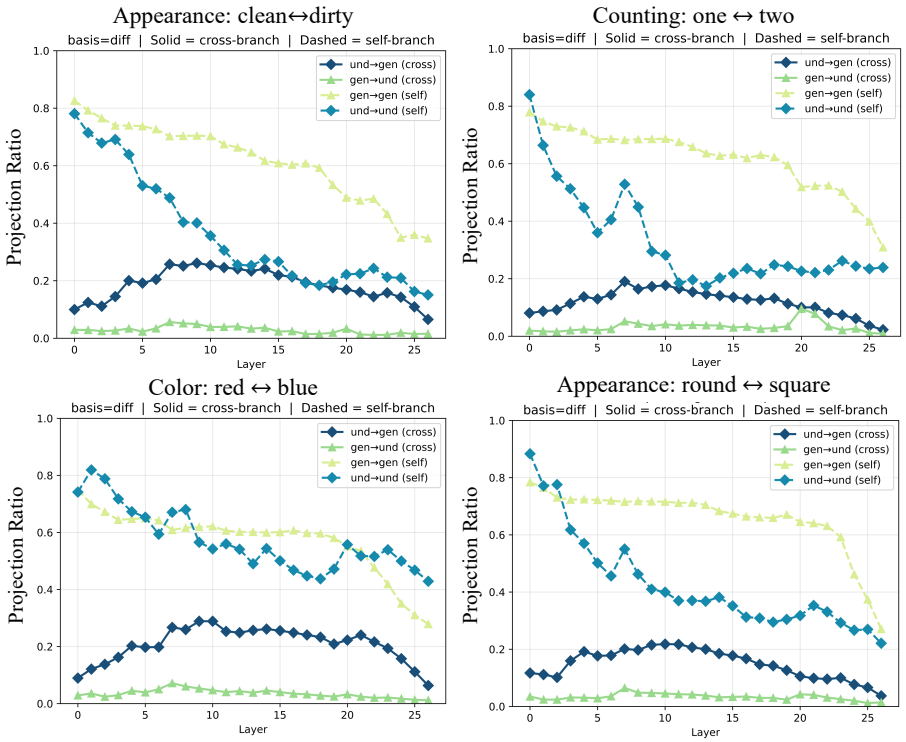


Figure 15: Additional Layer-wise PCA Subspace Projection Analysis.

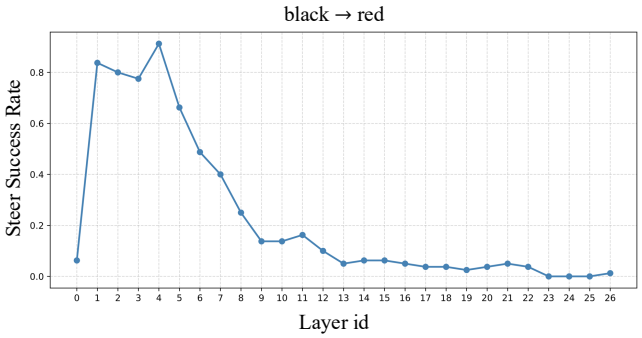


Figure 16: **Ablation about the steering layer.** Steering is most effective in early-to-middle layers (2–5) and degrades in deeper layers, indicating that transferable semantics are primarily encoded in intermediate representations.

At the same time, SSR generally decreases as the compositional difficulty $k$ increases. This suggests that controlling a target semantic attribute becomes more challenging when the prompt contains a larger number of interacting visual constraints. Overall, these results demonstrate partial generalization of Und→Gen steering beyond the controlled evaluation setting, while also highlighting highly compositional generation as an important remaining challenge.

### C.6 Ablation Study

**Ablation study about the steering layer.** In this work, we apply steering vectors to all layers by default. Here, we further analyze the effect of layer-wise steering by applying the steering vector to each layer individually. Specifically, we evaluate steering performance when the intervention is restricted to a single layer at a time, which allows us to examine the contribution of different layers to cross-branch transfer.

| $\alpha$ | 0.1 | 0.15 | 0.2 | 0.25 | 0.3 | 0.35 | 0.4 |
|---|---|---|---|---|---|---|---|
| CAA (Panickssery et al., 2024) | **100.0** | 96.25 | 91.25 | 81.25 | 48.75 | 23.75 | 10.0 |
| RepE (Zou et al., 2025) | 1.25 | 16.25 | **20.00** | 12.5 | 7.5 | 3.75 | 1.25 |
| ITI (Li et al., 2024) | 7.5 | 96.5 | **97.5** | 97.5 | 97.5 | 96.25 | 93.75 |

Table 8: **Ablation on steering strength $\alpha$ for CAA ('red $\rightarrow$ blue').** SSR is reported.

As shown in Fig. 16, steering effectiveness varies significantly across layers. The performance peaks in early-to-middle layers (around layers 2–5), where steering achieves the highest steering success rates, indicating that these layers play a dominant role in encoding transferable semantic directions. In contrast, steering applied to deeper layers leads to a sharp performance drop, with success rates approaching zero in the final layers. This suggests that later layers are less responsive to external semantic interventions and may be more specialized for task-specific decoding. Overall, these results indicate that cross-branch semantic control is primarily mediated by intermediate representations rather than late-stage processing.

**Ablation study about the steering strength $\alpha$.** We study the effect of the steering strength $\alpha$ on cross-branch transfer performance using the `red` $\rightarrow$ `blue` concept. Specifically, we vary $\alpha \in \{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4\}$ and evaluate steering effectiveness using the SSR metric in the generation branch.

As shown in Tab. 8, ITI (Li et al., 2024) demonstrates the strongest robustness across different $\alpha$ values, maintaining consistently high SSR, while RepE (Zou et al., 2025) is the most sensitive to the choice of $\alpha$ and exhibits the weakest performance overall. CAA achieves the best performance at smaller $\alpha$, but degrades rapidly as $\alpha$ increases.

Across all methods, we observe a consistent decline in SSR as $\alpha$ becomes larger, suggesting that excessive steering introduces strong perturbations that disrupt the underlying representations (i.e., over-steering), leading to degraded semantic control. This highlights a clear trade-off between insufficient steering (under-control) and excessive intervention.

Based on these observations, in all subsequent experiments across different semantic concepts, we set $\alpha = 0.1$ for CAA and $\alpha = 0.2$ for the other methods as default configurations.

### C.7 Ablation Study on Gen$\rightarrow$Und Intervention Positions

In our original generation-to-understanding (Gen$\rightarrow$Und) steering setup, we add the generation-derived steering vector to the hidden states of all text tokens. This design is consistent with the understanding-to-generation (Und$\rightarrow$Gen) setting, in which the understanding-derived vector is applied to all VAE-token hidden states. It also avoids making an a priori assumption about which text-token position is primarily responsible for predicting the target semantic concept.

To examine whether the intervention target accounts for the weak Gen$\rightarrow$Und transfer, we additionally evaluate the following intervention positions:

- *All text tokens.* The steering vector is added to the hidden states of all text tokens, following our original setup.
- *Concept-related input tokens.* The steering vector is applied only to input tokens associated with the queried object or attribute.
- *Answer-prediction hidden state.* The steering vector is applied only to the final hidden state used to predict the answer token.

We evaluate these intervention positions using generation-derived vectors obtained with all three extraction strategies introduced in Sec. C.2: global mean pooling, object-aware pooling, and attention-weighted pooling. All other experimental settings, including the intervention layers and steering strengths, are kept unchanged.

Across all evaluated extraction strategies and intervention positions, the steering success rate remains 0.0% for color, counting, style, and appearance. The result also remains unchanged when the steering

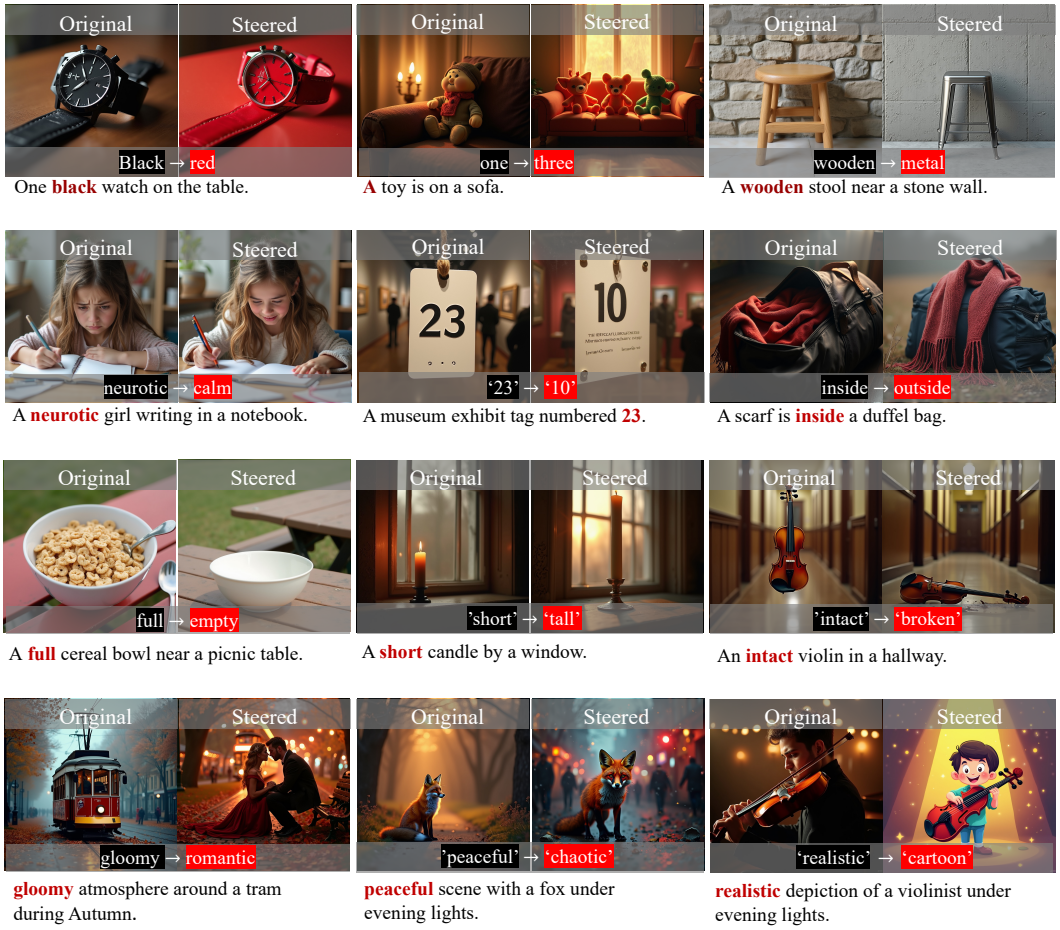


Figure 17: **Additional qualitative results of understanding↔generation steering.** Each pair shows the original generation (left) and the steered output (right). Steering modifies the target attribute while preserving object identity.

strength is varied over a broad range. Therefore, the weak Gen→Und transfer cannot be attributed solely to the use of all text-token hidden states as the intervention target.

These results show that the Gen→Und failure persists across both extraction strategies and intervention positions. Under the evaluated protocols, generation-derived steering directions do not encode the target concepts in a form that can be directly transferred to and used to control the understanding branch.

# D Additional Qualitative Result

## D.1 Additional Qualitative Results for Understanding-to-Generation Steering

In this section, we present additional qualitative examples in Fig. 17, illustrating cross-branch steering from the understanding branch to the generation branch using CAA (Panickssery et al., 2024). Each pair consists of the original output (left) and the steered output (right).

Across all semantic categories in our UMMSteer, steering consistently modifies the target attribute while preserving object identity and overall scene composition. For instance, color attributes are precisely altered (e.g., *black* → *red*, *blue* → *green*), object counts are adjusted without changing the scene layout (e.g., 1 → 3), and appearance and style are successfully manipulated (e.g., *full* → *empty*, *gloomy* → *romantic*).

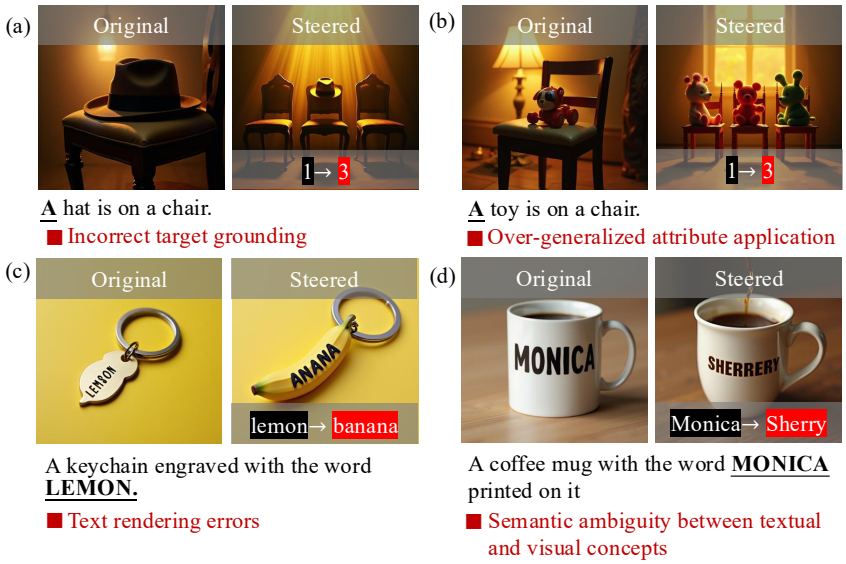


Figure 18: **Additional qualitative results of understanding↔generation steering.** Each pair shows the original generation (left) and the steered output (right). Steering modifies the target attribute while preserving object identity.

Overall, these qualitative results provide strong visual evidence that cross-branch steering from understanding to generation enables precise and compositional semantic control.

### D.2 Failure Case Analysis

As shown in Fig. 18, we further present representative failure cases of CAA-based understanding-to-generation steering. Although CAA generally enables precise cross-branch control, failures still arise when the model cannot reliably localize the target object or disentangle the intended semantic attribute.

We observe four main failure modes: (**i**) *Incorrect target grounding*, as shown in Fig. 18(a), where the steering is applied to a correlated but unintended object; (**ii**) *Over-generalized attribute application*, see Fig. 18(b), where the semantic direction affects multiple objects rather than the intended target; (**iii**) *Text rendering errors*, where the intended textual modification is not faithfully realized, e.g., incorrect spelling in Fig. 18(c); (**iv**) *Semantic ambiguity between textual and visual concepts*, see Fig. 18(d), where the model confuses text content with visual object attributes, e.g., altering object shape instead of the engraved word.

These results suggest that remaining errors stem from limitations in object-level grounding, insufficient disentanglement of semantic directions, and challenges in faithfully controlling discrete textual outputs within the generation process.

## E Societal Impact

Our work has both potential positive and negative societal impacts. On the positive side, cross-branch semantic steering provides a diagnostic tool for analyzing whether unified multimodal models share transferable semantic representations across understanding and generation. This may contribute to more interpretable, controllable, and reliable multimodal generation systems, and can help identify representational mismatches that lead to semantic errors or hallucinations. On the negative side, techniques that improve controllability over generated content may also be misused to manipulate visual outputs, create misleading images, or amplify harmful biases if deployed without safeguards. Therefore, we view our framework primarily as an analysis and diagnostic tool, and emphasize the need for careful evaluation, transparency, and safeguards when applying steering-based interventions in real-world systems.

# F LIMITATIONS

While our study provides evidence for asymmetric cross-branch transfer in UMMs, several limitations remain. First, our analysis focuses on a predefined set of semantic concepts constructed from contrastive pairs. Although our additional evaluation on out-of-template compositional prompts demonstrates that the extracted steering directions can generalize beyond the controlled setting, steering effectiveness decreases as prompts contain more simultaneous visual concepts. This suggests that reliably controlling target semantics under complex attribute compositions remains challenging. Moreover, our evaluation relies on automated judgment using a strong multimodal model. Although we perform human verification to ensure reliability, such evaluation may still introduce biases or miss subtle semantic variations.