Causal Semantic Steering from Numeric Feature Injection: A REAL–SHUFFLE Evaluation of Metaphor Ground Generation

Anonymous ACL submission

Abstract

Because numerical feature injections are increasingly used to guide generation, but its apparent effects can be confounded by prompt templates and marginal value statistics, we propose a simple counterfactual evaluation method to isolate the causal contribution of correct instance-value alignment. For each input, we compare REAL generations conditioned on the aligned value vector to a SHUFFLE control that preserves the same prompt format and the same marginal distribution of values while breaking alignment, and define the effect as VALUE = REAL − SHUFFLE. To quantify whether VALUE concentrates movement in the intended semantic subspace, we project embedding differences onto matched bipolar semantic axes and report two planned primary metrics, which is AbsDir, a polarity-robust magnitude measure of matched-axis attraction, and EnergyAboveChance, which measures matched-subspace energy beyond a chance baseline. Across 12 planned tests (3 injection families × 2 embedding spaces × 2 metrics) with paired inference and BH-FDR control, we find strong family-by-space heterogeneity. Injection A yields reliable matched-subspace attraction in sentence space on both metrics, while its word-space effects are weak. Injection B shows the clearest attraction in word space, with negligible effects in sentence space. In contrast, Injection C exhibits dispersion in both spaces, producing significantly negative VALUE on both metrics. These results demonstrate that alignment-driven steering can be detected causally, but its success depends on the injection family and the representational space used for evaluation.

1 Introduction

Large language models can be steered at generation time via prompts, structured control fields, or lightweight adaptation modules. In many applications, the control signal comes from external numeric features serialized into key–value prompts or injected through parameter-efficient tuning. Although this "value injection" method is popular, it remains unclear whether injected values causally steer outputs in the intended semantic direction, or whether apparent gains mainly reflect prompt-format artifacts.

This ambiguity is especially acute for short, semantically underspecified outputs such as metaphor ground generation, where the goal is to produce a brief Chinese ground phrase capturing a shared property linking tenor and vehicle. In this setting, surface-overlap metrics are weak proxies of semantic adequacy, and prompt form can strongly shape output style. A rigorous evaluation therefore needs (i) a counterfactual control that isolates the causal contribution of value–instance alignment and (ii) representation-level diagnostics that test whether generations move into a matched, interpretable semantic subspace.

We propose an evaluation protocol that meets both requirements. First, we introduce a same-template counterfactual baseline, SHUFFLE, which preserves the prompt-only exposure to the structured numeric block and the marginal distribution of injected values while breaking the instance–value pairing. Thus, we obtain a paired causal contrast method, VALUE = REAL−SHUFFLE, designed to factor out prompt-template effects and isolate alignment-driven steering. Second, we quantify targeted semantic movement by projecting prediction embeddings onto matched semantic axes defined by bipolar anchor descriptors and grouped by injection family. We evaluate steering in both word and sentence embedding spaces using polarity-robust metrics.

Therefore, applying this framework to three injection families in metaphor ground generation, we find that value injection is not uniformly effective. One family yields reliable matched-subspace concentration, another shows systematic dispersion, and a third exhibits weak or unstable effects under

current axis definitions. These findings support a intriguing view of value injection that numeric conditioning can causally steer generation, but uptake depends on the semantics of the injected factors and how those semantics are realized under strict output constraints.

2 Related Work

2.1 Controlled generation and steering

Controllable text generation (CTG) aims to steer model outputs toward desired attributes such as topic, sentiment, style, or safety while maintaining fluency. A prominent line of work achieves control via input conditioning, where the model is prompted or tagged with explicit control signals. For example, CTRL trains a conditional language model on naturally occurring control codes, enabling prompt-time steering without task-specific retraining (Keskar et al., 2019). More broadly, instruction tuning and RLHF-style alignment strengthen prompt responsiveness by optimizing models to follow user intent under natural-language instructions (Ouyang et al., 2022).

A complementary family performs control at inference time without parameter updates. Plug-and-Play Language Models (PPLM) guide generation by iteratively updating hidden activations using gradients from lightweight attribute models during decoding (Dathathri et al., 2019). Other approaches guide decoding by reweighting token probabilities with auxiliary models, such as GeDi, which uses a generative discriminator to bias generation toward target attributes (Krause et al., 2021), and DExperts, which combines a base LM with expert and anti-expert models at decoding time to encourage desired attributes while suppressing undesired ones (Liu et al., 2021). Recent work on multi-attribute steering further suggests that uptake can vary across control dimensions, motivating structured analyses rather than assuming uniform benefits across factors (Nguyen et al., 2025) .

Despite strong progress, steering evaluations often emphasize output-level success and provide limited evidence that interventions induce targeted movement in an interpretable semantic subspace. Moreover, when control is delivered via structured prompting, observed improvements may conflate the effect of format with the effect of the instance-aligned control signal itself. Therefore, it is necessary to conduct counterfactual evaluation, breaking the correspondence between values and instances while preserving the prompt structure, and perform diagnostic analysis to distinguish the alignment-driven effects from confounding factors such as templates and marginal distributions (Im and Li, 2025).

2.2 Feature injection and representation intervention

A second line of work conditions models using external features or values via parameter-efficient adaptation and modular conditioning. Parameter-efficient fine-tuning (PEFT) methods update only a small subset of parameters while keeping the backbone frozen, offering an attractive mechanism for injecting side information. Adapters insert lightweight trainable modules into Transformer layers to support efficient transfer, while prefix tuning learns continuous virtual tokens that steer attention and generation without full fine-tuning (Houlsby et al., 2019; Li and Liang, 2021). LoRA injects low-rank trainable updates into the weight matrices, achieving powerful adaptability with minimal trainable parameters and little inference overhead (Hu et al., 2022).

In practice, feature injection is also implemented as feature-to-text conditioning that structured signals including numeric features are serialized into compact key–value fields and placed into a fixed instruction template. This strategy is appealing because it is model-agnostic and easy to scale, but it introduces a persistent evaluation confound that gains may arise from the presence of a structured conditioning block rather than the correct alignment between values and instances.

Beyond weight updates and prompt conditioning, a growing body of work studies representation-level interventions that steer generation by modifying internal activations at inference time (Turner et al., 2023; Wehner et al., 2025). Recent results show that activation steering can improve instruction-following behavior (Stolfo et al., 2024), and that certain activation-space interventions can transfer across models (Oozeer et al., 2025). These developments further motivate evaluating steering in representation space to test whether a value-driven intervention produces a systematic shift in the output along meaningful directions.

Across these strands, a recurring weakness is the lack of counterfactual controls that keep the same template and the same marginal distribution of injected values while breaking instance-level alignment. This motivates a same-template, distribution-

matched counterfactual that can separate template effects from the causal impact of value–instance alignment.

2.3 Measuring semantic drift with embedding projections and interpretable axes

Semantic drift is often operationalized geometrically by embedding words or texts and measuring movement between conditions in vector space. In diachronic semantics, embedding-based drift has been used to quantify meaning change across corpora and time, and survey work consolidates best practices for alignment and drift measurement (Hamilton et al., 2016; Kutuzov et al., 2018; Rudolph and Blei, 2018).

To make semantic drift easier to interpret, many studies project the embedding vectors onto semantic directions defined by a set of anchor points. A standard strategy constructs a direction as the difference between the centroids of two anchor groups, enabling interpretable projection scores. Direction-based probing has been widely used to quantify associations and to perform statistically grounded tests of differential association using permutation testing (Bolukbasi et al., 2016; Caliskan et al., 2017). Building upon this idea, SemAxis proposes a lightweight framework that characterizes domain-specific semantics using many anchor-defined axes beyond sentiment, and related work extends axis-based analysis to framing at the document level (An et al., 2018; Kwak et al., 2021). Projection methods naturally extend to sentence space via sentence encoders such as SBERT (Reimers and Gurevych, 2019), enabling drift analysis directly on short generated outputs rather than only on isolated lexemes.

Axis-based projections are commonly used for interpretation or descriptive comparison, but they are rarely paired with counterfactual controls that identify whether observed movement is driven by value–instance alignment versus prompt artifacts. In our setting, we use matched semantic axes as a causal probe by estimating VALUE with a same-template, distribution-matched SHUFFLE baseline. This turns interpretable axes from a post-hoc explanatory tool into a falsifiable diagnostic for whether injection induces targeted subspace attraction beyond chance.

3 Method

3.1 Task and data

We study Chinese metaphor ground generation. Each instance consists of a tenor t, a normalized vehicle lexeme v, and a short context c. The model generates a brief ground phrase g that states an abstract property shared by tenor and vehicle in the given context:

$$g \sim p_\theta(\cdot \mid t, v, c, optionalinjection). \quad (1)$$

Because grounds are intentionally short and semantically underspecified, we enforce a strict output constraint to reduce stylistic variance across conditions. The model must output a single Chinese phrase of at most 10 characters with no punctuation and no additional explanation, wrapped in the tag format `<ANS>... </ANS>`. During post-processing, we extract the first valid span inside the tags; if the span exceeds 10 characters, we truncate it to 10 characters.

We start from a sentence-level Chinese metaphor dataset in which each instance contains an identifier, context, annotated tenor, normalized vehicle, and a gold ground phrase. We canonicalize the schema to these core fields and normalize surface forms to ensure consistent matching across pipeline stages.

To support controlled evaluation, we restrict the dataset to a fixed whitelist of 112 vehicle lexemes and sample 15 instances per vehicle, yielding a balanced inventory in which each vehicle contributes the same number of contexts. A key design requirement is to prevent leakage from vehicle-level injections: because injected numeric values are constant for a given vehicle, the split must be vehicle-disjoint. We therefore partition vehicles into 58/14/40 vehicles for train/validation/test, respectively, and assign all instances of a vehicle to the same split, producing 870/210/600 instances. The split is fixed by seed and exported as explicit vehicle lists so that all experimental conditions share identical instance inventories.

3.2 Generation setup

We use a locally hosted Qwen3-8B model as the base generator and perform parameter-efficient adaptation separately for each injection family. For efficiency, the base model is loaded with 4-bit quantization and FP16 computation.

We fine-tune the model with LoRA using rank $r = 16$, scaling $\alpha = 32$, and dropout 0.05. LoRA adapters are applied to attention and feed-forward

projection matrices, including $q/k/v/o$ projections and the gate/up/down projections. Training uses supervised fine-tuning: the prompt is concatenated with a tagged target answer `<ANS>... </ANS>`, and loss is computed only on the answer segment by masking prompt tokens.

At inference time, we use deterministic decoding to reduce run-to-run variance in the paired REAL−SHUFFLE contrast, adopting greedy decoding with $T = 0$ (no sampling) and a short maximum generation length. Outputs are post-processed by extracting the `<ANS>` span, stripping punctuation, and truncating to 10 characters.

3.3 Numeric injection families and counterfactual conditions

We evaluate three families of numeric injections including A, B, C, each mapped to a fixed subset of vehicle-level visual factors. Family A uses F_1–F_6 (brightness, colorfulness, chromatic contrast, hue diversity, color-distribution complexity, texture roughness); family B uses F_7–F_{12} (verticality, centrality, balance, containment, directionality, proximity); and family C uses F_{13}–F_{16} (sharpness, complexity, salience, coherence). At inference time, selected factors are verbalized as a compact `key=value` list under an `[INJECTION]` block with human-readable names.

For each metaphor instance, we attach its vehicle factor vector by merging a vehicle-level factor table into the sentence-level dataset via `vehicle_norm`. All instances sharing the same vehicle therefore receive the same injected vector, motivating vehicle-disjoint splits to avoid leakage across train and test.

Each factor is standardized within the split and rounded to three decimals to keep the injected text stable and comparable across conditions. The prompt defines the ground as a shared property that must apply to both tenor and vehicle in context.

Our main analysis isolates alignment-driven effects by comparing two generations produced under the same prompt template and the same family-specific adapted checkpoint. REAL uses the instance-aligned value vector, while SHUFFLE permutes value vectors within the same split and family, preserving the marginal value distribution but breaking instance–value alignment. Next, we define the paired effect as VALUE = REAL – SHUFFLE and evaluate it in embedding space.

We additionally report two non-causal baselines to diagnose prompt-only effects. NONE removes the injection block. PROMPT includes the same structured numeric block with aligned values but uses the base checkpoint without parameter adaptation. These baselines are used as sanity checks and are not part of the planned REAL–SHUFFLE causal claims.

3.4 Representation spaces

To evaluate semantic steering in representation space, we embed each generated ground using a fixed embedding backbone. From the same forward pass we extract two representations. Word space is a token-mean pooled embedding, computed by averaging token hidden states under the attention mask. Sentence space is represented by the final-layer hidden state at sequence position 0, which serves as a first-position token summary. Embeddings are cached in raw form, and normalization is applied only in downstream similarity and projection computations when needed.

3.5 Semantic axes and matched subspaces

We construct interpretable semantic axes using bipolar anchor descriptors. We define $K = 8$ axes (A1–A3, B1–B3, C1–C2). Each axis k is specified by a positive anchor set P_k and a negative anchor set N_k, with 8 anchors per pole (16 anchors per axis, 128 anchors total). Axes are constructed separately in word space and sentence space using the corresponding anchor embeddings.

Our default axis estimator is the centroid-difference direction

$$\mathbf{v}_k \;=\; norm\Big(\,\mu(P_k) - \mu(N_k)\,\Big), \qquad (2)$$

where $\mu(\cdot)$ denotes the mean embedding and $norm(\mathbf{x}) = \mathbf{x}/\mathbf{x}_2$.

As a robustness variant for small anchor sets, we also build an alternative axis direction using shrinkage-LDA, computing a direction proportional to

$$\tilde{\mathbf{v}}_k \;\propto\; \hat{\Sigma}^{-1}\Big(\,\mu(P_k) - \mu(N_k)\,\Big), \qquad (3)$$

followed by unit normalization, where $\hat{\Sigma}$ is a shrinkage estimate of the within-class covariance.

Axes are grouped to correspond to injection families. For each family we define a matched axis set M, and interpret effects as preferential movement within this matched subspace rather than arbitrary drift across all axes.

3.6 Directional metrics and inference

For each instance i, let $\mathbf{e}_i^{REAL}$ and $\mathbf{e}_i^{SHUFFLE}$ be the normalized embeddings of its generated ground

in a given space (word or sentence). We define the alignment-driven drift vector as the paired difference

$$\delta_i \;=\; \mathbf{e}_i^{REAL} - \mathbf{e}_i^{SHUFFLE}. \tag{4}$$

Given axis vectors $\{\mathbf{v}_k\}_{k=1}^{K}$, we compute per-axis projections

$$p_{ik} \;=\; \delta_i^{\top}\mathbf{v}_k. \tag{5}$$

We evaluate whether drift concentrates in the matched family subspace M versus the complement $O = \{1, \ldots, K\} \setminus M$ using two planned primary metrics designed to be robust to axis polarity.

AbsDir indicates matched-subspace concentration in magnitude.

$$AbsDir_i = \frac{1}{|M|}\sum_{k\in M}|p_{ik}| \;-\; \frac{1}{|O|}\sum_{k\in O}|p_{ik}|. \tag{6}$$

Positive values indicate that VALUE drift magnitude is more concentrated in the matched subspace than in unmatched axes; negative values indicate relative dispersion away from the matched axes.

EnergyAboveChance. We compute the fraction of squared projection energy that lies in the matched subspace and subtract a chance baseline

$$r_i = \frac{\sum_{k\in M} p_{ik}^2}{\sum_{k=1}^{K} p_{ik}^2}, \mathrm{EAC}_i = r_i - \frac{|M|}{K}. \tag{7}$$

EnergyAboveChance > 0 indicates that alignment causes drift energy to concentrate on matched semantics beyond chance; negative values indicate systematic dispersion.

All hypothesis tests are performed on paired instance-level VALUE samples. For each family and embedding space, we compute instance-wise AbsDir and EnergyAboveChance and test whether the mean paired effect differs from zero. We report (i) bootstrap 95% confidence intervals for the mean and (ii) a paired sign-flip permutation test to obtain a two-sided p-value.

To control false discoveries under multiple testing, we apply Benjamini–Hochberg FDR correction over the planned primary family of 12 tests

$$3\ families \times 2\ spaces \times 2\ metrics \;=\; 12. \tag{8}$$

In the main results table we report raw p-values together with FDR-adjusted q-values; unless stated otherwise, significance claims refer to the q-values. Figures visualize the same 12 planned contrasts as effect sizes with bootstrap confidence intervals and significance markers derived from the FDR-adjusted q-values.

4 Results

This section reports the causal effect of numeric feature injection on semantic steering using the REAL–SHUFFLE design. For each prompt instance, REAL uses the correct value vector aligned to the vehicle, while SHUFFLE preserves the same prompt template and marginal value statistics but breaks instance–value correspondence. We define the treatment effect as VALUE=REAL-SHUFFLE, computed in embedding space for each generated ground. Then we test whether VALUE concentrates movement in the matched semantic subspace using two planned primary metrics, AbsDir and EnergyAboveChance. As a design check for template-driven effects, we also compute PROMPT–NONE as a secondary diagnostic of the prompt-only exposure to the structured numeric block, while the main claims rely on the planned REAL–SHUFFLE causal contrast. All results use paired inference over instances and apply BH-FDR correction across the 12 planned tests.

4.1 Main VALUE Effects on Matched-Subspace Steering

Table 1 reports the numerical VALUE effects and FDR-adjusted q-values. The results show a consistent family-by-space dissociation rather than a uniform improvement. Injection A yields reliable matched-subspace concentration in sentence space on both metrics (AbsDir $q < .001$; EnergyAboveChance $q = .001$), while its word-space effects are small and not reliable after correction. In contrast, Injection B shows the clearest concentration in word space, with significant positive effects on both AbsDir and EnergyAboveChance (both $q < .001$), whereas its sentence-space counterparts remain near zero and non-significant. Finally, Injection C exhibits negative VALUE effects in both spaces, indicating dispersion away from matched C-subspace concentration. AbsDir is significantly below zero in word and sentence space (both $q = .003$), and EnergyAboveChance is also significantly negative in both spaces ($q = .013$ in word space; $q < .001$ in sentence space). In summary, these planned tests establish that VALUE effects depend strongly on the injection family and the representation space used for evaluation.

Figure 1 highlights a pronounced family-by-space dissociation. Injection A shows positive matched-subspace concentration primarily in sentence space, whereas Injection B shows its strongest

Table 1: Planned VALUE effects (REAL−SHUFFLE) on AbsDir and EnergyAboveChance with 95% bootstrap CIs. p values are from paired sign-flip permutation tests; q values are BH-FDR-adjusted over 12 planned tests.

Space	Inj	Metric	Mean	CI_{95} Lo	CI_{95} Hi	p	q
sent	A	AbsDir	0.003	0.001644	0.004681	$< .001$	$< .001$
sent	A	EnergyAboveChance	0.027	0.013426	0.041336	$< .001$	.001
sent	B	AbsDir	0.001	-0.000480	0.002372	.210	.280
sent	B	EnergyAboveChance	0.002	-0.012350	0.015983	.785	.785
sent	C	AbsDir	-0.002	-0.003810	-0.000920	.002	.003
sent	C	EnergyAboveChance	-0.026	-0.037540	-0.014100	$< .001$	$< .001$
word	A	AbsDir	-0.001	-0.003150	0.001447	.449	.539
word	A	EnergyAboveChance	0.003	-0.013920	0.020134	.745	.785
word	B	AbsDir	0.008	0.005519	0.010036	$< .001$	$< .001$
word	B	EnergyAboveChance	0.051	0.034727	0.066623	$< .001$	$< .001$
word	C	AbsDir	-0.002	-0.003960	-0.000880	.002	.003
word	C	EnergyAboveChance	-0.015	-0.026450	-0.004160	.008	.013

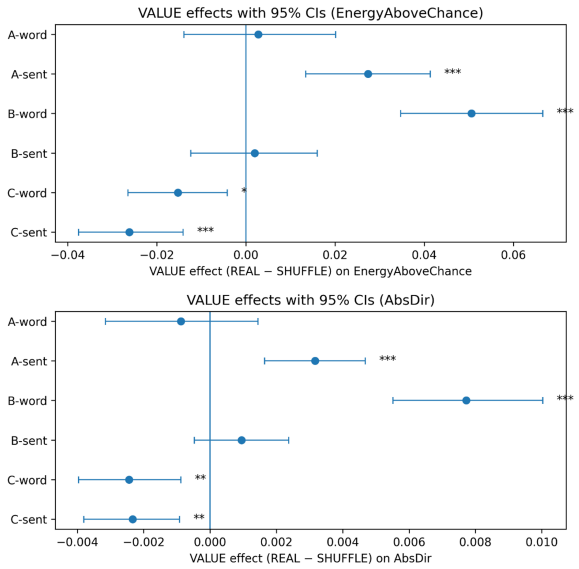

Figure 1: VALUE effects (REAL–SHUFFLE) on AbsDir and EnergyAboveChance.

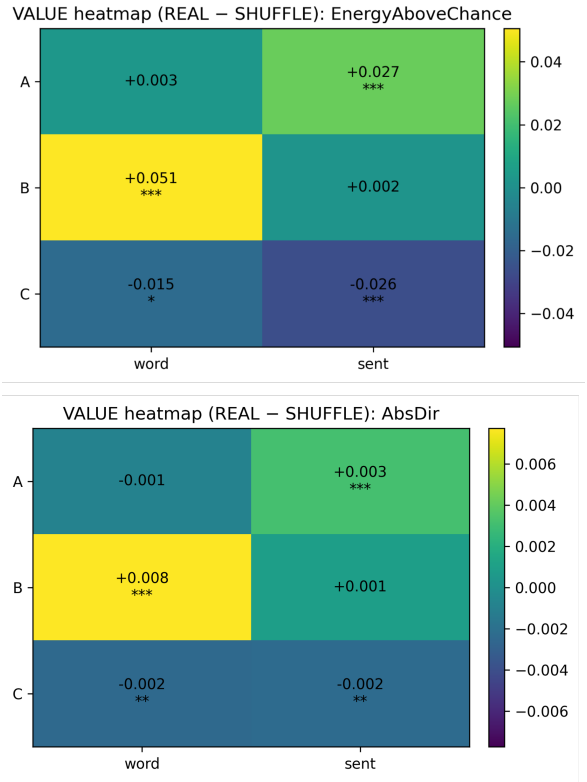

Figure 2: Figure 2 Family-by-space interaction for VALUE effects.

positive effects in word space. Injection C shows negative shifts in both spaces, consistent with dispersion away from matched C-subspace concentration, with the largest negative effect observed in the EnergyAboveChance metric in the sentence space.

4.2 Patterns across injection families and embedding spaces

Figure 2 summarizes the family-by-space interaction behind the main effects in Table 1. The heatmap reveals three regimes: family A concentrates effects in sentence space, family B concentrates effects in word space, and family C shows negative shifts in both spaces. Because all cells report REAL−SHUFFLE, these contrasts isolate alignment-driven effects while controlling for prompt form and the marginal value distribution.

Injection A is reliably positive in sentence space on both AbsDir and EnergyAboveChance, but its word-space effects are weak. This pattern is consistent with the observation that A primarily alters the overall semantics of the generated text, rather than producing stable word-level changes that can withstand pooling operations. In contrast, Injection B is strongly positive in word space but near zero in sentence space, suggesting an effect that is more visible in lexicalization-sensitive representations than in sentence-level aggregates.

Finally, Injection C is negative in both spaces, indicating reduced concentration of drift in the matched C-subspace. Since AbsDir and EnergyAboveChance are polarity-robust primary metrics, negative values reflect under-concentration in the matched axes and, for EnergyAboveChance, below-chance matched energy, rather than a signed reversal along any specific axis. This pattern differs from near-zero cells and is consistent with systematic mis-targeting rather than simple failure of uptake.

4.3 Sample-level distributions and robustness

To determine whether the aggregate VALUE effects in Table 1 reflect broad, population-level shifts or are driven by a small number of extreme samples, we examine sample-level VALUE distributions for the two primary metrics in Figure 3. Each panel summarizes the distribution of per-sample VALUE using the median, interquartile range, and 5–95% interval, allowing us to inspect both central tendency and dispersion under the paired design.

The distributional patterns align with the main effects and suggest that the significant results are not outlier-driven. For Injection A, the sentence-space distributions show a clear right-shift on both AbsDir and EnergyAboveChance, consistent with reliable matched-subspace attraction, whereas the word-space distributions are centered close to zero and show no systematic shift. For Injection B, the strongest right-shift appears in word space, again on both metrics, while the sentence-space distributions remain tightly centered near zero. For Injection C, both spaces exhibit left-shifts, indicating systematic dispersion away from matched-subspace concentration across a broad portion of samples rather than isolated failures.

We further assess robustness to modeling choices in axis construction by comparing alternative axis estimators used in the pipeline. The family-by-space qualitative pattern remains stable under these alternatives, supporting the interpretation that the observed dispersion effects reflect genuine differences in how injection families interact with representational spaces, rather than being an artifact of the specific axis fitting process.

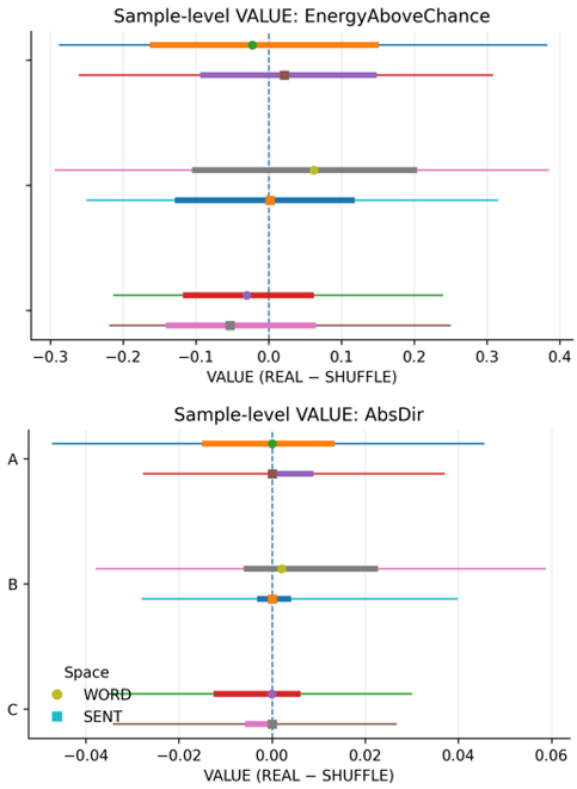

Figure 3: Sample-level VALUE effects (REAL−SHUFFLE) for EnergyAboveChance and AbsDir.

5 Discussion

Using the REAL–SHUFFLE counterfactual, we find that value-conditioned steering is heterogeneous and representation-dependent (Table 1; Fig. 1). Injection A concentrates movement in sentence space, Injection B in word space, and Injection C shows systematic dispersion across both spaces. Below we relate these patterns to prior work on controllable generation and representation-level diagnostics.

5.1 Why attraction depends on embedding space

The A/B dissociation suggests that "steering" can surface at different linguistic granularities: A is most visible in sentence embeddings, while B is most visible in word-level pooling (Table 1). This is consistent with the controllable generation lit-

erature's distinction between global semantic control and more local lexical control (Zhang et al., 2023). This also reminds us not to evaluate the prompting effect solely in a single representation space, since sentence encoders may down-weight lexical substitutions that nonetheless reflect control uptake, while token-level pooling may overemphasize local changes without guaranteeing coherent sentence-level movement (Reimers and Gurevych, 2019). Mechanistically, our results are consistent with the view that different control signals couple to different generation pathways, as seen in decoding-time controllers that can bias word choice and attribute markers with variable effects on global semantics (Dathathri et al., 2019; Krause et al., 2021).

5.2 Why aligned injection can systematically disperse

Injection C exhibits reliable negative effects on polarity-robust metrics in both spaces, indicating systematic under-concentration in the matched subspace rather than mere "no uptake" (Table 1). This supports prior observations that control signals can backfire when the intended semantics of the control variable are entangled or misinterpreted by the model (Liu et al., 2021). One plausible explanation is semantic mismatch between the injected numeric features and the evaluation axes: learned associations can be structured yet opaque, similar to how embedding spaces encode stable, unintended associations and biases (Bolukbasi et al., 2016; Caliskan et al., 2017). The fact that dispersion persists across spaces further aligns with recent work on representation or activation steering showing that interventions can yield robust but mis-targeted shifts when directions are not cleanly aligned with the desired semantics (Turner et al., 2023).

5.3 Implications for evaluation of value-conditioned steering

Methodologically, these results argue for counterfactual baselines and interpretable diagnostics in controllable generation (Zhang et al., 2023). The REAL–SHUFFLE contrast isolates alignment-driven effects from template and marginal-distribution confounds, while matched-axis measures connect evaluation to interpretable semantic directions (An et al., 2018; Kwak et al., 2021). At the same time, the family-by-space interaction and C-dispersion highlight that axis-based conclusions depend on representation choice and axis separability, echoing broader cautions in embedding-based measurement and semantic drift (Hamilton et al., 2016; Kutuzov et al., 2018; Rudolph and Blei, 2018). Practically, robust steering claims should (i) report multiple embedding spaces and (ii) treat axis definitions as a robustness dimension rather than a fixed ground truth.

6 Conclusion

This paper introduced a counterfactual framework for evaluating semantic steering from numeric feature injection that separates alignment-driven effects from prompt-template and marginal-distribution confounds. In this paper, we use the REAL-SHUFFLE comparison method to measure pairwise effects in the embedding space and evaluate whether these effects are concentrated in matching semantic subspaces using the AbsDir and EnergyAboveChance metrics. The planned tests reveal a stable family-by-space dissociation. Injection A produces reliable attraction in sentence space, Injection B produces reliable attraction in word space, and Injection C produces robust dispersion in both spaces. Beyond establishing a causal signal of steering, the dispersion pattern highlights that aligned numeric conditioning can systematically backfire, making counterfactual controls essential for diagnosing failure modes. Future work should test robustness under external embedding backbones and connect subspace steering more directly to task quality through paired human judgments of REAL versus SHUFFLE outputs.

7 Limitations

Several limitations constrain interpretation and suggest direct extensions. First, the evaluation relies on a particular embedding space; confirming the family-by-space pattern under at least one external embedding backbone would strengthen robustness claims. Second, the matched axes are anchor-defined and may encode design choices; reporting per-axis effects and sensitivity to anchor selection would clarify how much of the signal is driven by a subset of axes. Third, while the causal contrast demonstrates subspace steering, it does not by itself establish that REAL outputs are better grounds; paired human judgments comparing REAL versus SHUFFLE on applicability and plausibility would connect steering to task quality. Finally, dispersion results prompted us to conduct more targeted diagnostics, such as testing alternative value normalizations, sign conventions, or training objectives

that explicitly align injected features with axis semantics. These extensions would help distinguish whether dispersion reflects feature–axis mismatch, model miscalibration, or a deeper limitation of numeric injection as a control mechanism. More broadly, value-conditioned steering methods can be misused to manipulate model outputs or amplify unintended biases, so deployment should include monitoring and safeguards.

Acknowledgements

Generative AI tools were used solely for language editing (grammar, phrasing, and readability) of the authors' original writing. The tools were not used to generate new scientific content or ideas. All methods, results, and interpretations were produced and checked by the authors.

References

Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2018. Semaxis: A lightweight framework to characterize domain-specific word semantics beyond sentiment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2450–2461. Association for Computational Linguistics.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29, pages 4349–4357.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*. ArXiv:1912.02164.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzębski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Shawn Im and Yixuan Li. 2025. A unified understanding and evaluation of steering methods. *arXiv preprint arXiv:2502.02716*.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952. Association for Computational Linguistics.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: A survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397. Association for Computational Linguistics.

Haewoon Kwak, Jisun An, Eunsol Jing, and Yong-Yeol Ahn. 2021. Frameaxis: Characterizing framing bias and intensity with word embedding. *PeerJ Computer Science*, 7:e644.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Giang Nguyen, Cheng Li, Hao Peng, Shizhe Diao, Yiming Zhang, Hao Zhang, Yoon Kim, Ana Marasović, and Malihe Alikhani. 2025. Multi-attribute steering of language models via targeted intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9291–9317, Vienna, Austria. Association for Computational Linguistics.

Narmeen Fatimah Oozeer, Dhruv Nathawani, Nirmalendu Prakash, Michael Lan, Abir Harrasse, and Amir Abdullah. 2025. Activation space interventions can be transferred between large language models. *arXiv preprint arXiv:2503.04429*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992. Association for Computational Linguistics.

Maja Rudolph and David Blei. 2018. Dynamic embeddings for language evolution. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. ACM.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. 2024. Improving instruction-following in language models through activation steering. In *International Conference on Learning Representations (ICLR)*.

Alexander M. Turner, Leo Thiergart, Gabriel Leech, David Udell, Juan J. Vazquez, Unnat Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Jan Wehner, Sahar Abdelnabi, Daniel Tan, David Krueger, and Mario Fritz. 2025. Taxonomy, opportunities, and challenges of representation engineering for large language models. *Transactions on Machine Learning Research*. Published in TMLR (09/2025).

Ximing Zhang, Yulin Tian, Xinyue Chen, Sujian Li, and Ming Zhou. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*.