SteerForce at SemEval-2026 Task 11: Reducing Content Effects Using Layered Activation Steering

Noah Tratzsch[1], **Asmaa Al-Raian**[1], **Mounika Marreddy**[1], **Alexander Mehler**[1]
[1]Goethe University, Frankfurt am Main, Germany
ntratzsch@stud.uni-frankfurt.de, s6380199@rz.uni-frankfurt.de
mmarredd@em.uni-frankfurt.de, mehler@em.uni-frankfurt.de

Abstract

Large language models exhibit content effects, where surface plausibility interferes with formal logical reasoning. In SemEval-2026 Task 11, this appears as a performance gap between plausibility-aligned and plausibility-conflicting syllogisms, reflecting directional content bias. We address this issue using inference-time activation steering, modeling bias as a geometric deviation between plausibility-driven and validity-driven representations. We introduce a layered steering framework that combines Activation Transport (ACT) with input-adaptive contrastive steering (K-CAST), applied to layers identified through sensitivity analysis. This architecture-aware strategy enables targeted interventions without retraining.

On BERT, sequential multi-layer steering improves validity accuracy from 77.1% to 82.3% while reducing bias by 75%. In contrast, for the decoder-only Qwen2.5-1.5B-Instruct, a single mid-to-late layer intervention reduces bias from 0.26 to 0.04 with modest accuracy gains, whereas multi-layer steering offers no additional benefit. These results reveal a fundamental architectural distinction: encoder-based models benefit from distributed low-intensity corrections, while decoder-only instruction-tuned models concentrate reasoning signals within a narrow late-layer band. Our findings demonstrate that effective bias mitigation requires architecture-aware activation steering.

1 Introduction

Large Language Models (LLMs), including bidirectional architectures such as BERT (Devlin et al., 2019), achieve strong performance across language understanding and reasoning tasks. However, they often exhibit *content effects*, where surface plausibility or learned heuristics interfere with formal logical reasoning (Valentino et al., 2025). This results in systematic performance gaps between belief-consistent and belief-conflicting problems.

This limitation is central to SemEval-2026 Task 11 (Valentino et al., 2026), which requires predicting the logical validity of natural-language syllogisms independently of their real-world plausibility. The task exposes how models may rely on semantic shortcuts rather than abstract logical structure.

While fine-tuning with additional supervision can partially mitigate such effects (Sun et al., 2020), it is computationally costly and does not directly control internal reasoning representations. In contrast, recent work shows that model behavior can be modified at inference time through activation steering, which manipulates latent representations within transformer layers (Rimsky et al., 2024).

We frame content bias as a geometric deviation in hidden representation space and propose a layered steering framework that combines Activation Transport (ACT) (Rodriguez et al., 2024) with semantics-adaptive contrastive steering (K-CAST) (Wang et al., 2025). This sequential approach stabilizes global representations while applying input-specific corrections.

Crucially, we show that steering effectiveness depends on model architecture. Encoder-based models benefit from low-intensity sequential multi-layer interventions, reflecting distributed reasoning signals. In contrast, decoder-only instruction-tuned models concentrate bias-sensitive representations within a narrow late-layer band, where a single well-placed intervention suffices. These results highlight the need for architecture-aware activation steering to improve logical robustness without additional fine-tuning.

2 Methodology

Our approach follows a structured analysis-and-intervention pipeline. We first establish encoder-based and decoder-based baselines, extract internal representations, identify layers that are sensitive to content bias, and then apply targeted activation

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1050–1055

steering interventions. Our methods build on prior work on contrastive activation steering and activation transport (Valentino et al., 2025; Rodriguez et al., 2024; Panickssery et al., 2024).

2.1 Encoder-Based Model

For encoder-based experiments, we fine-tune `bert-base-uncased` using a dual-head classification architecture for validity and plausibility prediction (Devlin et al., 2019; Sun et al., 2020). Both heads operate on the shared encoder representation of the `[CLS]` token.

This setup intentionally exposes the encoder to both formal logical signals and heuristic plausibility cues. As a result, the latent space captures both reasoning-relevant and plausibility-driven information, providing a controlled setting in which content bias can be analyzed. At inference time, only the validity head is used.

2.2 Decoder-Based Models

For decoder-only models, we frame validity prediction as an instruction-following task. We evaluate instruction-tuned models, namely TinyLlama and Qwen (Zhang et al., 2024; Qwen et al., 2025).

No task-specific fine-tuning is performed. Instead, all modifications are applied at inference time by manipulating internal activations. This allows us to study and influence reasoning behavior without updating model parameters.

For TinyLlama, ACT is implemented as a projection-based geometric correction along the steering direction rather than full activation transport with learned scale and shift parameters. This formulation corresponds to removing the bias-aligned component of the hidden representation scaled by λ. The overall steering framework remains unchanged. In TinyLlama, the steering direction and kNN gating are computed using validity-labeled training activations.

2.3 Hidden State Extraction

For layer-wise analysis and intervention, we extract hidden representations from all transformer layers. For each input, we record a single layer representation: the `[CLS]` token for BERT and the final token for decoder-only models. These representations form the basis for both sensitivity analysis and activation steering.

2.4 Layer Sensitivity Analysis

To identify layers that contribute most to directional content bias, we perform a layer sensitivity analysis. We apply a lightweight test intervention independently at each layer and measure the resulting change in accuracy and bias.

Consistent with prior activation-steering findings (Valentino et al., 2025), we observe that steering effects are negligible in early layers, peak in upper-middle layers, and become unstable in the final layers. Subsequent interventions are therefore restricted to the most sensitive layers.

2.5 Global Contrastive Steering

As a baseline, we implement global contrastive steering following contrastive activation addition (Panickssery et al., 2024). We compute a steering direction in activation space as:

$$\Delta\phi = \mu^{+} - \mu^{-}, \tag{1}$$

where μ^{+} and μ^{-} denote the mean hidden representations of plausibility-aligned and plausibility-conflicting samples, respectively. The direction $\Delta\phi$ is added to all test representations with a fixed steering strength λ.

Although this approach can reduce bias, it applies the same correction to every input and often requires larger λ, which may negatively affect accuracy.

2.6 K-CAST and ACT

K-CAST. K-CAST introduces input adaptivity by applying contrastive steering only when a test sample is assigned to a specific regime using a k-nearest-neighbor (kNN) lookup in activation space (Valentino et al., 2025). The memory bank consists of hidden representations from the training set. The parameter K controls the locality of the intervention: smaller values yield highly input-specific corrections, while larger values produce smoother but less discriminative adjustments.

ACT. Activation Transport (ACT) performs a smooth geometric transformation of hidden states toward a reference activation distribution (Rodriguez et al., 2024). For a hidden activation a, the transported representation is defined as:

$$T(a;\lambda) = (1-\lambda)a + \lambda(\omega a + \beta), \tag{2}$$

where $\lambda \in [0,1]$ controls intervention strength, and ω (scale) and β (shift) are layer-specific parameters computed from training activation statistics.

This interpolation enables controlled, low-intensity updates without abrupt changes to the representation.

Steering parameters λ and K are selected empirically based on validation performance. We perform layer-wise tuning guided by sensitivity analysis. We observe that larger values of K tend to dilute the corrective signal, while excessively large λ values may suppress useful reasoning features.

2.7 K-ACT: Combined Steering

We combine ACT and K-CAST into a two-stage procedure termed **K-ACT**. First, ACT defines a transported target representation. Second, the resulting transport delta $(T(a;\lambda)-a)$ is applied only to samples selected by the K-CAST gating mechanism.

This combination integrates global stabilization with input-specific correction and allows effective bias mitigation at lower steering strengths (Valentino et al., 2025; Rodriguez et al., 2024).

2.8 Single-Layer and Sequential Interventions

We first evaluate all steering methods independently at each sensitive layer to assess their isolated effects. We then explore sequential steering across multiple layers.

In encoder-based models, distributing low-intensity interventions across adjacent sensitive layers provides the best trade-off between accuracy and bias reduction. In contrast, for decoder-only instruction-tuned models, a single well-placed intervention typically captures most of the achievable bias reduction, and additional layers offer limited benefit.

3 Experimental Setup

We evaluate activation steering on both encoder- and decoder-based architectures. For the encoder setting, we fine-tune `bert-base-uncased` (Devlin et al., 2019) using a dual-head classification architecture. For decoder-only instruction-tuned models (TinyLlama and Qwen), no task-specific fine-tuning is performed. Instead, validity prediction is formulated as an instruction-following task, and all steering interventions are applied exclusively at inference time.

3.1 Dataset

We use the official dataset released for SemEval-2026 Task 11, Subtask 1 (SemEval-2026 Task 11 Organizers, 2026). The dataset consists of English natural-language syllogisms annotated with two binary labels: *validity* (formal logical correctness) and *plausibility* (real-world believability). The objective is to predict logical validity independently of plausibility effects.

The training set contains approximately 800 instances. Table 1 shows a representative example illustrating the potential conflict between logical validity and surface plausibility.

Syllogism
Not all canines are aquatic creatures known as fish. It is certain that no fish belong to the class of mammals. Therefore, every canine falls under the category of mammals.
Validity: false **Plausibility:** true

Table 1: Example illustrating the distinction between logical validity and plausibility.

3.2 Evaluation Metrics

Systems are evaluated using the official SemEval metric, which measures accuracy over binary validity labels. Since classes are balanced, accuracy serves as the primary evaluation metric. Plausibility labels are not used at test time and function only as auxiliary supervision for the encoder-based model.

To quantify susceptibility to plausibility heuristics, we additionally report *Directional Content Bias*, defined as the absolute difference in validity accuracy between plausibility-aligned and plausibility-conflicting subsets:

$$\text{Bias} = \left|\text{Acc}_{\text{plaus}} - \text{Acc}_{\text{implaus}}\right|. \tag{3}$$

Lower values indicate greater robustness to content effects.

3.3 Experimental Environment

Experiments are conducted using `bert-base-uncased` (Devlin et al., 2019), `TinyLlama-1.1B-Chat-v1.0` (Zhang et al., 2024), and `Qwen2.5-1.5B-Instruct` (Qwen et al., 2025). All models are implemented in PyTorch with HuggingFace Transformers. We fix random seeds for reproducibility.

3.4 Multi-Head Training (Encoder Model)

Following Sun et al. (2020), we attach two independent classification heads to the shared [CLS] representation of BERT: **Validity head**: predicts

formal logical correctness and **Plausibility head**: predicts empirical believability.

This dual-head design encourages the encoder to encode both logical and heuristic signals in its latent space, intentionally inducing representational overlap. This controlled semantic interference provides the setting in which activation steering aims to disentangle plausibility-driven and validity-driven representations (Valentino et al., 2025).

4 Results

4.1 Layer Sensitivity Analysis

Across architectures, steering effects are negligible in early layers, increase toward intermediate-to-late layers, and either peak sharply (decoder models) or gradually accumulate (encoder models).

For BERT, sensitivity increases steadily from lower to higher layers, reaching its maximum in the final layers (8–11). This smooth upward trend suggests that logical and plausibility-related signals remain distributed across multiple upper layers. No single layer dominates the bias effect; instead, the influence of plausibility appears to accumulate progressively throughout the encoder stack.

In contrast, Qwen exhibits a highly non-monotonic profile. Sensitivity remains low in early layers, then rises sharply around layers 15–17, forming a narrow peak, before fluctuating in later layers. This concentrated spike indicates that bias-relevant reasoning signals are localized within a restricted band of upper-middle layers. The sharper peak compared to BERT suggests a more centralized representation of decision-critical features in decoder-only architectures.

This architectural contrast already anticipates the downstream steering behavior: distributed sensitivity in BERT should favor multi-layer interventions, whereas localized sensitivity in Qwen should benefit from single-layer correction.

4.2 Content Bias and Accuracy

BERT (Encoder-Based). The baseline BERT model achieves 0.7708 accuracy with a directional bias of 0.0833. Single-layer steering at layer 11 drastically reduces bias (0.0104) but slightly decreases accuracy (0.7604), suggesting that aggressive correction at a single late layer may partially suppress useful reasoning signals.

Sequential steering across layers 8 and 9 yields the strongest overall performance: accuracy improves substantially to 0.8229 while bias remains low (0.0208). This result indicates that distributing low-intensity corrections across adjacent sensitive layers preserves logical representations while attenuating plausibility-driven deviations. In contrast, increasing the neighborhood size (High-K) weakens performance and increases bias, suggesting that overly broad contrastive neighborhoods dilute the corrective signal.

Overall, BERT benefits from coordinated multi-layer steering, consistent with its gradually distributed sensitivity profile.

Qwen (Decoder-Only). The baseline Qwen model exhibits a substantially higher bias (0.2611), indicating strong susceptibility to plausibility heuristics. However, a single K-CAST intervention at layer 21 reduces bias dramatically to 0.0409, while slightly improving accuracy (0.6875 vs. 0.6771).

Importantly, sequential steering (layers 19 + 21) fails to produce additional gains and instead restores bias to near-baseline levels (0.2599). Similarly, ACT-based smoothing at layer 23 provides no meaningful improvement. These findings suggest that once the bias-relevant representation in the dominant layer is corrected, further interventions interfere with stabilized decision signals. In decoder-only models, late-layer activations appear to play a decisive role in classification, and redundant corrections may destabilize the final representation.

TinyLlama. TinyLlama follows a pattern similar to Qwen but with lower baseline performance. The model starts at 0.6250 accuracy and 0.1469 bias. A single-layer K-ACT intervention at layer 17 improves both metrics substantially (accuracy 0.6667, bias 0.0559), confirming that bias-sensitive signals are concentrated in a narrow upper-layer band.

Sequential steering across layers 21 and 14, however, degrades accuracy sharply (0.5104) and only partially reduces bias (0.1066). The partial reversal of bias direction suggests overcorrection and interference between interventions applied at different depths. This behavior indicates that later-layer corrections dominate earlier adjustments, and that distributed interventions can distort the learned decision boundary in decoder-only architectures.

Cross-Architectural Insights. Taken together, the results reveal a clear architectural distinction:

- **Encoder-based models (BERT)** encode reasoning and plausibility signals in a distributed

Model	Configuration	Layers	Acc	Bias
BERT	Baseline	None	0.7708	0.0833
BERT	Single-Layer	11	0.7604	**0.0104**
BERT	**Sequential**	**8 + 9**	**0.8229**	0.0208
BERT	High-K Value	8 + 9	0.7812	0.0625
TinyLlama	Baseline	None	0.6250	0.1469
TinyLlama	Single-Layer (K-ACT)	17	**0.6667**	**0.0559**
TinyLlama	Sequential (K-CAST→ACT)	21 + 14	0.5104	0.1066
TinyLlama	High-K Value	21	0.5417	0.0769
Qwen	Baseline	None	0.6771	0.2611
Qwen	Single-Layer (K-CAST)	21	**0.6875**	**0.0409**
Qwen	Sequential (Hybrid)	19 + 21	0.6833	0.2599
Qwen	High-K / ACT	23	0.6792	0.2674

Table 2: Comparison of steering outcomes across models. Sequential steering provides the best trade-off for the encoder-based BERT model, while single-layer steering is more effective for the decoder-only Qwen model. Sequential steering provides the best balance between accuracy and neutrality, while overly large neighborhoods lead to signal dilution.

manner across upper layers. Consequently, sequential low-intensity steering across adjacent layers provides the best trade-off between accuracy and bias reduction.

- **Decoder-only instruction-tuned models (Qwen, TinyLlama)** concentrate bias-sensitive reasoning signals within a narrow band of late layers. In these models, a single well-placed intervention is sufficient, while multi-layer steering can introduce instability.

These findings demonstrate that activation steering is not architecture-agnostic. Its effectiveness depends critically on the representational geometry and the layer-wise concentration of reasoning signals within the underlying transformer. Aligning the steering strategy with these architectural dynamics enables substantial bias reduction without sacrificing logical accuracy and, in some cases, even leads to performance improvements.

Table 2 confirms these architecture-dependent dynamics.

Steering can become unstable in the late layers of decoder-only models, and sequential multi-layer interventions may introduce interference. Larger values of K further reduce effectiveness by diluting the corrective signal.

5 Conclusion

We show that content bias in logical reasoning can be mitigated at inference time without sacrificing performance. By modeling bias as a geometric deviation in representation space, activation steering improves logical robustness through targeted latent interventions.

Crucially, the optimal strategy depends on model architecture. Encoder-based models such as BERT benefit from low-intensity sequential steering across multiple upper layers, where reasoning signals are distributed. In contrast, decoder-only instruction-tuned models such as TinyLlama and Qwen concentrate bias-sensitive representations within a narrow late-layer band, where a single well-placed intervention is sufficient. These findings demonstrate that effective bias mitigation requires architecture-aware steering aligned with the model's internal representational structure.

6 Limitations

Our study is limited to relatively small models and a modest dataset, which may restrict generalization to larger architectures or broader reasoning tasks. Steering layers and hyperparameters are selected empirically and may require adaptation across settings. Moreover, we evaluate only a small set of encoder and decoder models, leaving the generality of the observed architectural differences open.

7 Potential Improvements

Future work could extend this approach to larger models and broader datasets to evaluate scalability. In particular, studying a wider range of decoder-only and encoder–decoder architectures would help determine whether the observed architectural differences generalize. Automating the selection of

steering layers, steering strengths, and neighborhood sizes could further reduce the need for manual tuning. Additionally, exploring adaptive or learned combinations of ACT and K-CAST may yield more stable improvements in decoder-based models.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering llama 2 via contrastive activation addition. *Preprint*, arXiv:2312.06681.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. 2024. Controlling language and diffusion models by transporting activations. *Preprint*, arXiv:2410.23054.

SemEval-2026 Task 11 Organizers. 2026. Semeval-2026 task 11 dataset. `https://github.com/neuro-symbolic-ai/semeval_2026_task_11`. Training data for Subtask 1.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification? *Preprint*, arXiv:1905.05583.

Marco Valentino, Geonhee Kim, Dhairya Dalal, Zhixue Zhao, and André Freitas. 2025. Mitigating content effects on reasoning in language models through fine-grained activation steering. *Preprint*, arXiv:2505.12189.

Marco Valentino, Leonardo Ranaldi, Giulia Pucci, Federico Ranaldi, and André Freitas. 2026. Semeval-2026 task 11: Disentangling content and formal reasoning in large language models. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*.

Weixuan Wang, Jingyuan Yang, and Wei Peng. 2025. Semantics-adaptive activation intervention for llms via dynamic steering vectors. *Preprint*, arXiv:2410.12299.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.