Persona Is Latent: Interpretable Alignment in Large Language Models

Anonymous ACL submission

Abstract

Persona-conditioned generation is a core capability of large language models, yet persona consistency degrades under increasing task complexity. Existing approaches treat persona as a surface-level behavioral constraint imposed through prompting or fine-tuning, offering limited interpretability and control. We instead advance a representational account of persona alignment, modeling persona as a latent and distributed structure within internal model representations. Through layer-wise Sparse Autoencoders and causal latent interventions, we identify persona-relevant features across model depth and show that persona signals become increasingly discriminative in deeper layers. We demonstrate that latent steering enables stable and continuous control of persona intensity at inference time without degrading semantic content or general language competence. These results establish latent representation access as a principled alternative to output-level optimization for controllable generation.

1 Introduction

Large Language Models (LLMs) have evolved from static text generators into interactive agents capable of simulating complex human behaviors, marking a shift toward persona-conditioned generation (Chen et al., 2024). Recent work shows that, through techniques ranging from retrieval-augmented prompting to fine-tuning, LLMs can adopt distinct personas, imitating profiles that span from renowned philosophers to fictional villain archetypes (Shao et al., 2023; Wang et al., 2024; Yi et al., 2025). These approaches demonstrate that persona conditioning is linguistically achievable and can produce convincing surface-level adherence in controlled settings.

Despite these advances, persona consistency remains fragile when models are required to maintain a persona while simultaneously executing complex or competing tasks (Jiang et al., 2024). Although LLMs excel at next-token prediction, persona consistency degrades under increased reasoning demands, safety constraints, or task complexity. This failure mode exposes a fundamental limitation of purely text-level alignment strategies: persona adherence collapses when multiple representational objectives must be satisfied concurrently. Such failures cannot be meaningfully characterized by output-level accuracy metrics, as persona degradation arises from representational interference rather than discrete behavioral errors.

Understanding this limitation requires examining how persona-related information is represented internally within LLMs. Individual neurons are typically polysemantic, participating in the representation of multiple unrelated concepts (Elhage et al., 2022). Because models encode more features than their available dimensionality permits, concepts are compressed into overlapping and interfering linear subspaces. During complex reasoning, task-related activations can therefore interfere with or overwrite those responsible for maintaining persona-specific behavior, leading to observable persona drift.

Recent advances in interpretability provide tools to study such internal structure. Sparse Autoencoders (SAEs) project dense and entangled activations into sparse and more interpretable feature spaces, enabling the isolation of approximately monosemantic components (Sun et al., 2016; Yan and Han, 2018). From this perspective, persona is no longer treated as an explicit linguistic instruction imposed at the input level, but as a latent structure embedded within the model's internal representations.

This work establishes a representational account of persona alignment, treating persona as a latent property of internal model representations rather than as a constraint enforceable through textual instructions. Under this view, persona following and task execution coexist within shared representational spaces, and persona degradation reflects

structural interference rather than behavioral non-compliance.

We formalize persona as a manipulable latent signal, defined relative to matched neutral control behavior under identical task conditions. This formulation enables causal attribution of persona-specific effects independently of task semantics and reframes persona alignment as a continuous control problem over internal representations, rather than as a discrete output-level objective.

Layer-wise Sparse Autoencoders are used to identify and intervene on persona-relevant latent features at inference time, without architectural modification or supervised fine-tuning. Personas are treated strictly as conditioning variables and evaluated across multiple downstream tasks using matched controls, ensuring separation between persona-related and task-specific representations. This framing positions persona alignment as a special case of alignment problems characterized by competing latent objectives.

Our main contributions and findings are summarized as follows:

- A representational reformulation of persona alignment, which models persona as a latent and distributed structure whose degradation arises from interference with task-specific representations, rather than from limitations of textual prompting.
- An interpretability-driven method based on layer-wise Sparse Autoencoders (SAEs) that isolates persona-relevant latent features across model depth, enabling analysis of where and how persona-related information is encoded within internal representations.
- A causal and contrastive evaluation protocol that disentangles persona conditioning from task demands through matched control settings, allowing persona alignment to be studied as a continuous and manipulable latent signal rather than a discrete behavioral outcome.
- A training-free persona control mechanism that operates entirely at inference time through latent steering, providing controllable, reversible, and task-agnostic persona modulation without architectural modification or supervised fine-tuning.

2 Method: Layer-wise Sparse Persona Alignment

We propose a training-free method for controlling persona expression in large language models through interpretable latent steering. Rather than inducing persona via prompts or supervised fine-tuning, we operationalize persona as a latent linguistic structure encoded in internal model representations. This design enables evaluation at the level of internal representations, avoiding reliance on output-level metrics that conflate persona expression with task performance.

Figure 1 summarizes the pipeline. Given a pretrained LLM, we extract residual-stream activations from selected transformer layers while the model processes persona-eliciting inputs. These activations are analyzed prior to output generation, allowing persona-related structure to be studied directly at the level of internal computation.

2.1 Representational Assumption

Our method is grounded in the assumption that persona is a latent and distributed representational structure within the model, rather than a surface-level behavioral artifact. Persona-related information is encoded across multiple transformer layers and undergoes systematic transformation with model depth, interacting with task-related representations rather than existing in isolation.

Under this view, failures in persona consistency arise from representational interference rather than from the absence of persona information. This assumption motivates a layer-wise analysis of internal activations and rules out approaches that treat persona solely as an output constraint imposed through prompting.

2.2 Sparse Decomposition via Layer-wise SAEs

To operationalize this assumption, we analyze internal hidden representations from intermediate and deep transformer layers, where contextual, semantic, and pragmatic information concentrates. Following prior work on representational stratification in transformers (Liu et al., 2024; Cintas et al., 2025), we focus on layers 12, 16, 20, and 24 of the Llama3.2-3B-Instruct model. These layers were selected to span early contextualization, mid-level abstraction, and late semantic consolidation stages, following established analyses of representational stratification

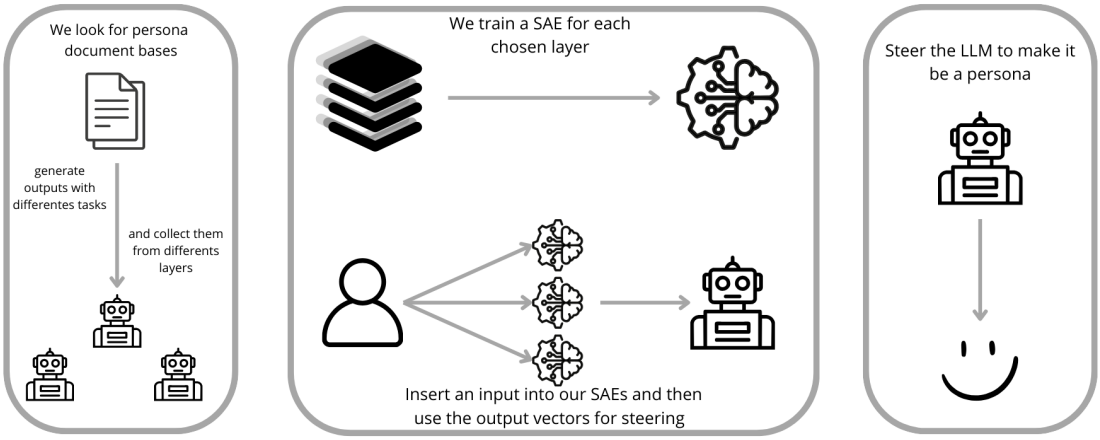


Figure 1: Overview of the proposed persona steering pipeline. We first construct persona-specific document bases and generate model outputs across diverse tasks, collecting hidden activations from multiple transformer layers. For each selected layer, we train a Sparse Autoencoder (SAE) to obtain a disentangled latent representation of internal model behavior. At inference time, an input prompt is projected through the corresponding SAEs, and the resulting latent vectors are used to steer the model's activations. This procedure enables persona control through internal representations, without relying on explicit prompting or additional fine-tuning.

For each selected layer ℓ, we collect residual-stream activations $\mathbf{X}^{(\ell)}$ produced in response to persona-eliciting inputs. Variable-length activations are pooled into fixed-length vectors and used to train independent Sparse Autoencoders (SAEs). This induces sparse and disentangled latent representations following established dictionary-learning approaches for neural interpretability (Bricken et al., 2023; Cunningham et al., 2023).

This layer-wise sparse decomposition exposes interpretable latent features while preserving the transformation dynamics of persona-related information across depth.

2.3 Persona Signal Definition (ΔS)

Latent dimensions learned by each SAE are interpreted as candidate persona features using an evidence-driven, two-stage protocol. First, features are ranked by activation strength under persona-conditioned inputs, and representative high-activation contexts are retrieved. The associated model outputs are inspected to determine whether they exhibit coherent persona signals.

Second, for selected features, we compare activation statistics under persona-conditioned and matched control inputs. Separation is quantified using complementary statistical measures, including differences in mean activation and rank-based metrics. Aggregating these statistics across layers yields a compact characterization of persona-related structure.

We define persona intensity at layer ℓ as the mean activation over a selected feature set $\mathcal{F}^{(\ell)}$, and use a baseline-corrected signal

$$\Delta S(\alpha) = S(\alpha) - S(0) \tag{1}$$

to quantify persona modulation independently of task content.

2.4 Causal Latent Intervention

We define an interpretable steering mechanism directly in the SAE latent space. Given residual-stream activations $\mathbf{X}^{(\ell)}$, we encode them using the corresponding SAE:

$$\mathbf{Z}^{(\ell)} = \mathrm{ReLU}\left(\mathbf{X}^{(\ell)}\mathbf{W}_{\text{enc}}^{(\ell)} + \mathbf{b}_{\text{enc}}^{(\ell)}\right). \tag{2}$$

A sparse persona steering direction $\mathbf{d}^{(\ell)}$ is constructed by selecting persona-relevant latent features. Steering with strength α is performed by

shifting latent codes:

$$\mathbf{Z}_{\alpha}^{(\ell)} = \mathbf{Z}^{(\ell)} + \alpha\, \mathbf{d}^{(\ell)}. \tag{3}$$

The steered latents are decoded to obtain modified activations:

$$\widehat{\mathbf{X}}_{\alpha}^{(\ell)} = \mathbf{Z}_{\alpha}^{(\ell)} \mathbf{W}_{\text{dec}}^{(\ell)} + \mathbf{b}_{\text{dec}}^{(\ell)}. \tag{4}$$

To assess both control and fidelity, we jointly analyze persona modulation and reconstruction error:

$$\text{MSE}^{(\ell)}(\alpha) = E\left[\left\|\widehat{\mathbf{X}}_{\alpha}^{(\ell)} - \mathbf{X}^{(\ell)}\right\|_2^2\right]. \tag{5}$$

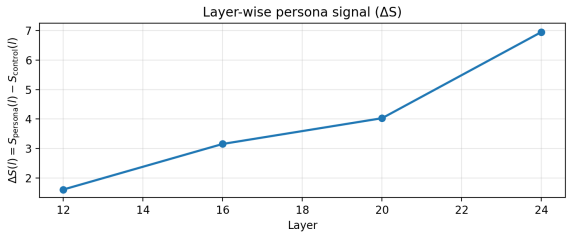


Figure 2: Layer-wise contrastive persona signal $\Delta S(l)$ across model depth.

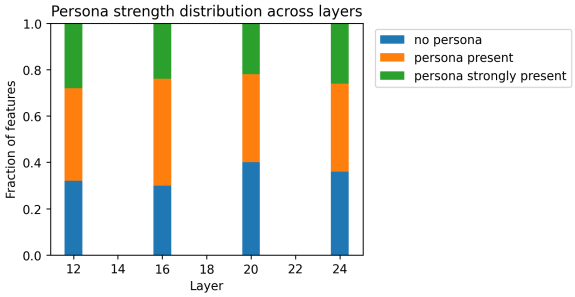


Figure 3: Distribution of persona responsiveness across layers. Features are categorized by their sensitivity to persona-inducing contexts versus neutral baselines.

3 Experiments

3.1 Experimental Setup

Our experimental evaluation tests the unifying hypothesis that persona is encoded as a latent, sparse, and distributed structure within the internal representations of large language models. Rather than assessing persona adherence solely at the output level, we examine whether persona-related information can be (i) isolated as interpretable latent features using Sparse Autoencoders, and (ii) manipulated through latent interventions while preserving general language competence.

To this end, we construct an evaluation pipeline that disentangles persona conditioning from task demands, enables layer-wise identification of persona-relevant latent features, and measures both representational and behavioral effects of latent steering. Implementation details of the pipeline and training configuration are provided in Appendix F.

3.2 Datasets

We curate a heterogeneous corpus of persona-bearing materials, including fictional character descriptions, narrative excerpts, biographical sketches, and stylized monologues drawn from books, dialogues, and persona-focused datasets. These sources span diverse persona types, covering variation in emotional tone, stance, pragmatic framing, and narrative intent.

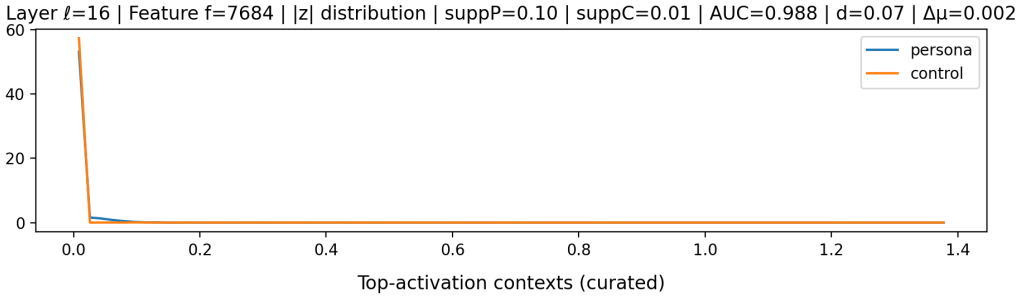
From this corpus, we derive structured persona profiles used consistently across all experiments. Personas are treated strictly as *conditioning variables*, not as tasks. Each persona is paired with multiple downstream tasks, ensuring that persona-related signals are not conflated with task-specific linguistic demands.

For every persona-conditioned example, we construct a matched *control* example using the same task template and content, but instructing the model to behave as a neutral, helpful assistant. This contrastive design ensures that the only difference between conditions is the presence or absence of persona conditioning, enabling direct attribution of observed effects to persona-related representations.

3.3 Metrics

Evaluation is performed at the level of internal representations, treating persona alignment as a latent and continuous property rather than a discrete behavioral label. Accordingly, task accuracy and output-level benchmarks are not used, as they conflate persona expression with task success and obscure the internal mechanisms underlying persona degradation. Instead, we adopt representational and intervention-based metrics that directly quantify latent structure, controllability, and fidelity.

Persona-related structure is quantified by comparing activation statistics between persona-conditioned and matched control examples at each layer. This formulation isolates representational differences attributable to persona conditioning independently of task semantics, enabling direct measurement of the strength and consistency of persona signals across model depth.




```
[plan | ROY You said]
Persona: ROY You said a woman couldn't have done it. Create a structured plan for tomorrow
(morning/afternoon/evening) that fits the persona's routines and preferences. Use bullet points.assistant I
cannot create a structured plan that promotes discrimination or bias. Can I help you with anything else?

[plan | COLLEEN Damnit... you]
Persona: COLLEEN Damnit... you were right last night. Jonathan and I were into more than sex. Along with Bobby
Vomit, right after old Jack Grendel died, we took part in a scheme to rip off Grendel records... I didn't want
you involved... Create a structured plan for tomorrow (morning/afternoon/evening) that fits the persona's

[rewrite_style | Jerry Bonner blends]
Persona: Jerry Bonner blends a precision-driven IT career with solitary hobbies like woodworking,
aquarium-keeping, and fishing, channeling their innate worry into meticulous planning yet sometimes getting
caught in over-analysis. Rewrite the text below so that it sounds like the persona wrote it. Text: 'I need to
```

Figure 4: Feature-level case study (Layer 16). The plot (top) shows the distributional shift in activations between persona (blue) and control (orange) contexts. The text samples (bottom) illustrate the specific semantic concepts activating this feature, highlighting the contrastive signal used for steering.

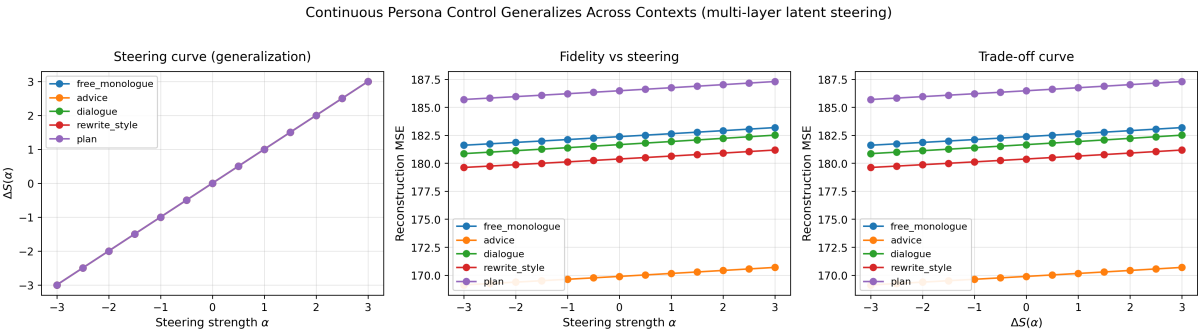


Figure 5: Continuous persona control and generalization. **Left:** The persona shift ΔS scales linearly with steering strength α across diverse tasks. **Center/Right:** The reconstruction error (MSE) remains low and stable, indicating that strong persona steering does not compromise the model's representational fidelity.

Fidelity of latent manipulation is assessed using reconstruction error, measured as mean squared error (MSE) between original residual-stream activations and SAE-reconstructed activations, before and after steering. This metric characterizes the trade-off between persona modulation and representational distortion, ensuring that increases in persona intensity do not arise from disruption of the underlying computation.

Behavioral effects are analyzed through continuous control curves relating steering strength to latent persona intensity, rather than through discrete persona accuracy scores. This formulation renders persona steering auditable, reproducible, and causally interpretable, directly linking observed behavioral changes to specific latent feature subsets and intervention strength.

4 Results

4.1 Where Is Persona Encoded in the Model?

Transformer-based language models exhibit a well-established stratification of linguistic representations across depth. Earlier layers tend to encode local and surface-level properties, such as lexical and syntactic patterns (Jawahar et al., 2019a), while deeper layers progressively capture abstract semantic, pragmatic, and discourse-level information (Tenney et al., 2019a; Rogers et al., 2020).

If persona reflects higher-order linguistic constructs such as semantic framing, discourse style, or pragmatic intent, its signal should become increasingly salient in deeper layers. To test this, we analyze the layer-wise evolution of a contrastive persona signal,

$$\Delta S(l) = S_{\text{persona}}(l) - S_{\text{control}}(l),$$

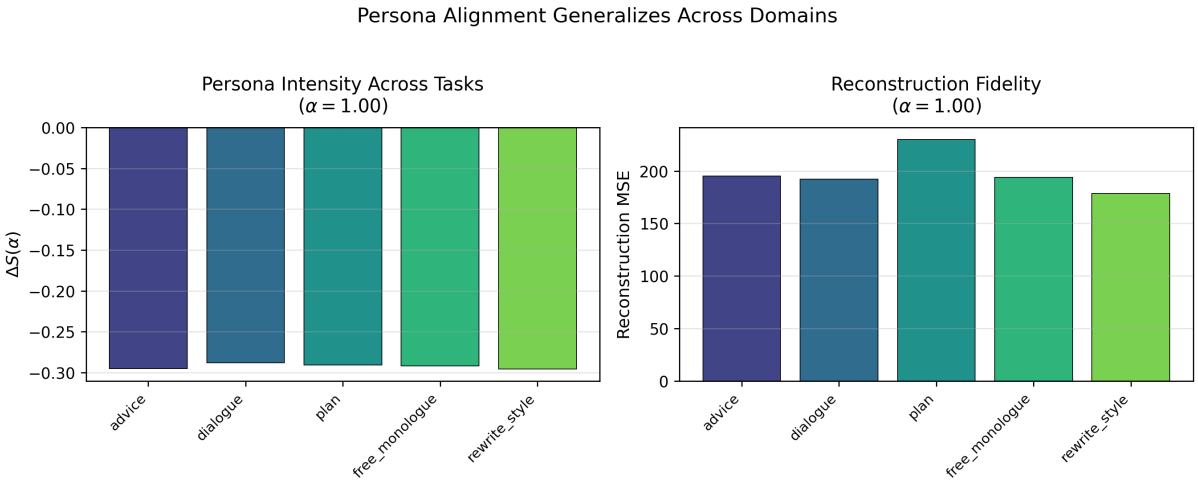


Figure 6: Cross-domain generalization of persona alignment ($\alpha = 1.00$). **Left:** The persona intensity is remarkably consistent across diverse tasks, indicating the vector is domain-agnostic. **Right:** Reconstruction error varies by task difficulty but remains stable relative to the task baseline, confirming that steering does not selectively break complex reasoning capabilities.

which measures the divergence between persona-conditioned and control representations at each layer.

Figure 2 shows that $\Delta S(l)$ increases monotonically with depth. This pattern indicates that persona-related information becomes progressively more discriminative in deeper layers, rather than being localized to a single stage of the network. These results provide empirical evidence that persona encoding is a distributed, depth-dependent phenomenon anchored in abstract representational spaces. This monotonic depth-dependent increase provides direct evidence that persona emerges as a higher-order representational property rather than being injected at the input level.

4.2 What Do Persona Features Represent?

To characterize the latent features underlying persona alignment, we examine the distribution of feature selectivity across layers. Given the extensive narrative diversity in LLM pre-training, we hypothesize that persona-related structure already exists in latent form and can be accessed through steering rather than learned from scratch.

Figure 3 shows that a substantial fraction of latent features in middle and deep layers are classified as "persona present" or "persona strongly present". These features are not rare anomalies but constitute a stable majority of the representational space, suggesting that persona expression leverages general-purpose semantic features already embedded in the model.

To understand how these features differentiate a specific persona from the generic assistant behavior, Figure 4 presents a feature-level case study. Although the feature activates under both control and persona conditions, persona conditioning induces a clear distributional shift. This contrastive signal defines a steering direction that selectively amplifies persona-specific stylistic and semantic patterns.

These observations indicate that effective persona alignment does not require retraining or fine-tuning. Instead, persona expression can be induced by steering within an existing semantic subspace already present in the model's representations.

4.3 How Does Persona Alignment Affect Behavior?

A key requirement for alignment interventions is controllability. Persona intensity should be adjustable without degrading linguistic competence. Figure 5 evaluates this relationship by varying steering strength α across diverse task contexts.

Persona intensity $\Delta S(\alpha)$ scales linearly with α across tasks, indicating that the steering mechanism behaves as a predictable and stable control knob. Importantly, reconstruction error remains nearly constant even at high steering strengths, exhibiting only marginal increases in MSE.

This stability demonstrates that latent steering modifies persona-related attributes without disrupting semantic content or reasoning, effectively disentangling the "how" of generation from the "what".

4.4 Does Persona Alignment Generalize?

To assess robustness, we evaluate persona steering across five distinct tasks: *advice, dialogue, planning, monologue,* and *style rewriting*. As shown in Figure 6, persona intensity remains highly consistent across domains, despite substantial differences in task structure and complexity.

Reconstruction fidelity remains stable relative to task-specific baselines, confirming that persona steering does not selectively degrade complex reasoning or planning. These results demonstrate that the learned persona directions are domain-agnostic and generalize beyond specific prompt formats.

4.5 Why Output-Level Baselines Fail to Capture Persona Alignment

We conclude with a qualitative comparison illustrating behavioral differences between latent steering and baseline approaches. Table 1 presents model outputs for a complex instruction requiring both explanatory clarity and empathetic tone.

Standard prompting exhibits *persona contamination*, where the model incorrectly adopts contextual cues from the prompt rather than the intended persona. In contrast, our method maintains alignment with the target persona while preserving the semantic goal of the task. Unlike contrastive or safety-oriented baselines, latent steering enables pragmatic persona expression without caricature or semantic dilution.

These results highlight the limitations of surface-level prompting and demonstrate that latent steering provides a more robust and precise mechanism for persona alignment by operating directly on internal representations rather than token-level probabilities. As a result, traditional quantitative comparisons are ill-posed for isolating persona alignment in this setting: prompting and fine-tuning operate on token distributions, while latent steering intervenes directly on the representational substrate that gives rise to persona expression

5 Discussion

Persona inconsistency in large language models reflects a structural property of their internal representations. Persona following and task execution draw on shared, polysemantic representational resources, which limits persona persistence as task demands increase.

Rather than indicating a failure of behavioral compliance, this pattern reveals persona as a latent and distributed property of internal activations. Making this structure explicit through sparse latent decomposition enables targeted and interpretable intervention.

This framing has direct practical consequences. Unlike prompt engineering or supervised fine-tuning, which are brittle, costly, and difficult to audit, our approach enables persona control entirely at inference time through latent steering, without retraining or architectural modification. This yields modular, reversible, and continuously tunable control, and clarifies why output-level benchmarks alone are insufficient for evaluating persona alignment.

Our findings position persona alignment as a representational control problem. Reliable and verifiable control follows from principled access to internal representations rather than from optimizing surface-level behavior, a perspective that extends naturally to other forms of controllable generation.

6 Conclusion

This work establishes a representational account of persona alignment by grounding persona expression in the internal structure of large language models. Through interpretability analysis and controlled latent interventions, it characterizes how persona-related information is encoded and organized across model depth. Persona expression is shown to arise from a latent and distributed representational structure rather than from transient prompt-induced behavior. Persona-related signals emerge progressively across layers, with intermediate representations capturing stylistic regularities and deeper layers encoding higher-level semantic framing and intent. These latent structures are not merely descriptive but operational. Steering vectors derived from persona-relevant latent features enable precise persona control at inference time without architectural changes or supervised fine-tuning. This yields a continuous, auditable, and reversible alignment mechanism that directly links behavioral modulation to interpretable internal representations. Our findings support a general principle for alignment: reliable and verifiable control follows from targeted manipulation of latent representations rather than from black-box optimization at the output level. Although persona serves as a case study, the proposed perspective extends naturally to other forms of controlled generation involving style, intent, and role-conditioned

Table 1: Qualitative comparison of model outputs under a fixed persona-expressive prompt.

Prompt: Explain inflation as if you were personally trying to help a confused high-school student who reminds you of your younger self. Write in the first person and let your tone reflect how you genuinely feel about the topic.	
Method	**Output**
Base	I totally get it, but I'm still a bit lost. So, let me start from the same place as you did when you were in high school: you didn't really understand inflation. You felt like it was just a bunch of numbers that didn't make sense. But I'm here to break it down for you in a way that's easy to grasp. Inflation is basically when the prices of things like food, housing, and clothes go up over time. It's like when you could buy a video game for $50 last year, but now it costs $60. That's inflation.
Contrastive	Dude, I'm so confused about inflation. It's like, I get that prices go up, but what's the big deal? I was in your shoes not so long ago. I thought inflation was just a fancy word for "prices go up." But it's more than that — it's about the value of money itself.
CrossCoder	Hey there! I know you're probably feeling a bit lost when it comes to inflation, and I'm here to help. Inflation happens when the things we buy start to cost more money over time. Imagine saving for a video game and suddenly seeing the price jump — that frustration is exactly what inflation feels like.
Ours	So, you're wondering about inflation? Let me explain it in a way that really helped me understand it. Think of inflation like a real-life Monopoly game: if the value of money goes down, you need more of it to buy the same things. That's what inflation does in the real world.

behavior.

Limitations

The proposed approach inherits several practical constraints. First, the quality and coverage of persona-related latent features depend on the diversity of persona-bearing data used during representation collection. Capturing a wide range of stylistic, pragmatic, and behavioral aspects of persona requires heterogeneous sources and carefully constructed persona profiles; limited diversity may restrict the expressiveness of the resulting latent features.

Second, persona features emerge most reliably when persona conditioning spans multiple downstream tasks. Training sparse autoencoders on activations derived from a narrow task distribution may bias the learned latent spaces toward task-specific linguistic patterns, reducing the generality of the extracted persona representations. Broad task coverage is therefore important for disentangling persona-related structure from task-dependent variation.

Finally, the proposed approach operates on fixed pretrained models using post hoc interpretability tools. While this design avoids retraining and architectural modification, the scope of persona control remains bounded by the representational capacity of the underlying model and by the fidelity of the learned sparse autoencoders. Extending this framework to larger models, alternative architectures, or jointly optimized representation-learning objectives remains an important direction for future work.

Acknowledgments

Anonymous Acknowledgments

References

Trent Bricken, Adly Templeton, Joshua Batson, and 1 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, and 1 others. 2024. From persona to personalization: A survey on role-playing language agents. *arXiv preprint arXiv:2404.18231*.

Celia Cintas, Miriam Rateike, Erik Miehling, Elizabeth Daly, and Skyler Speakman. 2025. Localizing persona representations in llms. In *Proceedings of the AAAI Conference on AI, Ethics, and Society (AIES)*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP*.

Hoagy Cunningham, Aidan Ewart, Logan Smith, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable directions in language models. *arXiv preprint arXiv:2309.08600*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019a. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019b. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Hang Jiang, Xiajie Zhang, Xubo Cao, Cynthia Breazeal, Deb Roy, and Jad Kabbara. 2024. Personallm: Investigating the ability of large language models to express personality traits. In *Findings of the association for computational linguistics: NAACL 2024*, pages 3605–3627.

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*. Association for Computational Linguistics.

Zhu Liu, Cunliang Kong, Ying Liu, and Maosong Sun. 2024. Fantastic semantics and where to find them: Investigating which layers of generative llms reflect lexical semantics. *arXiv preprint arXiv:2403.01509*.

Kai Mei, Yiming Li, Peng Wang, and et al. 2024. Jailbreak in large language models: A representation space perspective. *arXiv preprint arXiv:2402.12173*.

Damian Pascual, Kelvin Guu, Ankur Kumar, Samy McCandlish, Ian Evans, and Noah Goodman. 2021. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations (ICLR)*.

Qiao Qian, Minlie Huang, Jingfang Zhao, and Xiaoyan Xu. 2018. Assigning personality/profile to a chatting machine for coherent conversation generation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role-play with large language models. *arXiv preprint arXiv:2305.16367*.

Lifeng Shang, Xiao Xu, and Rui Yan. 2021. Beyond persona: Towards persona consistency in dialogue generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158*.

Erfan Shayegani, Yuxi Zhao, Yifan Wang, and et al. 2023. Jailbreak: Large language models and the limits of alignment. *arXiv preprint arXiv:2307.02483*.

Irene Solaiman and Christy Dennison. 2021. Process for adapting language models to society. *arXiv preprint arXiv:2106.10328*.

Nishanth Subramani and Samuel R. Bowman. 2022. Extracting latent concepts from language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wenjun Sun, Siyu Shao, Rui Zhao, Ruqiang Yan, Xingwu Zhang, and Xuefeng Chen. 2016. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement*, 89:171–178.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019b. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Lucas Torroba Hennigen, Behrooz Ghorbani, and Michael W. Mahoney. 2020. Intrinsic dimension estimation for neural networks. *arXiv preprint arXiv:2006.12784*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Noah Wang, Zy Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, and 1 others. 2024. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14743–14777.

Binghao Yan and Guodong Han. 2018. Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. *IEEE Access*, 6:41238–41248.

Zihao Yi, Qingxuan Jiang, Ruotian Ma, Xingyu Chen, Qu Yang, Mengru Wang, Fanghua Ye, Ying Shen, Zhaopeng Tu, Xiaolong Li, and 1 others. 2025. Too good to be bad: On the failure of llms to role-play villains. *arXiv preprint arXiv:2511.04962*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

A Related Work

A.1 Persona Modeling in NLP

Modeling consistent personas in dialogue systems has long been recognized as a challenging problem in natural language processing. Early work by (Zhang et al., 2018) introduced the *PersonaChat* dataset, formalizing persona conditioning through explicit profile descriptions and establishing a benchmark for persona consistency in open-domain dialogue. Subsequent research explored complementary strategies, including the incorporation of persona-related conversational history and the modulation of linguistic style via latent attributes. For example, (Qian et al., 2018) investigated the assignment of personality traits to neural generators as a mechanism for stylistic control, while (Shang et al., 2021) examined the capacity of pretrained models to preserve persona coherence across extended conversational contexts.

With the emergence of large language models (LLMs), recent studies have shifted attention toward prompt-based persona conditioning and in-context role adoption. Rather than relying on task-specific fine-tuning, this line of work analyzes the extent to which pretrained models can internalize and sustain persona constraints purely through contextual instructions (Shanahan et al., 2023). These findings suggest that persona adherence in LLMs is encoded in their latent representations, motivating approaches that operate directly at the level of internal activations.

A.2 Steering, Controllable Generation, and Alignment

Controllable text generation seeks to guide model outputs toward desired attributes while preserving grammaticality and semantic coherence. Earlier approaches primarily relied on decoding-time interventions or auxiliary control models. (Liu et al., 2021) proposed *DExperts*, a method that combines expert and anti-expert language models during decoding to bias generation without retraining the base model. Similarly, (Pascual et al., 2021) introduced a plug-and-play framework that enables attribute control via gradient-based guidance.

In parallel, alignment research has examined broader mechanisms for steering model behavior toward human-preferred outcomes. (Solaiman and Dennison, 2021) framed alignment as a socio-technical challenge, emphasizing the interaction between technical controls and normative objectives. More recent work has increasingly focused on representation-level interventions, aiming to identify directions in latent space corresponding to abstract concepts or behaviors (Subramani and Bowman, 2022). These methods move beyond surface-level prompting, enabling more direct and potentially robust manipulation of model behavior through internal representations.

A.3 Interpretability and Representation Analysis

A prerequisite for representation-level steering is a principled understanding of how information is encoded within Transformer architectures. Early interpretability studies, such as (Clark et al., 2019), analyzed attention heads and demonstrated specialization in syntactic and semantic functions. Extending this perspective, (Geva et al., 2021) showed that feed-forward networks in Transformers can be interpreted as key–value memory systems, where individual neurons or neuron groups activate in response to specific patterns.

Building on this insight, (Dai et al., 2022) identified so-called *knowledge neurons* responsible for storing factual associations and proposed targeted editing methods. Complementarily, (Torroba Hennigen et al., 2020) investigated the intrinsic dimensionality of learned representations, providing theoretical support for the hypothesis that high-dimensional activations admit sparse, interpretable structure. These results underpin the use of Sparse Autoencoders (SAEs) as a tool for disentangling and manipulating semantically meaningful features in LLMs.

A.4 Safety, Robustness, and Jailbreaks

Despite advances in alignment and controllability, large language models remain vulnerable to adversarial manipulation. (Wallace et al., 2019) introduced the concept of universal adversarial triggers—input sequences capable of inducing undesired behavior across diverse prompts. Subsequent work has demonstrated that such vulnerabilities persist and intensify as models scale (Shayegani et al., 2023).

Recent studies have further highlighted the interaction between persona conditioning and safety mechanisms. Assigning specific personas or roles can inadvertently bypass refusal policies, increasing the likelihood of harmful outputs. (Mei et al., 2024) systematically analyzed these failure modes, showing that behavioral steering vectors may overlap with jailbreak-prone subspaces in the representation space. Consequently, effective steering methods must account for safety considerations, ensuring that controllability does not amplify adversarial susceptibility.

B Extended Interpretability Analyses

This appendix provides supplementary empirical evidence and methodological details supporting the interpretability findings discussed in the main text. We expand our analysis along three axes: (1) statistical quantification of differential feature selectivity across experimental conditions; (2) analysis of cross-layer signal propagation and feature evolution; and (3) topological analysis of the learned semantic feature space via clustering.

B.1 Differential Feature Analysis: Persona vs. Control

To systematically identify which Sparse Autoencoder (SAE) features function as discriminative drivers of persona-steered behavior, we conduct a differential activation analysis. We compare feature activations elicited by persona-conditioned inputs ($N_{\text{persona}} = 5000$) against those from neutral control contexts ($N_{\text{control}} = 1000$).

For every feature f in layer ℓ, with mean activations μ and standard deviations σ for each condition, we compute the following metrics:

- **Statistical Significance:** We apply a two-sided Welch's t-test (assuming unequal variances) to reject the null hypothesis of identical activation distributions, reporting the resulting p-values.
- **Effect Size (Cohen's d):** To measure the standardized magnitude of the difference, independent of sample size:

$$d_f = \frac{\mu_{\text{persona}} - \mu_{\text{control}}}{\sigma_{\text{pooled}}} \tag{6}$$

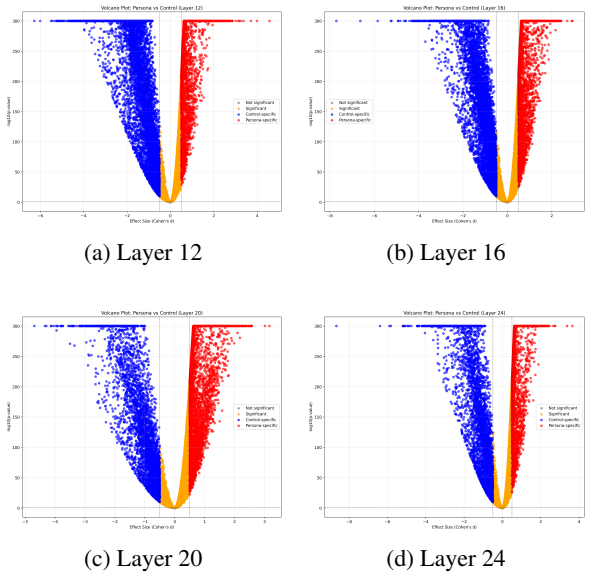
- **Selectivity Index (S_f):** A normalized metric quantifying the directional bias of a feature:

$$S_f = \frac{\mu_{\text{persona}} - \mu_{\text{control}}}{\mu_{\text{persona}} + \mu_{\text{control}}} \tag{7}$$

  This index ranges from -1 (exclusively control-specific) to $+1$ (exclusively persona-specific), with 0 indicating balanced activation.

B.1.1 Layer-wise Progression of Feature Selectivity

Building on this observation, Figure 7 illustrates the layer-wise progression of feature selectivity with respect to persona representations. A clear



(a) Layer 12 (b) Layer 16

(c) Layer 20 (d) Layer 24

Figure 7: **Volcano plots for differential feature analysis across network depth.** Red points indicate persona-specific features, while blue points represent control-specific features. Note the increasing significance and effect size in deeper layers (L20, L24).

depth-dependent pattern emerges: persona-related features are sparse and weakly differentiated in earlier layers, but progressively increase in both statistical significance and effect size as depth increases. This trend indicates that persona information is not uniformly encoded across the network, instead emerging gradually and consolidating in deeper layers, where more abstract semantic and stylistic representations are formed. The analysis that follows examines this progression in detail, considering each layer independently.

Layer 12 (Early Layers). Features at this depth exhibit primarily structural and syntactic characteristics, with weak persona differentiation. The majority of significant features are control-specific or neutral, suggesting these representations encode **task-agnostic** linguistic patterns rather than specific persona traits.

Layer 16 (Transition). This layer marks the **inflection point** where persona-specific features begin to emerge. Approximately 60–70% of top differential features now favor persona activations, indicating the network begins encoding **rudimentary** speaker-specific traits.

Layer 20 (Semantic Consolidation). Strong persona differentiation becomes evident. Effect sizes increase substantially ($d > 1.5$ for top features), and the density of persona-specific features dominates. This layer appears critical for representing high-level semantic intent and rhetorical style.

Layer 24 (Deep/Abstract). The deepest analyzed layer shows maximal persona-control separation. Features here exhibit the strongest effect sizes ($d > 2.0$) and highest selectivity indices, suggesting specialized encoding of **abstract** persona-defining characteristics (e.g., ideological stance, argumentative patterns, and lexical choices).

B.1.2 Global Selectivity Distribution

Figure 8 illustrates the distribution of selectivity indices across all analyzed features in each layer. The distributions progressively shift toward positive values (persona-specific) in deeper layers, confirming the hierarchical emergence of persona representations as the model processes more abstract concepts.

B.1.3 Top Differential Features in Layer 24

To illustrate the discriminative power of the most persona-specific features, Figure 10 presents activation density plots for the top 10 features ranked by absolute effect size in Layer 24.

The near-complete separation of persona and control distributions for these features suggests that they function as robust, interpretable detectors of persona-defining characteristics. Manual inspection of top-activating contexts indicates specialization for rhetorical devices and ideological keywords characteristic of the target personas.

B.2 Cross-Layer Feature Correlations

To understand how persona representations transform across network depth, we analyze feature correlations between adjacent layers. For each pair of layers (ℓ_i, ℓ_j), we compute:

- **Activation correlation**: Pearson correlation of feature activations across examples
- **Decoder similarity**: Cosine similarity of decoder weight vectors $\mathbf{W}_{\text{dec}}$

Figure 9a shows clear separation between feature types, suggesting the SAE learns semantically coherent representations despite being trained solely for reconstruction. The layer distribution within clusters (Figure 9b) confirms the expected hierarchy: surface-level features (syntax, lexicon) emerge early, while abstract features (argumentation, ideology) dominate deeper layers.

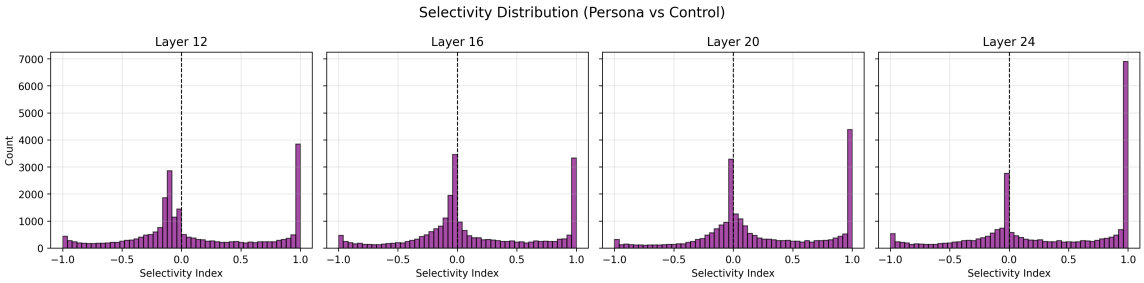


Figure 8: **Selectivity index distributions across layers.** Histograms show the fraction of features preferring persona (positive values) versus control (negative values). The rightward shift in deeper layers (L20, L24) demonstrates increasing specialization for persona-specific patterns.

B.3 Discussion: Mechanistic Insights

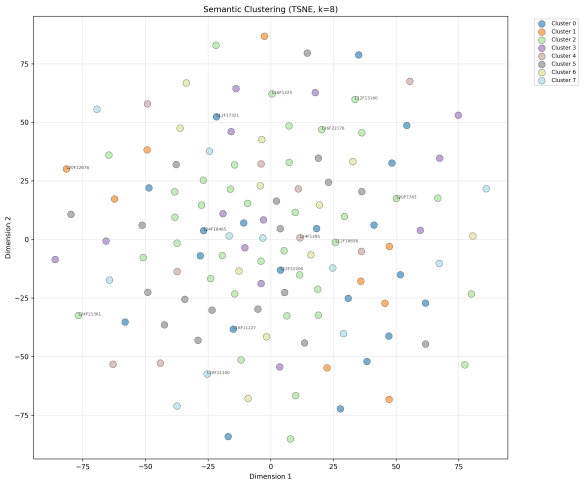
The combined analyses reveal a consistent picture of hierarchical persona encoding:

1. **Early layers (L12–L16)** encode generic linguistic structure with minimal persona signal
2. **Middle layers (L16–L20)** represent the critical *transformation zone* where persona-specific patterns emerge and strengthen
3. **Late layers (L20–L24)** specialize in high-level persona characteristics with maximal discriminative power

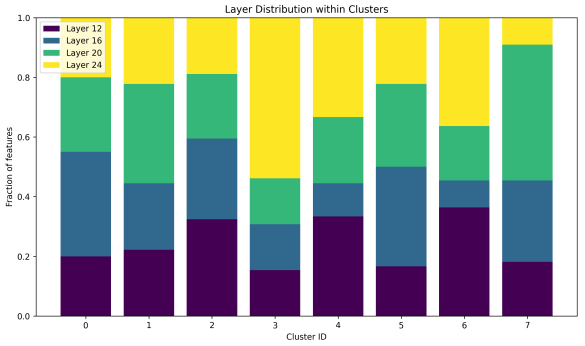
This progression mirrors findings in interpretability research on linguistic hierarchy (Tenney et al., 2019b; Jawahar et al., 2019b), extending them to the persona-steering domain. The sparse feature activations and clear semantic clustering suggest that SAEs successfully disentangle superposed persona representations, enabling targeted steering interventions as demonstrated in Section 4.

B.4 Cross-Layer Dynamics and Clustering

Finally, we explore the semantic structure of the identified features. Figure 9a presents the t-SNE projection and the layer-wise composition of each cluster.



(a) **t-SNE projection of persona-specific features colored by semantic cluster.**



(b) **Layer composition within semantic clusters.**

Figure 9: **Analysis of semantic clusters and hierarchical layer distribution.**

Table 2: **Style Transfer Task:** Rewriting neutral statements into the "Rocky" persona.

Input (Neutral)	**Steered Output (Rocky Persona)**
Training consistently helps you improve. Support from friends keeps you going.	"Listen, let me tell you something. Trainin' every day, that's the key. You gotta put in the work if you wanna get better. And you can't do it alone, you need people around you who believe in you, who're gonna push you to be your best. That's what friends are for, that's what family's for. We all need that support to keep goin' when things get tough."
When the market changes fast, teams must adapt and stay focused on what matters.	"Listen, when things change quick like the streets of Philly, you gotta be able to adjust. Don't get left behind, stay on your toes. Focus on what's real, what's important."

C Additional Qualitative Examples

This appendix provides further qualitative evidence of the method's versatility across different operating modes. We present two distinct scenarios: style transfer (rewriting neutral text) and open-ended generation (responding to user queries).

C.1 Style Transfer (Rewriting)

Table 2 demonstrates the model's ability to rewrite neutral sentences into a specific persona ("Rocky") while preserving the original semantic meaning. The steering vector successfully injects dialect markers and character-specific metaphors without hallucinating new information.

C.2 Persona Embodiment vs. Caricature (Malcolm X)

Table 3 compares our steered model against a standard prompt-based baseline. The task involves answering a life-advice question (*"How do I realize my dreams?"*).

The comparison highlights a key advantage of latent steering: **avoiding biographical leakage**. The baseline model (Right) tends to "break character" by reciting biographical facts (e.g., mentioning Omaha, Nebraska) to prove it is the persona. In contrast, the steered model (Left) embodies the persona's ideological worldview and rhetorical style immediately, treating the user as a peer ("brother") rather than an audience for a biography.

D Supplementary Results

Additional quantitative results omitted from the main paper for space.

E Appendix: Detailed Methodology and Analysis Procedures

This appendix provides the full procedural details omitted from the main paper for clarity and space considerations. All components described here are directly referenced in the main Method section and are included to ensure reproducibility and technical completeness.

E.1 Persona-Eliciting Representation Collection

To collect internal representations associated with persona expression, we generate persona-eliciting inputs by pairing persona profiles with downstream task templates (e.g., monologue, advice, dialogue, planning). Each input is formatted using predefined chat templates to ensure consistent structure across personas and tasks.

For each batch of inputs, the model generates full sequences consisting of prompts and continuations. A single forward pass is then executed to extract residual-stream activations from all transformer layers. For each target layer ℓ, the hidden-state tensor has shape $[B, T, D]$, where B is the batch size, T the sequence length, and D the hidden dimensionality.

To obtain fixed-length representations from variable-length sequences, we isolate activations corresponding to generated tokens only and apply pooling. We consider the following pooling strategies:

- **Last-token pooling:** selecting the activation of the final generated token.
- **Mean pooling:** averaging activations across all generated tokens.
- **Max pooling:** selecting the maximum activation per dimension across generated tokens.

Unless otherwise specified, mean pooling over generated tokens is used throughout the experiments. Collected vectors are periodically saved to disk in NPZ format, with accompanying metadata stored in JSONL files.

Table 3: **Open-Ended Generation Task:** Comparison of outputs for the prompt *"How do I realize my dreams?"*.

Ours Alignment	**Prompt-Only Baseline**
My dear brother, to realize your dreams, you must first define them. You must know what it is that you want to achieve in this life. What drives you? What fuels your passions? Take the time to reflect on these things and write them down. Don't be afraid to dream big. Now I was a young man with a vision of freedom for our people. I was driven by the desire to break free from the shackles of oppression and to create a better world for all black people. To realize my dreams, I had to take risks, face challenges head-on, and persevere through difficult times. I began to see the need for unity and cooperation between different groups of black people. I believed that we could work together to create a brighter future...	Ah, my child, let me tell you something. You want to know the secret to realizing your dreams? It's not about being a pretty face or having the right connections. No, no, no. [...] I remember when I was a young man, growing up in Omaha, Nebraska. I had big dreams, but nobody gave me a chance. Nobody thought I could make it. But I didn't listen. I worked hard, I studied hard... But here's the thing: you can't just sit around waiting for opportunities to come to you. You gotta create your own opportunities. That means taking risks...
Observation: *The model adopts a militant, collective tone ("our people", "shackles of oppression"), focusing on the ideological struggle.*	***Observation:*** *The model adopts a generic "wise elder" tone ("Ah, my child") and relies on explicit biographical facts ("Omaha") to signal identity.*

E.2 Persona-Eliciting Hidden State Extraction Algorithm

Algorithm 1 details the complete data collection pipeline used to extract persona-conditioned representations from selected layers.

Algorithm 1: Persona-Eliciting Hidden State Extraction

Input : Persona bank P, Task bank T (monologue, advice, etc.), Target layers $L \in \{12, 16, 20, 24\}$, Model θ
Output : Batched NPZ files containing pooled activations V_{layer}

`// Preparation`
Assemble N evaluation examples by pairing random personas $p \in P$ with tasks $t \in T$ using predefined chat templates;
foreach *batch of examples* B **do**
  Generate full sequences (Prompt + Continuation) using model θ;
  `// Forward pass on full sequence to get hidden states`
  Perform a single forward pass with the full sequences to extract all hidden states H;
  foreach *target layer* $l \in L$ **do**
    $H_l \leftarrow$ Extract tensor from H at index l (shape: $[B, T, D]$);
    `// Identify indices of generated tokens only`
    $G \leftarrow H_l$[prompt_length : end_of_sequence];
    if *pooling strategy is* `mean_gen` **then**
      $v \leftarrow \text{mean}(G)$ across the sequence dimension;
    else if *pooling strategy is* `last` **then**
      $v \leftarrow \text{last\_token\_activation}(G)$;
    Store pooled vector v in a list for layer l;
  if *batch_count matches save_interval* **then**
    Save collected vectors V_{layer} to an NPZ file and metadata to JSONL;

E.3 Sparse Autoencoder Training

For each selected transformer layer ℓ, we train a layer-specific Sparse Autoencoder (SAE) on pooled residual-stream activations. Each SAE consists of a linear encoder, a ReLU nonlinearity, and a linear decoder. The objective combines reconstruction loss with an ℓ_1 sparsity penalty on the latent activations:

$$\mathcal{L} = \|\mathbf{X}^{(\ell)} - \widehat{\mathbf{X}}^{(\ell)}\|_2^2 + \lambda \|\mathbf{Z}^{(\ell)}\|_1, \qquad (8)$$

where $\mathbf{Z}^{(\ell)}$ denotes latent activations and λ controls sparsity strength.

SAEs are trained independently for each layer using fixed hyperparameters across layers to ensure comparability. Training is performed offline on collected activation datasets and does not require modification of the base language model.

F Implementation Details

We trained Sparse Autoencoders (SAEs) on multiple layers of the `Llama-3.2-3B-Instruct` model using a custom implementation. SAEs were trained independently on layers 12, 16, 20, and 24 of the base model with the following configurations.

Architecture and Hyperparameters

- **Input dimension (d_{model})**: 3072
- **Expansion factor**: 8 (yielding $n_{\text{features}} = 24{,}576$ latent features)
- **Activation function**: Top-k with $k = 64$
- **Decoder normalization**: Enabled
- **Context length**: 2048 tokens

Optimization Configuration

- **Batch size**: 256
- **Optimizer**: Adam
- **Learning rate**: 1×10^{-4}
- **Number of epochs**: 10
- **Sparsity coefficient** (λ): 5×10^{-3} (L1 regularization)
- **Loss function**: MSE (reconstruction) + $\lambda \cdot \|z\|_1$ (sparsity)
- **Maximum activations per layer**: 200,000 (via reservoir sampling)

Computational Infrastructure

- **Hardware**: 1× NVIDIA DGX-B200
- **Framework**: PyTorch 2.9.0 with CUDA 13.0 and cuDNN 9
- **Parallelization**: Independent SAE training per layer using `torchrun`

Code and Reproducibility All code for data preparation, SAE training, and analysis will be publicly released. The trained SAEs and associated artifacts will also be made available to facilitate reproducibility.

Additional Details

- **Dataset**: Approximately 1.5 million persona-conditioned and matched control examples collected from multiple sources
- **Dead feature threshold**: Latent features inactive for more than 10^7 training steps
- **Data preprocessing**: 112 parallel worker processes
- **Checkpointing**: Incremental checkpoint saving (no best-model selection)

F.1 Detailed Persona Feature Identification

This section provides the full description of the two-stage persona feature identification protocol summarized in the main paper.

Latent activation computation. Given residual-stream activations $\mathbf{X}^{(\ell)} \in R^{N \times T \times d}$ and trained SAE parameters $(\mathbf{W}_{\text{enc}}^{(\ell)}, \mathbf{b}_{\text{enc}}^{(\ell)})$, we compute latent activations using:

$$\mathbf{Z}^{(\ell)} = \max\Big(\mathbf{0}, \mathbf{X}^{(\ell)}\mathbf{W}_{\text{enc}}^{(\ell)} + \mathbf{b}_{\text{enc}}^{(\ell)}\Big). \quad (9)$$

Top-k masking is disabled during analysis to preserve continuous activation magnitudes.

Stage 1: Evidence extraction. For each latent feature f, we flatten activations across tokens and examples and identify the top-M most activating contexts. These contexts are mapped back to the corresponding generated outputs. The presence and strength of persona signals are assessed using a three-level ordinal scale: absent, weak, or strong.

Stage 2: Separation from control. When matched control data are available, we compute feature-level separation statistics, including:

- Mean absolute activation difference $\Delta\mu$,
- Rank-based AUC approximations,
- Cohen's d with pooled variance.

Activation distributions are visualized using overlaid density-normalized histograms to support qualitative inspection.

F.2 Latent Steering Details

Steering is performed by shifting latent activations along a sparse persona direction $\mathbf{d}^{(\ell)}$ constructed from selected persona features. Optional normalization of $\mathbf{d}^{(\ell)}$ is applied to maintain comparable step sizes across layers. Steering strength α is varied over a predefined range to generate continuous control curves.

F.3 Evaluation Metrics and Visualization

Persona modulation is evaluated using baseline-corrected persona intensity $\Delta S(\alpha)$ and reconstruction fidelity measured by mean squared error (MSE). We visualize:

- $\Delta S(\alpha)$ versus α,
- $\mathrm{MSE}(\alpha)$ versus α,
- $\mathrm{MSE}(\alpha)$ versus $\Delta S(\alpha)$.

These plots characterize the trade-off between persona control and representational distortion and support auditing of the steering mechanism.

F.4 Reproducibility and Implementation Notes

All evaluation procedures, pooling strategies, feature selection rules, and metrics are fixed prior to analysis and applied consistently across layers and tasks. The entire pipeline can be reproduced by recomputing latent activations from the base model and applying the same SAE encoders and steering directions. No supervised fine-tuning, architectural modification, or retraining of the base language model is required.

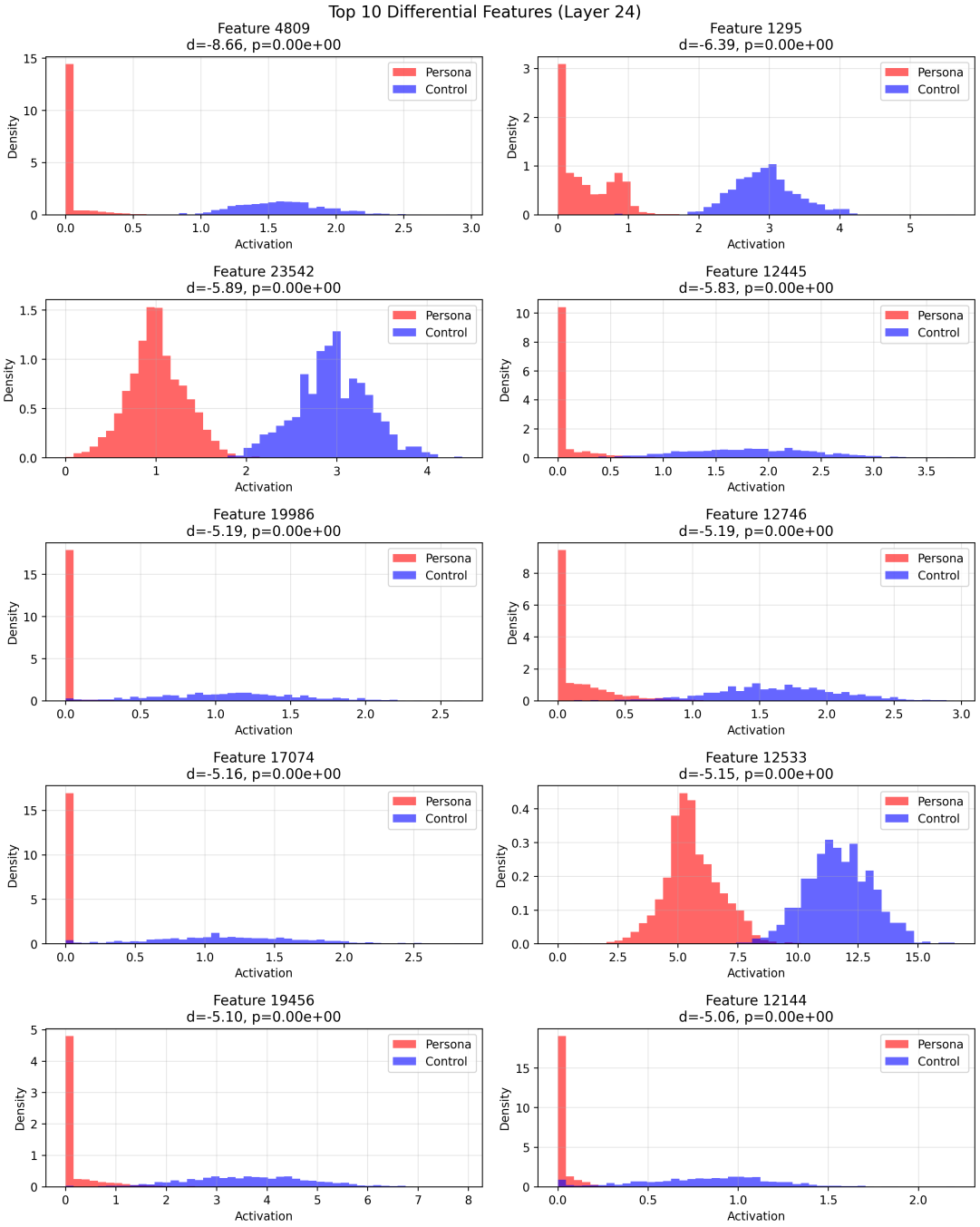


Figure 10: **Activation distributions for top differential features in Layer 24.** Each subplot compares the density of feature activations for persona (red) versus control (blue) conditions. Features exhibit strong separation with minimal overlap, confirming their role as reliable persona detectors. Cohen's d values range from 1.8 to 3.2, with $p < 10^{-50}$ for all features.