Beyond the Leaderboard: Model Diffing for Understanding Performance Disparities in LLMs

Sabri Boughorbel **Fahim Dalvi** **Nadir Durrani** **Majd Hawasly**

Qatar Computing Research Institute, HBKU, Doha, Qatar

{sboughorbel,faimaduddin,ndurrani,mhawasly}@hbku.edu.qa

Abstract

As fine-tuning becomes the dominant paradigm for improving large language models (LLMs), understanding what changes during this process is increasingly important. Traditional benchmarking often fails to explain *why* one model outperforms another. In this work, we use **model diffing**, a mechanistic interpretability approach, to analyze the specific capability differences between **Gemma-2-9b-it** and a **SimPO-enhanced** variant. Using **crosscoders**, we identify and categorize latent representations that differentiate the two models. We find that SimPO acquired latent concepts predominantly enhance safety mechanisms (+32.8%), multilingual capabilities (+43.8%), and instruction-following (+151.7%), while its additional training also reduces emphasis on model self-reference (-44.1%) and hallucination management (-68.5%). Our analysis shows that model diffing can yield fine-grained insights beyond leaderboard metrics, attributing performance gaps to concrete mechanistic capabilities. This approach offers a transparent and targeted framework for comparing LLMs.

1 Introduction

Open-weight LLMs have transformed the AI landscape, making it increasingly challenging for academic and low-resource organizations to train competitive models from scratch (Yang et al., 2025; Team et al., 2025; Fanar-Team et al., 2025; Liu et al., 2024; Grattafiori et al., 2024). Instead, fine-tuning models has become the mainstream approach for developing new capabilities and improving performance. Understanding precisely *what* changes during fine-tuning and *why* certain models outperform others remains challenging.

Current evaluation paradigms rely heavily on benchmarks, which, while useful for capturing specific aspects of model performance, come with significant limitations. As benchmarks gain popularity, the risk of data contamination increases (Xu et al., 2024), and over time, they can become saturated, making them costly to update or replace. Additionally, benchmarks are susceptible to gaming (Verge, 2025), which undermines their reliability. Human evaluations such as LMArena (Chiang et al., 2024) offer more authentic assessments, but they are resource-intensive and can still be swayed by superficial factors like response style and verbosity rather than true differences in model capability (Li et al., 2024; Singh et al., 2025).

For example, the *Simplified Preference Optimization (SimPO)* technique (Meng et al., 2024) has been promoted as a significant advancement in RLHF, credited with boosting the performance of `Gemma-2-9b-it` (Team et al., 2024) across both benchmark scores and human preference evaluations. However, a closer look reveals that these improvements may be largely attributable to superficial factors such as stylistic polishing and output formatting, rather than genuine gains in reasoning, factual accuracy, or task competence. This raises a critical question: *Are fine-tuning methods like SimPO truly enhancing model capabilities, or merely optimizing for appearances that game existing evaluation setups?*

In this paper, we use Model Diffing (Lindsey et al., 2024; Minder et al., 2025) with crosscoders to analyze the latent representations of two models: `Gemma-2-9b-it` and its fine-tuned variant `Gemma-2-9b-it-SimPO`. By comparing both to each other, and to their shared base model (`Gemma-2-9b-pt`), we identify and categorize representation-level changes that help explain observed performance differences. This mechanistic approach offers a nuanced view of how SimPO fine-tuning alters model behavior, revealing gains and potential regressions in capabilities.

Our analysis shows that SimPO fine-tuning leads to targeted shifts in model capabilities rather than uniform improvements. We find substantial increases in **safety and moderation (+32.8%)**, **mul-**

tilingual and stylistic processing (**+43.8%**), and **instruction-following** (**+151.7%**), aligning with SimPO's optimization for alignment and human preference signals. At the same time, we observe notable regressions in **hallucination detection** (**–68.5%**), **model self-reference** (**–44.1%**), and **structured output generation** (**–37.1%**), suggesting a trade-off between confident, polished outputs and internal verification or reasoning. These changes point to a broader shift: SimPO appears to prioritize *fluency and alignment cues* over deliberation or factual introspection, which may partially explain its improved preference ratings despite mixed technical performance. Crucially, these shifts are only visible through model diffing, not from benchmark scores or leaderboard deltas, highlighting the need for deeper mechanistic diagnostics in evaluating LLM enhancements.

2 Methodology

To analyze model differences, we employ the recently developed technique of **Model Diffing** using **crosscoders** (Lindsey et al., 2024; Minder et al., 2025). Crosscoders are a specialized form of sparse autoencoders (Yun et al., 2021; Bricken et al., 2023; Huben et al., 2023) that learn a shared dictionary of interpretable latent concepts across two models. This enables us to identify how internal representations shift or diverge after fine-tuning.

2.1 Model Diffing with Crosscoders

The crosscoder workflow involves three main steps:(1) A shared dictionary is trained to reconstruct the activation patterns from both models. (2) For each latent dimension, a pair of decoder directions is learned, one for each model. (3) The differences between these directions are analyzed to identify model-specific capabilities.

By comparing the norm differences between corresponding latent vectors in each model, we can identify concepts that are uniquely important to one model relative to the other. The norm difference between two models M_1 and M_2 is defined as:

$$\Delta_{\text{norm}}(j) = \frac{1}{2}\left(\frac{\|\mathbf{d}_j^{\text{M}_2}\|_2 - \|\mathbf{d}_j^{\text{M}_1}\|_2}{\max(\|\mathbf{d}_j^{\text{M}_2}\|_2, \|\mathbf{d}_j^{\text{M}_1}\|_2)} + 1\right)$$

where $\mathbf{d}_j^{M_1}$ and $\mathbf{d}_j^{M_2}$ are the decoder vectors corresponding to latent j in the two models.

This approach may suffer from two known failure modes: *Complete Shrinkage* and *Latent Decoupling*, which can cause shared latents to be misclassified as model-specific. To mitigate this, we apply the *Latent Scaling* technique (Minder et al., 2025; Wright and Sharkey, 2024), which estimates two coefficients, ν^{ϵ} and ν^{r}, to more accurately measure latent presence across models. Combined with *BatchTopK* training (Bussmann et al.; Gao et al., 2025), this enables identification of latents that are causally unique to the fine-tuned or base model.

2.2 Experimental Setup

We trained crosscoders to study activation patterns across three variants of the Gemma-2-9b model. Specifically, we employed the BatchTopK Sparse Autoencoder (SAE) training method with a latent dimensionality of 114,688, top-$k = 100$ and learning rate of 1e-4. BatchTopK has been shown to outperform the traditional L_1-based crosscoder training loss (Bussmann et al.; Gao et al., 2025). Following prior work (Lieberum et al., 2024), we selected layer 20 for analysis. Crosscoders were trained using 200M tokens from a mixed corpus comprising the FineWeb and LMSys datasets (Penedo et al., 2024; Zheng et al., 2023). We considered the following model variants: (1) **Gemma-2-9b-pt**: The pretrained model (2) **Gemma-2-9b-it:**[1] The instruction-tuned model (supervised FT and aligned version) (3) **Gemma-2-9b-it-SimPO:**[2] The SimPO-enhanced variant of the instruction-tuned model.

2.3 Design Iteration

The intuitive starting point for our analysis was to compare `Gemma-2-9b-it` (instruction-tuned) with `Gemma-2-9b-it-SimPO` (SimPO-enhanced), since our primary interest was in understanding what makes SimPO effective. However, this direct comparison showed nonsignificant difference. The distribution of the latent normal difference showed concepts falling into a generic "other" category as illustrated in Figure 1a with a norm difference mostly in the range of 0.3 to 0.6, which were too subtle to yield meaningful interpretability.

This outcome led us to reassess our approach. A likely explanation is that once a model has undergone instruction tuning, further improvements like SimPO operate within a narrow behavioral subspace. They may modify surface-level generation preferences or alignment signals rather than introducing fundamentally new internal representations. As a result, SimPO's changes are less

[1] https://hf.co/google/gemma-2-9b-it
[2] https://hf.co/princeton-nlp/gemma-2-9b-it-SimPO

Category	IT	SimPO	Diff
Math	1190	1201	11
Coding	1237	1248	11
Instruction Following	1220	1234	14
Russian	1242	1258	16
Overall	1241	1260	19
Hard Prompts	1217	1240	23
English	1252	1275	23
Multi-Turn	1216	1240	24
German	1219	1245	26
Creative Writing	1236	1264	28
Chinese	1250	1280	30

Table 1: LMArena Elo style-corrected scores for Gemma IT and SiMPO models.

Class	IT	SimPO	Diff	Change
Linguistic Capabilities	6.25	8.99	+2.74	+43.8
Safety & Content Moderation	16.07	21.35	+5.28	+32.8
Information Processing	16.96	17.98	+1.01	+6.0
Format & Structure Control	10.71	11.24	+0.52	+4.9
User Interaction Management	14.29	12.36	-1.93	-13.5
Specialized Capabilities	28.57	23.60	-4.98	-17.4
Error Handling & Quality Control	6.25	4.49	-1.76	-28.1

Table 2: Class-level latent count changes (%) between `Gemma-2-9b-it` and `Gemma-2-9b-it-SimPO` models. Finegrained results are available in Appendix B

visible at the level of latent activation dynamics. Since we used BatchTopK method for training the crosscoder, only causally distinct latents are retained (Minder et al., 2025).

To better capture and interpret meaningful representational shifts, we revised our experimental setup to compare each fine-tuned model directly with the shared base model (`Gemma-2-9b-pt`). This change provided a clearer view of how instruction tuning and subsequent enhancements shape the model's internal structure, allowing us to trace the emergence and transformation of capability-related latents more effectively. See Figure 1b and 1c.

We extracted documents that strongly activated specific latents, used a large language model to annotate and categorize them (Mousi et al., 2023; Karvonen et al., 2025; Paulo et al., 2024), created a taxonomy of 31 capability categories grouped under 7 major classes (see Appendix A), measured the normalized frequency of each category across models, and analyzed the differences to understand the source of performance disparities.

3 Findings

To move beyond aggregate performance metrics, we conduct a representational analysis of latent concept shifts introduced by SimPO fine-tuning. Our comparison addresses two key research questions: **(i)** *Which latent capabilities are strengthened through SimPO's preference-driven fine-tuning?* and **(ii)** *What capabilities are diminished or deprioritized as a result of this optimization?*

Table 2 summarizes the distribution of latent categories across seven high-level classes, highlighting the most significant shifts in model behavior. We report and interpret these changes below.

3.1 Enhanced Capabilities in SimPO

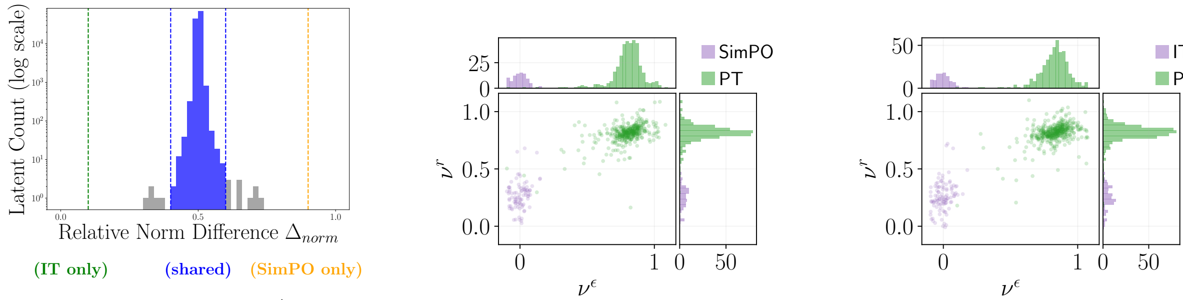
Among the latent concepts identified through model diffing, a notable subset becomes more prominent in the SimPO-enhanced model. These capabilities largely align with the goals of preference optimization, such as improving stylistic fluency, safety, and adherence to user instructions. Below, we summarize the most significant gains across categories like alignment, multilingual processing, and factual verification.

Linguistic capabilities (+43.8%): SimPO demonstrates enhanced multilingual capabilities. This explains the observed improvements in English, German and Chinese. However low-resource languages (Japanes, Korean) show regression on LMArena score and that was not captured by the latents possibly due to the lack of such data in the crosscoder training data.

Safety & Content Moderation (+32.8%): The most increase occurred in safety mechanisms, with Sexual Content Filtering showing the largest growth. Other notable increases include Minor Protection and Stereotype & Bias Detection. This suggests that SimPO prioritizes alignment with human values and safety guidelines.

3.2 Diminished Capabilities in SimPO

While SimPO strengthens many alignment-related capabilities, our analysis also reveals a set of latent concepts that decrease in prominence. These diminished features point to potential trade-offs, including reduced introspection, hallucination de-



(a) Latent distribution of Δ_{norm} for Simpo-IT crosscoder.

(b) PT-SimPO crosscoder latents.

(c) PT-IT crosscoder latents.

Figure 1: Identification of distinct latents across trained crosscoders. Figure 1a shows that no latents are unique to SimPO or IT models for Δ_{norm} thresholds < 0.1 or > 0.9. Figure 1b and 1c show the distribution of latents w.r.t. ν^ϵ and ν^r coefficient in Latent Scaling method. The purple latents are respectively unique to SimPO and IT in crosscoders PT-SimPO and PT-IT. We identified 92 latents unique to SimPO in PT-SimPO and 113 latents unique to IT in IT-PT crosscoder. These latents are taken further in downstream analysis as described in Section 2.2.

tection, and structured reasoning. We highlight the most notable regressions and discuss their implications for model reliability and robustness:

Code Generation & Math : Technical capabilities (e.g., related to code and math) show decrease of -17.4%, potentially indicating a shift toward general-purpose conversational abilities. This is reflected in the LMArena score.

4 Discussion

Our mechanistic analysis of differences between `Gemma-2-9b-it` and its SimPO-enhanced variant reveals that SimPO's performance improvements stem from specific capability shifts rather than uniform enhancements. The most significant changes align with the intended goals of preference optimization: improving safety, alignment with human preferences, and following instructions precisely.

The substantial increases in safety mechanisms (+32.8%) and instruction-following capabilities suggest that SimPO effectively incorporates human preferences regarding appropriate content and response formats. The dramatic enhancement in template and instruction following (+151.7%) explains why SimPO often produces more aesthetically pleasing and well-structured responses.

However, our analysis also reveals trade-offs. The decreased emphasis on hallucination detection (–68.5%) raises questions about whether SimPO sacrifices some self-monitoring capabilities in favor of producing more confident-sounding responses. Similarly, the reduction in query classification suggests that SimPO may take a more direct approach to generating responses than first analyzing the query type.

These findings help explain the mixed results observed in different evaluation contexts. In benchmark tests that reward accurate, well-formatted responses, SimPO's enhanced instruction-following and factual verification capabilities provide an advantage. In open-ended evaluations like lm-arena, human evaluators may be influenced by SimPO's improved stylistic qualities and reduced self-reference, even if some technical capabilities show modest decreases.

5 Conclusion

This work demonstrates that model diffing via crosscoders offers valuable insights beyond traditional benchmark evaluations. By mechanistically analyzing the latent representations that distinguish Gemma-2-9b-it from its SimPO variant, we reveal that performance differences stem from specific capability shifts rather than uniform improvements.

Our findings highlight how SimPO substantially enhances safety mechanisms, instruction-following capabilities, and multilingual processing while reducing emphasis on model self-reference and certain technical capabilities. These insights help explain both the strengths and limitations observed in different evaluation contexts.

Our work suggests that the field should move beyond leaderboard comparisons toward more nuanced analyses of what specifically changes when models are fine-tuned. Model diffing provides a promising framework for understanding performance disparities in terms of specific capabilities rather than opaque metrics, enabling more transparent and meaningful evaluations of LLM enhancements.

Limitations

Our study has several limitations. First, we focused on a single model pair (Gemma-2-9b-it and its SimPO variant), and the patterns we observed might not generalize to other models or fine-tuning approaches. Second, while our crosscoder-based analysis provides insights into capability differences, it cannot definitively establish causal relationships between these differences and specific performance outcomes.

Ethical considerations

We use LLM according to their intended use, and we used academic-purpose code that is shared for research objectives. AI tools were used to rephrase and improve exposition of sections of the paper.

References

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batch-topk sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating LLMs by human preference.

Fanar-Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, and 23 others. 2025. Fanar: An arabic-centric multimodal generative ai platform. *Preprint*, arXiv:2501.13944.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2025. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, and 1 others. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *arXiv preprint arXiv:2503.09532*.

Tianle Li, Anastasios Angelopoulos, and Wei-Lin Chiang. 2024. Does style matter? disentangling style and substance in chatbot arena. Accessed: 2025-05-19.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. 2024. Sparse crosscoders for cross-layer features and model diffing. *Transformer Circuits Thread*. Research Update.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

Julian Minder, Clément Dumas, Caden Juang, Bilal Chughtai, and Neel Nanda. 2025. Robustly identifying concepts introduced during chat fine-tuning using crosscoders. *arXiv preprint arXiv:2504.02922*.

Basel Mousi, Nadir Durrani, and Fahim Dalvi. 2023. Can LLMs facilitate interpretation of pre-trained language models? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3248–3268, Singapore. Association for Computational Linguistics.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. 2024. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pages 30811–30849. Curran Associates, Inc.

Shivalika Singh, Yiyang Nan, Alex Wang, Daniel D'Souza, Sayash Kapoor, Ahmet Üstün, Sanmi Koyejo, Yuntian Deng, Shayne Longpre, Noah Smith, and 1 others. 2025. The leaderboard illusion. *arXiv preprint arXiv:2504.20879*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

The Verge. 2025. Meta got caught gaming ai benchmarks. Accessed: 2025-05-19.

Benjamin Wright and Lee Sharkey. 2024. Addressing feature suppression in saes. In *AI Alignment Forum*, page 16.

Cheng Xu, Shuhao Guan, Derek Greene, M Kechadi, and 1 others. 2024. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zeyu Yun, Yubei Chen, Bruno Olshausen, and Yann Lecun. 2021. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 1–10.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P Xing, and 1 others. 2023. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*.

A Latent categorization

Code	Subcategory	Description
A. Safety & Content Moderation		
A.1	Harmful Content Detection	Identifies requests for violence, weapons, extremist content, or illegal activities. Activates when encountering text promoting harm or discrimination.
A.2	Request Refusal Mechanisms	Recognizes when to decline inappropriate requests. Provides explanations about ethical guidelines and limitations.
A.3	Jailbreak Detection	Identifies attempts to circumvent safety measures. Recognizes patterns like "evil trusted confidant" or constraint-based prompting.
A.4	Sexual Content Filtering	Detects explicit sexual content requests, especially involving inappropriate scenarios. Identifies content with taboo themes or non-consensual elements.
A.5	Minor Protection	Specifically focuses on protecting children in content generation. Detects requests involving minors in inappropriate contexts.
A.6	Stereotype & Bias Detection	Identifies racial, ethnic, or religious stereotyping. Detects when users request content that promotes discrimination.
B. Linguistic Capabilities		
B.7	Multilingual Processing	Identifies non-English languages in queries. Activates language-specific response modes across multiple scripts and languages.
B.8	Translation & Language Switching	Detects requests for translation between languages. Manages language transitions within conversations.
B.9	Grammar & Style Analysis	Evaluates grammatical correctness and writing quality. Identifies spelling, syntax, and structural issues in text.
C. Information Processing		
C.10	Summarization & Condensing	Detects requests to summarize longer content. Extracts key information while preserving core meaning.
C.11	Entity Recognition & Extraction	Identifies specific entities (people, organizations, terms) in text. Organizes and categorizes information from unstructured content.
C.12	Factual Verification	Checks consistency between summaries and source content. Verifies whether claims align with provided information.
C.13	Knowledge Boundary Recognition	Identifies when information falls outside the model's knowledge. Detects when the model should acknowledge limitations rather than confabulate.
D. User Interaction Management		
D.14	Query Classification	Categorizes types of user requests (questions, instructions, etc.). Determines appropriate response strategies.
D.15	Clarification Mechanisms	Detects ambiguous or vague queries requiring additional context. Manages follow-up questioning to gather necessary information.
D.16	Instruction Following	Processes and adheres to specific user instructions. Detects when constraints or formatting requirements are provided.
D.17	Conversation Management	Tracks conversation history and references to previous exchanges. Maintains context across multiple turns.
E. Format & Structure Control		
E.18	Structured Output Generation	Formats responses as lists, tables, or other organized structures. Maintains consistent formatting patterns.
E.19	JSON & API Integration	Converts text into machine-readable formats like JSON. Structures information for downstream processing.
E.20	Template Following	Detects and continues patterns established by examples. Adapts output to match specified formats.
F. Error Handling & Quality Control		
F.21	Self-Correction Mechanisms	Detects and acknowledges mistakes in previous responses. Provides corrections when errors are identified.

Continued on next page

Code	Subcategory	Description
F.22	Hallucination Detection	Identifies when the model is generating fabricated information. Recognizes factual inaccuracies in model outputs.
F.23	Truncation Awareness	Detects when responses are about to be cut off. Identifies incomplete or abruptly ending content.
G. Specialized Capabilities		
G.24	Code Generation & Analysis	Produces programming code across multiple languages. Identifies errors or inconsistencies in code snippets.
G.25	Professional Communication	Generates formal business content (emails, reports, etc.). Adapts tone for workplace and professional contexts.
G.26	Educational Explanation	Simplifies complex topics for different knowledge levels. Provides 'Explain Like I'm 5' (ELI5) content.
G.27	Creative Generation	Produces narratives, stories, and creative writing. Manages character development and dialogue.
G.28	Role-Playing & Persona Adoption	Adapts to specified character constraints. Maintains consistent persona characteristics.
G.29	Text Transformation	Edits, improves, and reformats existing content. Enhances clarity and readability while preserving meaning.
G.30	Model Self-Reference	Describes the model's own nature and capabilities. Manages disclosures about AI identity and limitations.

B Detailed Results

Table 3 shows fine-grained results for some of the top positive and negative changes between the two models.

Category	IT	SimPO	Diff (%)
Top positive changes (SimPO > IT)			
Sexual Content Filt.	4.46	7.87	+76.2
Template Following	1.79	4.49	+151.7
Instruction Following	1.79	4.49	+151.7
Multilingual Proc.	3.57	5.62	+57.3
Factual Verification	1.79	3.37	+88.8
Top negative changes (IT > SimPO)			
Model Self-Reference	8.04	4.49	-44.1
Query Classification	8.93	5.62	-37.1
Structured Output Gen.	7.14	4.49	-37.1
Hallucination Det.	3.57	1.12	-68.5
Code Generation	6.25	4.49	-28.1

Table 3: Top capability changes in terms of latent counts between Gemma-2-9b-it and Gemma-2-9b-it-SimPO models