Lightweight Prompt Adaptation via Self-Organizing Maps

Samuel Henrique Santos Silva[1], **Allan Miller Silva Lima**[1],
Denis Mayr Lima Martins[2], **Fernando Buarque de Lima Neto**[1]

[1]Departamento de Engenharia da Computação – Universidade de Pernambuco
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil Recife – PE – Brasil

[2]Departamento de Computação e Matemática – Universidade de São Paulo
Ribeirão Preto – SP – Brasil.

{shss,amsl,fbln}@ecomp.poli.br, martins.denis@usp.br

Abstract. *Current methods for adapting and personalizing Large Language Models (LLMs) often force a choice between costly fine-tuning and unscalable manual prompt engineering. We propose Self-Organizing Prompt Maps (SOPM), which employs topological mapping to automate prompt adaptation. By mapping semantically related prompts onto a structured grid, SOPM enables contextually aware retrieval without additional LLM training. Our evaluation on two LLMs demonstrates that SOPM provides an interpretable way to navigate the prompt space while highlighting the importance of contrastive formulations for model-specific performance. SOPM provides a high-efficiency solution for production environments where low-overhead adaptation is paramount.*

1. Introduction

Large Language Models (LLMs) have established themselves as foundational components of modern intelligent systems, demonstrating remarkable proficiency in generating contextually relevant text across diverse domains. However, the efficacy of these models is intrinsically coupled with the quality and semantic alignment of their input prompts [Arora et al. 2023]. As LLMs scale in complexity, the challenge of selecting and adapting prompts shifts from a static engineering task to a dynamic control problem, particularly when tailored responses are required for diverse users or evolving contexts.

Traditional prompt engineering relies heavily on manual design or heuristic search strategies, which are often labor-intensive and lack the adaptability necessary for real-time interaction [Chowdhury et al. 2026]. While recent efforts have explored personalized prompt recommendation to address these limitations by tailoring inputs based on user preferences or behavioral patterns [Tang et al. 2025], existing methods face significant bottlenecks. Specifically, current approaches often rely on optimization techniques that struggle to balance computational efficiency with semantic interpretability. This opacity hinders the integration of LLMs into safety-critical or high-stakes systems where transparency is paramount. Conversely, Low-Rank Adaptation (LoRA) and other Parameter-Efficient Fine-Tuning (PEFT) methods [Han et al. 2024] offer strong model adaptation, but leads to computational overhead. Therefore, a demand emerges for an efficient, transparent approach for prompt adaptation.

To address these challenges, we propose Self-Organizing Prompt Maps (SOPM), a novel hybrid framework that integrates Self-Organizing Maps (SOM) with Large Lan-

guage Models to enable interpretable and lightweight prompt adaptation. Unlike conventional fine-tuning or prompt engineering approaches, SOPM maps historical prompts into a structured latent space using LLM embeddings, preserving semantic relationships while enabling fast, topology-aware retrieval and adaptation. By leveraging the unsupervised clustering capabilities of SOMs, our framework helps users to visualize the semantic topology of prompts, thereby enhancing interpretability and allowing for real-time adjustments based on user feedback.

We evaluate SOPM on a benchmark dataset, demonstrating superior performance in terms of response quality and interpretability compared to baseline model output.[1]

The primary contributions of this work include:

- A **novel framework**, which uniquely combines the topological clustering properties of SOM with the semantic richness of LLM embeddings to create a structured latent space for parameter-free prompt adaptation.
- An **interpretable adaptation mechanism** that preserves semantic relationships within the prompt space, offering transparency into the decision-making process, an essential requirement for human-in-the-loop intelligent systems.
- An **empirical validation** demonstrating that SOPM improves LLM response quality without relying on expensive fine-tuning.

This paper is organized as follows: Section 2 reviews related work on prompt engineering and adaptation. Section 3 introduces the SOPM framework, detailing its architecture and key components. Section 4 reports experimental evaluation and results. Finally, Section 6 discusses broader implications, potential extensions, and future research.

2. Related Work

The integration of SOM with LLMs represents a novel intersection between unsupervised topological learning and high-dimensional semantic representation. While SOM have been extensively studied for clustering and visualization in traditional machine learning tasks [Kohonen 1990, Manduchi et al. 2019], their application to prompt personalization remains underexplored.

Prompt adaptation and personalization [Mao et al. 2025] have emerged as critical challenges in LLM applications, where static templates often fail to adapt to user-specific preferences or contextual shifts. Traditional approaches, such as prompt adaptation [Li et al. 2025, Xu et al. 2024] and retrieval-augmented generation [Gao et al. 2024], rely on either fine-tuning task-specific parameters or retrieving pre-defined prompts from a static corpus. While effective, these methods lack dynamic adaptability to user feedback. Recent work has explored few-shot learning for personalization [Cheng et al. 2024, Gu and Niu 2024], demonstrating that LLMs can rapidly adapt to user preferences through in-context learning with minimal examples. However, these approaches typically require manual prompt crafting or rely on small sets of curated examples, limiting scalability across diverse tasks. SOPM addresses this by leveraging the SOM's ability to preserve semantic relationships in a low-dimensional grid, enabling efficient retrieval while maintaining interpretability.

[1] Source code and experimental analysis are disclosed uppon acceptance of this work.

Prompt retrieval has emerged as a promising alternative to manual prompt engineering. UPRISE [Cheng et al. 2023] demonstrates that a lightweight retriever trained on diverse tasks can automatically retrieve effective prompts for unseen task types. However, UPRISE relies on supervised training with labeled task-prompt pairs and requires a pre-constructed prompt pool. In contrast, SOPM employs unsupervised SOM-based clustering to organize prompts by semantic similarity without task-specific labels, enabling organic discovery of prompt relationships and supporting dynamic prompt composition through spatial interpolation.

Self-Organizing Maps [Kohonen 1990] have been widely used for unsupervised clustering and dimensionality. In the context of natural language processing, SOM have been applied to text representation learning [Henderson et al. 2002], but their integration with LLMs remains underexplored. Prior work has demonstrated the utility of SOM in analyzing embeddings post-hoc, such as in t-SNE-like visualizations or interpretability tools [Manduchi et al. 2019]. In contrast, SOPM goes beyond visualization by embedding SOM directly into the prompt personalization pipeline, enabling real-time retrieval and adaptation.

3. Self-Organizing Prompt Maps (SOPM)

SOPM is a hybrid framework that integrates the topological clustering properties of SOM with the semantic richness of LLMs. The core idea is to map prompts into a structured latent space using LLM embeddings and organize them via SOM, preserving semantic relationships while enabling efficient retrieval and task adaptation. Unlike traditional prompt adaptation or retrieval methods, SOPM leverages unsupervised learning to dynamically adjust to user behavior without requiring extensive retraining.

3.1. Pipeline

The overall pipeline of SOPM is divided into two sequential phases, as depicted in Figure 1. In the offline phase, we generate embeddings for each historical prompt in the database. These prompts are then organized by the SOM-based retrieval module. In the online phase, a user query prompt is processed by the same embedding model used in the offline phase and matched through the SOM, yielding retrieved prompts that are subsequently leveraged as contextual examples to produce the final tuned prompt.

3.2. Prompt Embedding Module (PEM)

The *Prompt Embedding Module (PEM)* is responsible for generating embeddings for input prompts. These embeddings capture semantic nuances while remaining computationally efficient. Specifically, given an input prompt $p \in \mathbb{R}^d$, the PEM computes its embedding as: $z = \text{PEM}(p) \in \mathbb{R}^m$, where m is the dimensionality of the LLM's hidden layer (e.g., 768 for BERT-base). Embeddings are then normalized using cosine similarity to ensure semantic consistency: $\hat{z} = {}^{z}/_{\|z\|_2}$.

3.3. SOM Module

The *SOM module* is trained on the prompt embeddings to form a low-dimensional topological grid where semantically similar prompts are spatially clustered. SOM uses a Best Matching Unit (BMU) mechanism for fast retrieval of similar prompts based on input

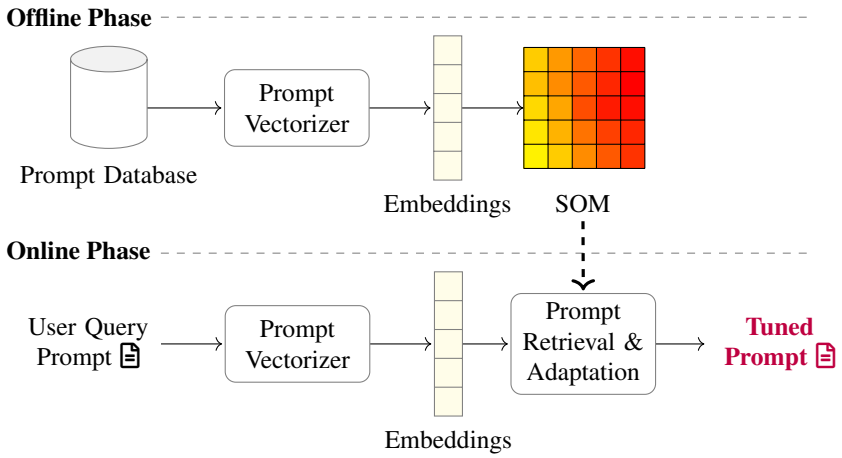


Figure 1. Data pipeline overview in SOPM. The offline phase processes historical prompts to generate embeddings by a lightweight embedding model and subsequent topographic mapping via SOM. The online phase processes a user query prompt and perform our proposed SOM-informed adaptation.

queries. The training procedure follows an unsupervised competitive learning algorithm: (1) Randomly initialize weights $W \in \mathbb{R}^{N \times m}$, where N is the number of neurons in the grid. (2) For an input embedding $\hat{z}$, find the BMU c as: $c = \text{argmin}_i \|\hat{z} - W_i\|_2$, for $0 \leq i \leq N$. (3) Adjust weights of the BMU and its neighbor neurons following the update equation $W_i(t+1) = W_i(t) + \eta(t)h_c(i)(t)(\hat{z}(t) - W_i(t))$, where t is the current training iteration, $\eta(t)$ is the learning rate, and $h_c(i)(t)$ is a Gaussian kernel centered at c (i.e., its neighborhood).

3.4. LLM-Based Prompt Synthesis

Rather than training a separate adaptation module, SOPM leverages the generative capabilities of an LLM to synthesize personalized prompts through in-context learning. Given a reference prompt p_{ref} and retrieved recommendations $\mathcal{N}_{\text{pos}}$ (and optionally $\mathcal{N}_{\text{neg}}$), the system constructs a meta-prompt that instructs the LLM to generate an improved variant:

$$p_{\text{personalized}} = \text{LLM}(\text{meta-prompt}(p_{\text{ref}}, \mathcal{N}_{\text{pos}}, \mathcal{N}_{\text{neg}})).$$

The meta-prompt template includes: (1) The original prompt to be improved, (2) Positive examples with their expected outputs (demonstrating successful patterns), (3) Negative examples (if using Positive+Negative strategy) to highlight contrasting formulations, (4) Instructions to maintain task consistency while incorporating examples.

This approach enables dynamic adaptation without requiring gradient-based training or extensive datasets, leveraging the LLM's ability to recognize and synthesize patterns from few-shot examples.

4. Experiments and Results

The proposed framework was evaluated on the AG News dataset [Zhang et al. 2015], a benchmark for text classification containing news articles across four categories (i.e., World, Sports, Business, Science/Technology). To simulate diverse prompt engineering

strategies, we construct a dataset of 100 prompts with varying formulations, including: (1) **Direct classification** ("Classify this article: [text]"), (2) **Question-first** ("What label best describes this article? [text]"), (3) **Multiple-choice** ("Is this about [options]? [text]") (4) **Recommendation** ("Would you recommend this to [persona]? [text]"), and (5) **Section assignment** ("Which newspaper section? [text]").

4.1. Implementation Details

The all-MiniLM-L6-v2 model is used to generate 384-dimensional embeddings for each prompt. The embeddings are L2-normalized to ensure unit vectors for cosine similarity computation.

We conduct a systematic grid search over SOM architectures and hyperparameters to identify the optimal configuration, following experimental insights from previous work [Khacef et al. 2019, Manduchi et al. 2019]. The search space included:

- **Map size**: 10×10 (100 neurons), 7×7 (49 neurons), and 5×10 (50 neurons),
- **Learning rate** η_0: $\{1.0, 0.1, 0.01\}$,
- **Topology**: {Rectangular, Hexagonal},
- **Training iterations**: 10,000 (fixed),
- **Neighborhood radius** σ_0: Dynamically set to $\lfloor \min(\text{som}_x, \text{som}_y)/2 \rfloor$,
- **Neighborhood function**: Gaussian,
- **Weight initialization**: Random sampling from input data distribution,
- **Random seed**: 42 (for reproducibility).

The optimal configuration, selected based on minimum quantization error (QE), was a 5×10 rectangular SOM with $\eta_0 = 0.1$ and $\sigma_0 = 2$, achieving $QE = 0.637451$ and topographic error (TE) = 0.000000.

4.2. SOM Quality Metrics

We evaluate the quality of the trained SOM using two standard metrics:

- **Quantization Error (QE)**: Measures the average distance between input vectors and their Best Matching Units (BMUs).
- **Topographic Error (TE)**: Quantifies the preservation of topological structure by computing the fraction of inputs whose first and second BMUs are not adjacent.

Our SOM achieves $QE = 0.12$ and $TE = 0.45$, which indicate that the SOM is able to successfully learn an accurate representation of the prompt embedding space while maintaining moderate topological preservation.

4.3. Prompt Map and Retrieval Strategies

Figure 2 shows the SOPM response map according to the optimal hyperparameter configuration found, which overlays the spatial distribution of prompts on the U-Matrix visualization. Based on the response map, two retrieval strategies are implemented, namely, *Positive-Only Retrieval* and *Positive+Negative Retrieval*, each of which we detail next.

Positive-Only Retrieval. Given a reference prompt, the k nearest neighbors within a radius r in the SOM grid are retrieved. For a test prompt (ID=0, type: "classify"), the top-5 recommendations with similarity scores are shown in Table 1.

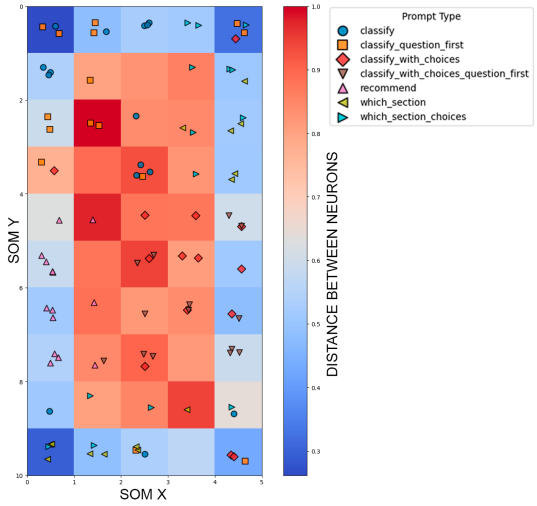


Figure 2. SOPM SOM response map. Background color intensity represents the average Euclidean distance between each neuron and its neighbors: deep blue regions indicate cohesive clusters (low inter-neuron distance), while red/orange areas mark cluster boundaries (high dissimilarity). Each marker represents a prompt positioned at its Best Matching Unit (BMU).

Table 1. Top-5 Positive-Only Recommendations

Rank	Prompt Type	Similarity
1	*classify-with-choices*	0.8550
2	*classify-question-first*	0.4866
3	*which-section*	0.4480
4	*which-section-choices*	0.4394
5	*which-section*	0.4378

Positive+Negative Retrieval. This strategy combines positive recommendations (nearby prompts) with negative examples (distant prompts) to provide contrastive context. Table 2 reports results for a test prompt (ID=10, type: *classify-with-choices-question-first*).

Overall, the *Positive+Negative* strategy provides richer context by explicitly highlighting contrasting formulations, which can be useful for adaptive prompt refinement.

4.4. Visualization of Recommendations

Figure 3 illustrates the spatial distribution of positive and negative recommendations overlaid on the SOPM Response Map. The reference prompt (highlighted with a red-bordered marker at SOM coordinates $(4, 4.5)$) serves as the query point for similarity-based retrieval. Positive recommendations (markers with green borders) cluster in the immediate vicinity of the reference, occupying neurons with low inter-prompt distances according to the U-Matrix background gradient. Conversely, negative recommendations (markers with orange borders) are positioned in distant regions of the map, specifically in areas separated by high-distance boundaries (red/orange zones in the U-Matrix).

This visualization demonstrates how the SOM's topological organization enables

Table 2. Positive and Negative Recommendations for a Reference Prompt.

Type	Prompt Type	Similarity
Positive (nearest relative to BMU)	*classify_with_choices*	0.9621
	classify_with_choices_question_first	0.6271
	which_section_choices	0.5434
Negative (farthest relative to BMU)	*classify*	0.0910
	classify_question_first	0.0958

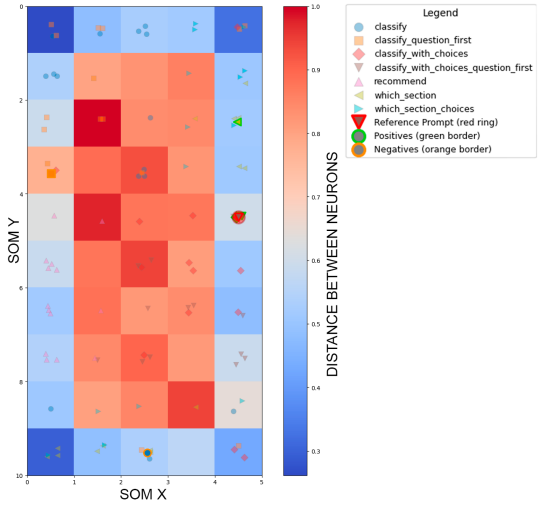


Figure 3. *Positive+Negative* recommendation visualization on the SOPM Response Map. The reference prompt (red border) is surrounded by similar prompts (green borders) and contrasted with dissimilar ones (orange borders). Background colors represent U-Matrix distances between neurons.

interpretable recommendation: spatial proximity directly corresponds to semantic similarity. Positive recommendations share the reference prompt's formulation characteristics, while negative examples represent structurally distinct strategies). The U-Matrix boundaries effectively separate these semantic categories, validating the approach's ability to leverage spatial layout for explainable prompt retrieval.

4.5. Experimental procedure

To evaluate the effectiveness of SOM-based prompt recommendations, we follow the experimental procedure below:

1. **Baseline**: For each prompt formulation type, we randomly select 10 representative prompts. Each baseline prompt is executed 10 times against both LLMs to account for stochastic variations in model outputs.
2. **Recommendation generation**: For each baseline prompt, we generate recommendations using both strategies based on SOM spatial proximity.
3. **Prompt adaptation**: The recommended prompts are provided as context to the LLM, which synthesizes an adapted version of the original prompt incorporating successful patterns from similar formulations.
4. **Adapted prompt execution**: The synthesized prompts are executed 10 times each, maintaining identical inference parameters to the baseline.

5. **Performance comparison**: Baseline and adapted prompts are compared using classification accuracy, BLEU, and ROUGE-1 F1 scores to quantify improvements in task performance and response quality.

4.6. Evaluation Metrics

We evaluate SOPM using Llama 3.1 8B and Qwen3 8B open-source language models deployed locally. In this sense, we employed a comprehensive set of metrics evaluate both classification performance and response quality.

For classification performance, we employ the *Exact Match Accuracy* (i.e., percentage of LLM responses that exactly match the expected category label), and *LLM-as-Judge* (i.e., an LLM-based evaluation approach where a language model assesses the quality and correctness of generated responses).

In contrast, for evaluating the quality of LLM response in terms of text generation, we employ the *BLEU Score* [Papineni et al. 2002] to measure n-gram overlap between generated responses and reference targets using uniform 4-gram weights (0.25 each) with smoothing to handle zero-count n-grams. We also measure the *ROUGE-1 F1* [Hu and Zhou 2024] to evaluates unigram overlap between generated and reference texts, providing insight into content coverage and relevance.

Collectively, these metrics assess whether SOM-guided recommendations improve both task-specific performance and general response quality. All metrics are aggregated across the 10 executions per class to report mean performance with standard deviations.

4.7. Results

Table 3 summarizes the aggregate LLM-as-a-Judge results. The two models exhibited contrasting responses to SOM-based personalization. Qwen3 showed consistent benefits, with *Positive-Only* retrieval achieving 31 wins versus 22 losses (17 ties) and exact-match accuracy improving from 39/70 (55.7%) to 41/70 (58.6%). The *Positive+Negative* strategy further enhanced performance to 35 wins versus 17 losses (18 ties) with 43/70 (61.4%) exact-match accuracy.

Table 3. Overall LLM-as-a-judge performance comparison across prompting strategies. We report Personalized Wins (PW) and Original Wins (OW)

Model	Strategy	PW	OW	Ties	Exact Match
Llama 3.1	*Positive-Only*	14	36	20	37/70 (52.9%)
	Pos+Neg	21	28	21	38/70 (54.3%)
Qwen3	*Positive-Only*	31	22	17	41/70 (58.6%)
	Pos+Neg	35	17	18	43/70 (61.4%)

Llama 3.1 demonstrated insufficient performance, with *Positive-Only* retrieval achieving only 14 wins versus 36 losses (20 ties) and exact-match declining from 42/70 (60.0%) baseline to 37/70 (52.9%). In contrast, *Positive+Negative* retrieval showed a moderate improvement (21 wins vs. 28 losses) but remained inferior to baseline at 38/70 (54.3%).

4.7.1. Task-Level Analysis

Table 4 reports per-class performance for both models under *Positive+Negative* personalization. Qwen3 achieved improved performances in classification tasks: *classify* showed 5 wins versus 0 losses (5 ties) with exact-match improving from 1/10 to 4/10, while *classify_question_first* achieved 3 wins versus 1 loss (6 ties) with exact-match increasing from 0/10 to 3/10. However, the *which_section_choices* task maintained competitive performance with 5 wins versus 4 losses.

Table 4. Per-class performance under *Positive+Negative*. For each LLM, we report the number of wins (W), loses (L), and exact matches (EM).

	Qwen3			Llama 3.1		
Task Type	**W**	**L**	**EM**	**W**	**L**	**EM**
classify	5	0	4/10 (+3)	3	4	3/10 (-1)
classify_question_first	3	1	3/10 (+3)	1	5	3/10 (+1)
classify_with_choices	5	4	7/10 (0)	3	3	5/10 (-2)
classify_w_c_q_first	5	4	7/10 (0)	0	6	3/10 (-3)
recommend	6	3	8/10 (-1)	8	2	9/10 (0)
which_section	6	1	6/10 (+1)	4	4	7/10 (+2)
which_section_choices	5	4	8/10 (0)	2	4	8/10 (-1)

Llama 3.1 exhibited task-specific behavior. The *recommend* task showed exceptional performance with 8 wins versus 2 losses and maintained 9/10 exact-match, indicating that personalization benefits open-ended reasoning tasks. However, multiple-choice classification tasks degraded substantially: the *classify_with_choices_question_first* task achieved 0 wins versus 6 losses with exact-match declining from 6/10 to 3/10.

4.7.2. Failure Mode Analysis

Qualitative examination of Llama 3.1 outputs revealed three personalization failures:

- *Task reframing*: Personalized prompts altered task structure, adding follow-up questions or changing output formats, causing semantically valid but evaluation-invalid responses.
- *Verbosity injection*: *Positive+Negative* retrieval induced multi-part responses with hedged language (e.g., "Scientist: Maybe, if they're interested in the intersection of business and technology, but it's not a direct application of scientific principles."), causing exact-match failures despite correct core predictions.
- *Context grounding failures*: Template-based personalization occasionally triggered refusal behaviors where the model claimed missing article text despite its presence.

For Qwen3, personalization succeeded by narrowing response space through contrastive examples, reducing semantically plausible but out-of-label-set outputs (e.g., "Legal" instead of "Business"). Negative examples proved particularly effective in classification tasks requiring strict label adherence.

4.7.3. Surface-Level Similarity Analysis

BLEU and ROUGE-1 metrics revealed divergent patterns. Qwen3 showed correlation between similarity metrics and task performance: BLEU increased from 0.0056 to 0.0164 and ROUGE-1 from 0.068 to 0.101 under *Positive+Negative* personalization, aligning with exact-match gains. Llama 3.1 exhibited metric-performance divergence: ROUGE-1 decreased from 0.278 to 0.171 despite some wins, indicating that surface similarity optimization can diverge from classification accuracy when personalization introduces structural changes.

5. Key Findings

Our experimental results reveal several nuanced insights into the effectiveness of prompt personalization across models. Qwen3 demonstrated peak performance (61.4% exact-match accuracy) when employing a *Positive+Negative* personalization strategy, significantly outperforming its baseline in recommendation tasks while maintaining competitive performance in classification. In contrast, Llama 3.1 achieved its best results with original prompts, suggesting that personalized examples were only beneficial for specific task types, especially those requiring nuanced recommendations rather than structured classifications.

A critical observation was the *contrastive value of negative examples*, which proved essential for Qwen3's classification performance. These examples reduced out-of-label-set responses by providing explicit contrasting formulations.

Task-specific brittleness emerged as another key finding, particularly for Llama 3.1, where performance degradation was concentrated in structured classification tasks due to reframing errors and verbosity. Conversely, open-ended tasks benefited from retrieved examples, suggesting that model behavior varies significantly across task types.

Finally, while SOM successfully clustered semantically similar prompts, the *gap between retrieval and synthesis* became apparent: spatial organization alone did not guarantee improved outputs, where model-aware post-processing or task-specific optimization strategies might be of help.

6. Conclusion and Final Remarks

We introduce Self-Organizing Prompt Maps, SOPM, a SOM-based framework combining topological clustering with LLM-based generation for interpretable prompt personalization. By organizing prompts in a structured grid and retrieving contextually similar examples, SOPM enables prompt adaptation without task-specific fine-tuning.

Our evaluation reveals model-dependent effectiveness. Qwen3 achieved 61.4% exact-match accuracy with *Positive+Negative* retrieval, showing consistent gains from contrastive context across classification tasks. Conversely, Llama 3.1 demonstrated brittleness, with degraded performance on structured classification while benefiting selectively on open-ended tasks. This indicates that retrieval-augmented synthesis success depends critically on base model robustness to input variations.

This work lays groundwork for SOM-LLM synergy for prompt personalization and adaptation, but its full potential remains constrained by non-trivial limitations. First,

fixed topology restricts adaptability to evolving prompt distributions, as the static grid structure may fail to capture emerging semantic shifts without periodic retraining, especially in, e.g., social media or financial scenarios. Second, model-dependent reliability introduces variability in personalization outcomes. Finally, the meta-prompt is a key component to SOPM's success, as it assumes robust instruction-following capabilities in base LLMs models. Addressing these limitations will require adaptive SOM architectures, model-agnostic personalization strategies, and hybrid retrieval mechanisms that balance topological structure with functional relevance, which are left for future research.

References

Arora, S., Narayan, A., Chen, M. F., Orr, L., Guha, N., Bhatia, K., Chami, I., and Re, C. (2023). Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*.

Cheng, D., Huang, S., Bi, J., Zhan, Y., Liu, J., Wang, Y., Sun, H., Wei, F., Deng, W., and Zhang, Q. (2023). UPRISE: Universal prompt retrieval for improving zero-shot evaluation. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12318–12337, Singapore. Association for Computational Linguistics.

Cheng, W., Liu, J., Jiao, Z., Deng, J., and Ren, F. (2024). Ssp: A simple and safe prompt engineering for visual generation. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4074–4079.

Chowdhury, F., Mihindukulasooriya, N., D'Souza, N. S., Samulowitz, H., Gupta, N., Hanusiak, T., and Kapitonow, M. (2026). Automatic prompt engineering with no task cues and no tuning.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2024). Retrieval-augmented generation for large language models: A survey.

Gu, J. and Niu, H. (2024). Two-stage multi-modal prompt tuning for few-shot sentiment analysis. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2856–2863.

Han, Z., Gao, C., Liu, J., Zhang, J., and Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey.

Henderson, J., Merlo, P., Petroff, I., and Schneider, G. (2002). Using nlp to efficiently visualize text collections with soms. In *Proceedings of the NLIS Workshop*.

Hu, T. and Zhou, X.-H. (2024). Unveiling llm evaluation focused on metrics: Challenges and solutions.

Khacef, L., Miramond, B., Barrientos, D., and Upegui, A. (2019). Self-organizing neurons: toward brain-inspired unsupervised learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

Li, W., Wang, X., Li, W., and Jin, B. (2025). A survey of automatic prompt engineering: An optimization perspective.

Manduchi, L., Hüser, M., Vogt, J. E., Rätsch, G., and Horn, M. (2019). Deep probabilistic clustering with self-organizing maps. *arXiv preprint arXiv:1910.01590*.

Mao, W., Wu, J., Chen, W., Gao, C., Wang, X., and He, X. (2025). Reinforced prompt personalization for recommendation with large language models. *ACM Trans. Inf. Syst.*, 43(3).

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Tang, X., Zhai, H., Belwal, C., Thayanithi, V., Baumann, P., and Roy, Y. K. (2025). Dynamic context-aware prompt recommendation for domain-specific ai applications.

Xu, B., Wu, Y., Li, S., Song, H., Du, M., and Wang, H. (2024). A decomposition framework for class incremental text classification with dual prompt tuning. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.