

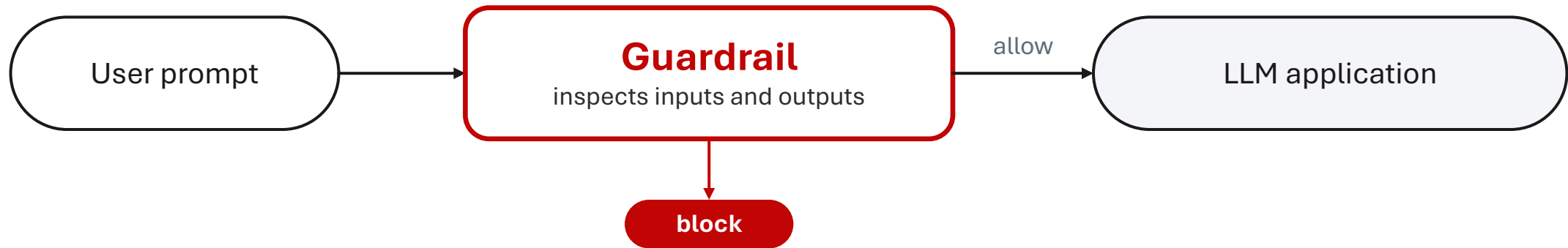
# kNNGuard

Turning LLM Hidden Activations into a Training-Free Configurable  
Guardrail

**Mahmoud Abdelfattah, Hamid Nasiri, Peter Garraghan**

23/10/2026

# LLM Guardrails



## Topical guardrails

Allow only requests within the application's domain, e.g. coding



## Safety & security guardrails

Block harmful requests, jailbreaks and prompt injections

# Current Approaches



## Training-based

Fine-tuned classifiers, e.g. Llama Guard 3, Prompt Guard 2

- ✓ Accurate on their training data
- ✗ Retraining for every new domain
- ✗ Slow: ~100–450 ms per prompt
- ✗ Evaded by simple perturbations

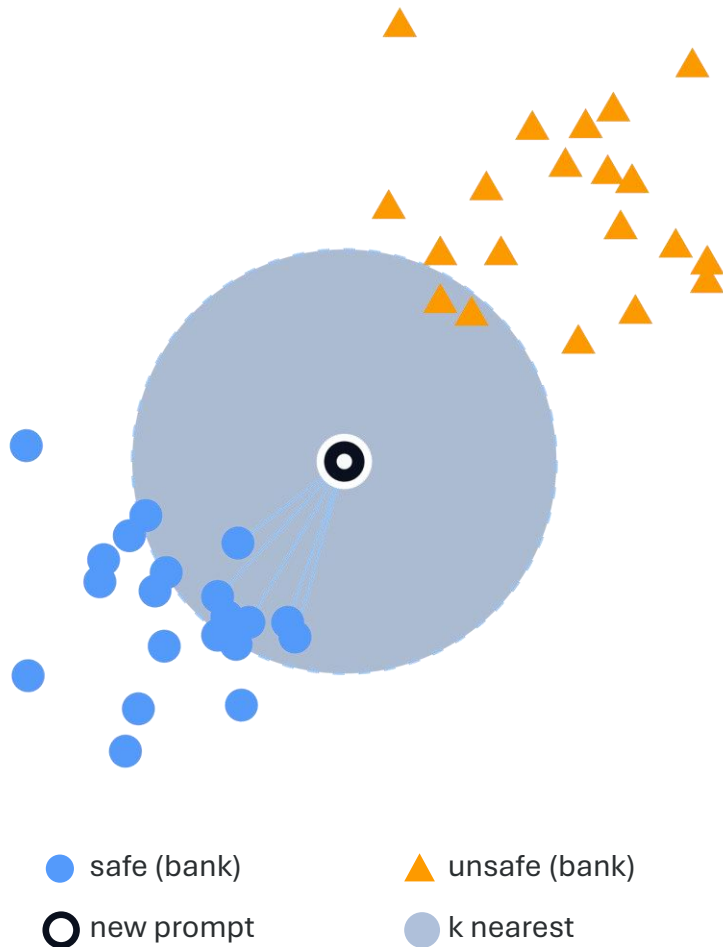


## Training-free

Embedding kNN over a labelled bank of prompts

- ✓ Fast: about 4 ms per prompt
- ✓ Adapt by swapping bank/policies
- ✗ Relies on surface similarity
- ✗ High false-positive rate

**Goal:** fine-tuned accuracy with training-free speed and flexibility.



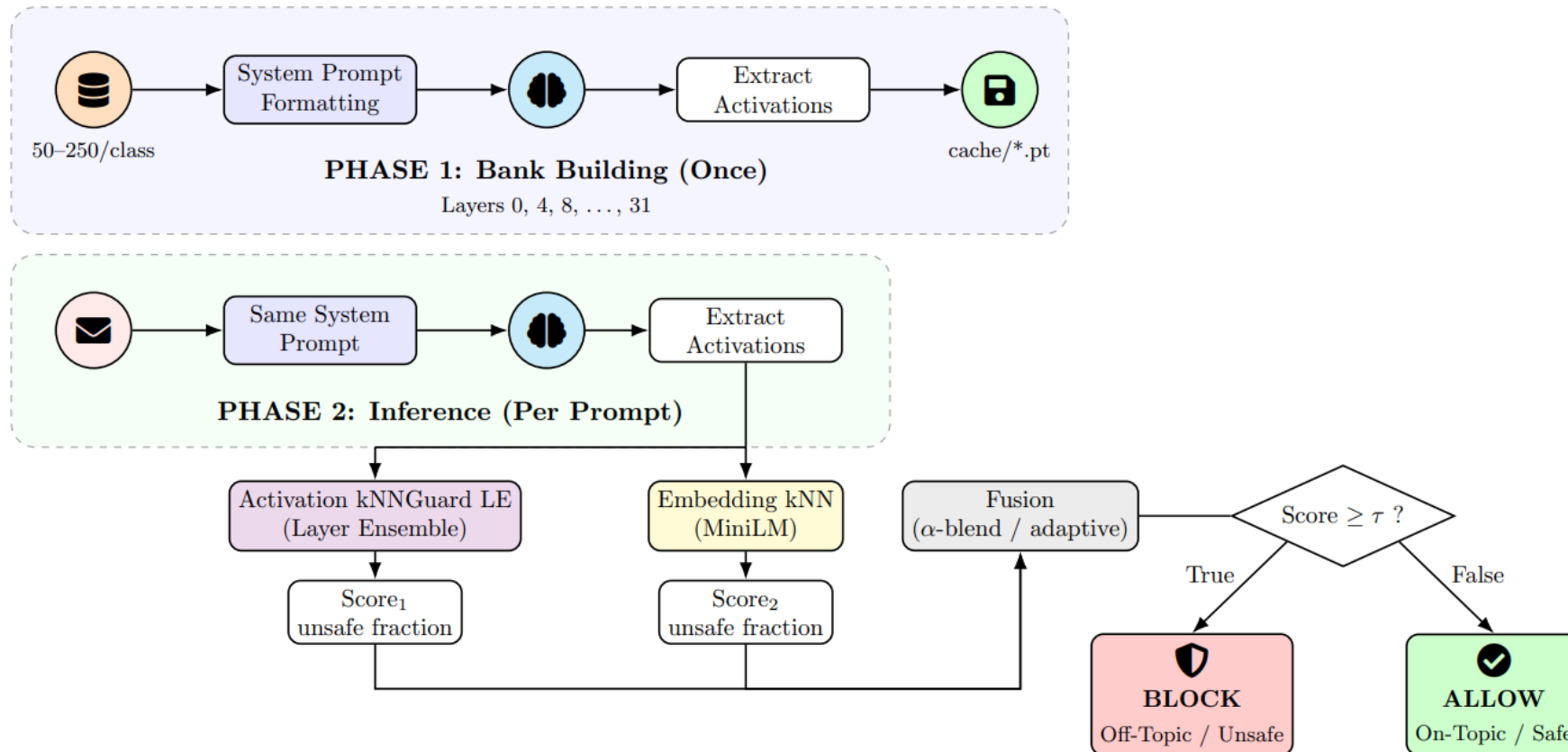
# kNNGuard's Idea

Interpretability research shows LLM activations include distinct features (e.g. the refusal direction; Ardit et al., 2024)

**Use the LLM's own hidden activations as the classifier.**

# Methodology

## 2 Main Phases: Bank Building (Once) & Inference (Per Prompt)



# kNNGuard LE (Layer Ensemble)

Part of phase 2

**Step 1**

$$h^{(\ell)}(\tilde{x})$$

**Extract**

Across 9 layers

**Step 2**

$$s_{\ell} = (1/k) \sum y_i$$

**Per-layer kNN**

Unsafe share of kNN

**Step 3**

$$v_{\ell} = \mathbb{I}[s_{\ell} > \tau]$$

**Vote**

Each layer votes

**Step 4**

$$s_{\text{act}} = \sum a_{\ell} v_{\ell}$$

**Combine**

Weighted vote share

**Layer weights**  $w_{\ell}$  · config option layer\_weighting

**Uniform**  $\alpha_{\ell} = 1 / M$

Every layer counts equally

**Fisher**  $\alpha_{\ell} = \text{softmax}(J_{\ell})$

Layers that separate the classes count more

# kNNGuard FE (Fused Ensemble)

Part of phase 2

**Combine the activation score with an embedding score, applying adaptive fusion.**



## **Clear winner**

If one score is much more confident (confidence gap > 0.1), use that score.

## **No clear winner**

If confidence is similar, average the scores, weighted by confidence.

Confidence = distance of a score from threshold (0.5) · block if the final score ≥ 0.5

# Experimental Setup



## 6 domains 16 datasets

4000 test per domains (2000 for jailbreak and prompt injection domains)



## Configuration

$k = 13$

Prompts per class: 50

Backbone model: Llama-3.1-8B-Instruct

Threshold ( $\tau$ ) = 0.5



## Guardrails

- Llama Nemotron Guard V1 (All domains)
- Llama Nemotron Safety Guard V2 (Safety)
- Llama Guard 3 (Safety)
- Prompt Guard 2 (Security)
- Embedding kNN (All domains)



## Metrics

- F1 Score
- FPR (False Positive Rate)
- FNR (False Negative Rate, =ASR)
- Latency

# Datasets

Bank and test prompts are from different datasets.

| Domain              | Bank Safe                                             | Bank Unsafe                                | Eval Safe                                               | Eval Unsafe                                       |
|---------------------|-------------------------------------------------------|--------------------------------------------|---------------------------------------------------------|---------------------------------------------------|
| Coding Instructions | MBPP <a href="#">Austin et al. (2021)</a>             | Alpaca <a href="#">Taori et al. (2023)</a> | Code ( <i>instr. col</i> ) <a href="#">Tarun (2023)</a> | Dolly-15k <a href="#">Conover et al. (2023)</a>   |
| Coding Outputs      | PromptSet <a href="#">Pister et al. (2024)</a>        |                                            | Code ( <i>output col</i> ) <a href="#">Tarun (2023)</a> |                                                   |
| Medical             | MedMCQA <a href="#">Pal et al. (2022)</a>             |                                            | ChatDoctor <a href="#">Li et al. (2023)</a>             |                                                   |
| Safety              | Aegis Safety 2.0 <a href="#">Ghosh et al. (2025)*</a> |                                            | Safety Benchmark <a href="#">qualifire (2025)</a>       | Prompt Safety <a href="#">SalKhan12 (2025)</a>    |
| Jailbreak           | Jailbreak Classification <a href="#">Hao (2023)*</a>  |                                            | PI Benchmark <a href="#">rogue-security (2026)</a>      | WildJailbreak <a href="#">Jiang et al. (2024)</a> |
| Prompt Injection    | BIPIA-GPT <a href="#">Alamsabi et al. (2026)*</a>     |                                            | Deepset PI <a href="#">deepset (2023)</a>               | PI Dataset <a href="#">NeurAlchemy (2026)</a>     |

\* Domains in which the same source dataset was used for both the safe and unsafe classes. In those cases, the two banks were constructed by using the different labels provided in the dataset for safe/unsafe samples.

# Results – Topical Domains

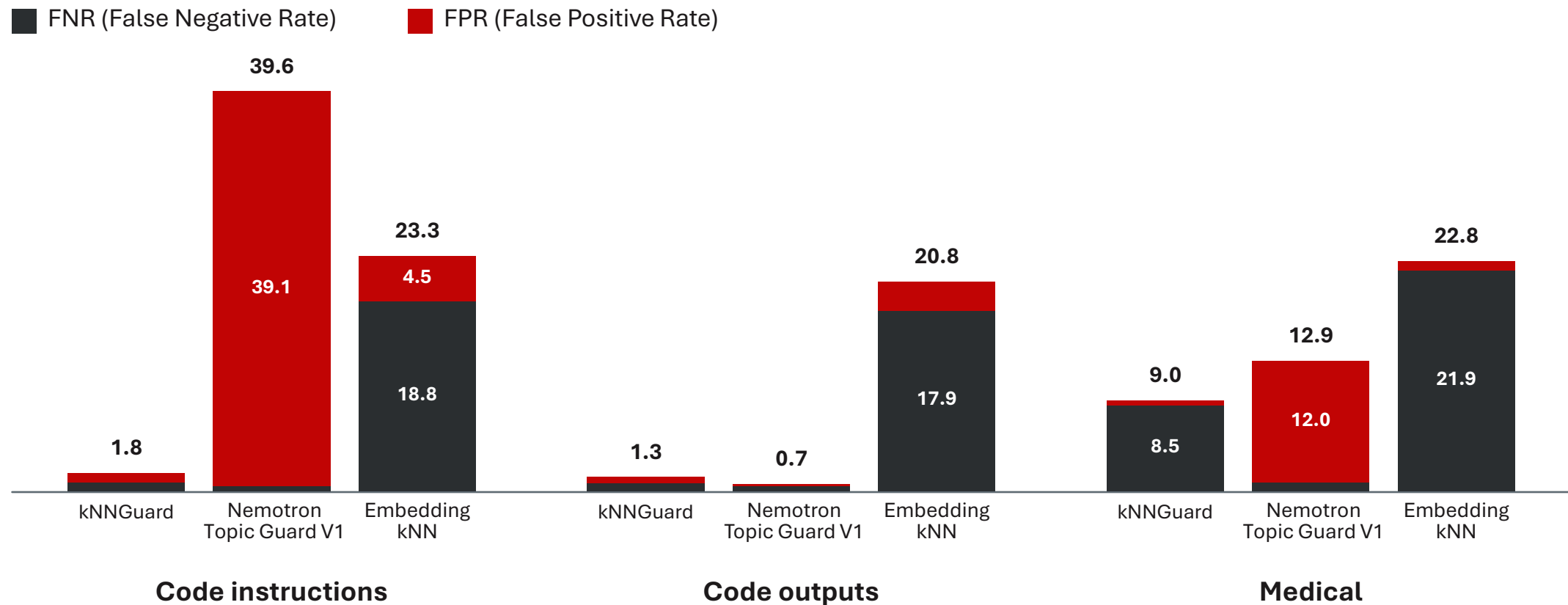
## F1 Scores & Latency

| Guardrail                     | Code instructions | Code outputs | Medical     | Latency    |
|-------------------------------|-------------------|--------------|-------------|------------|
| <b>kNNGuard FE</b>            | <b>99.1</b>       | 99.3         | <b>95.3</b> | 46.8       |
| Llama Nemotron Topic Guard V1 | 83.4              | <b>99.6</b>  | 93.9        | 126.1      |
| Embedding-kNN                 | 87.5              | 88.7         | 87.3        | <b>4.0</b> |

F1 in %, latency in ms per prompt. Bold: best in each column.

# Results – Topical Domains

## Error Rates



# Results – Security/Safety Domains

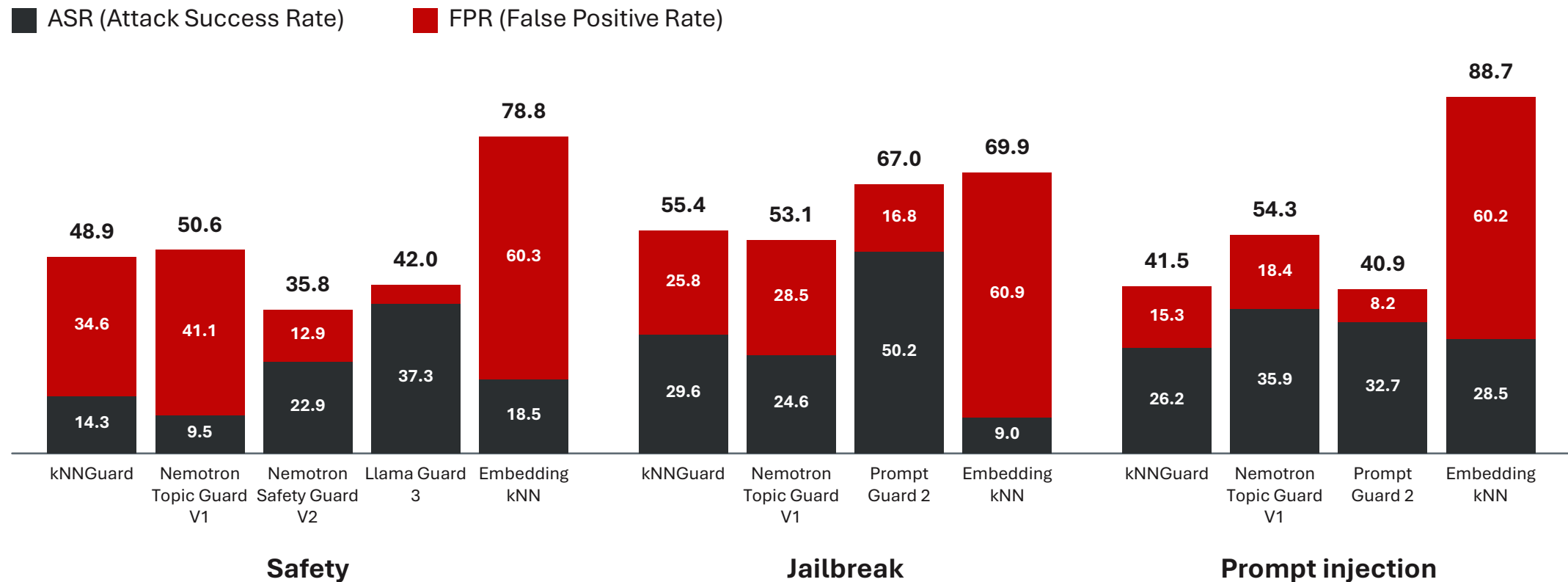
## F1 Scores & Latency

| Guardrail                | Safety      | Jailbreak   | Prompt injection | Latency    |
|--------------------------|-------------|-------------|------------------|------------|
| <b>kNNGuard FE</b>       | 73.7        | 75.0        | <b>82.2</b>      | 46.8       |
| Nemotron Topic Guard V1  | 73.5        | 77.5        | 75.0             | 126.1      |
| Nemotron Safety Guard V2 | <b>79.2</b> | —           | —                | 454.7      |
| Llama Guard 3            | 74.2        | —           | —                | 104.4      |
| Prompt Guard 2           | —           | 61.9        | 79.0             | 9.6        |
| Embedding-kNN            | 61.9        | <b>78.4</b> | 73.6             | <b>4.0</b> |

F1 in %, latency in ms per prompt. Bold: best in each column. —: not evaluated on that domain

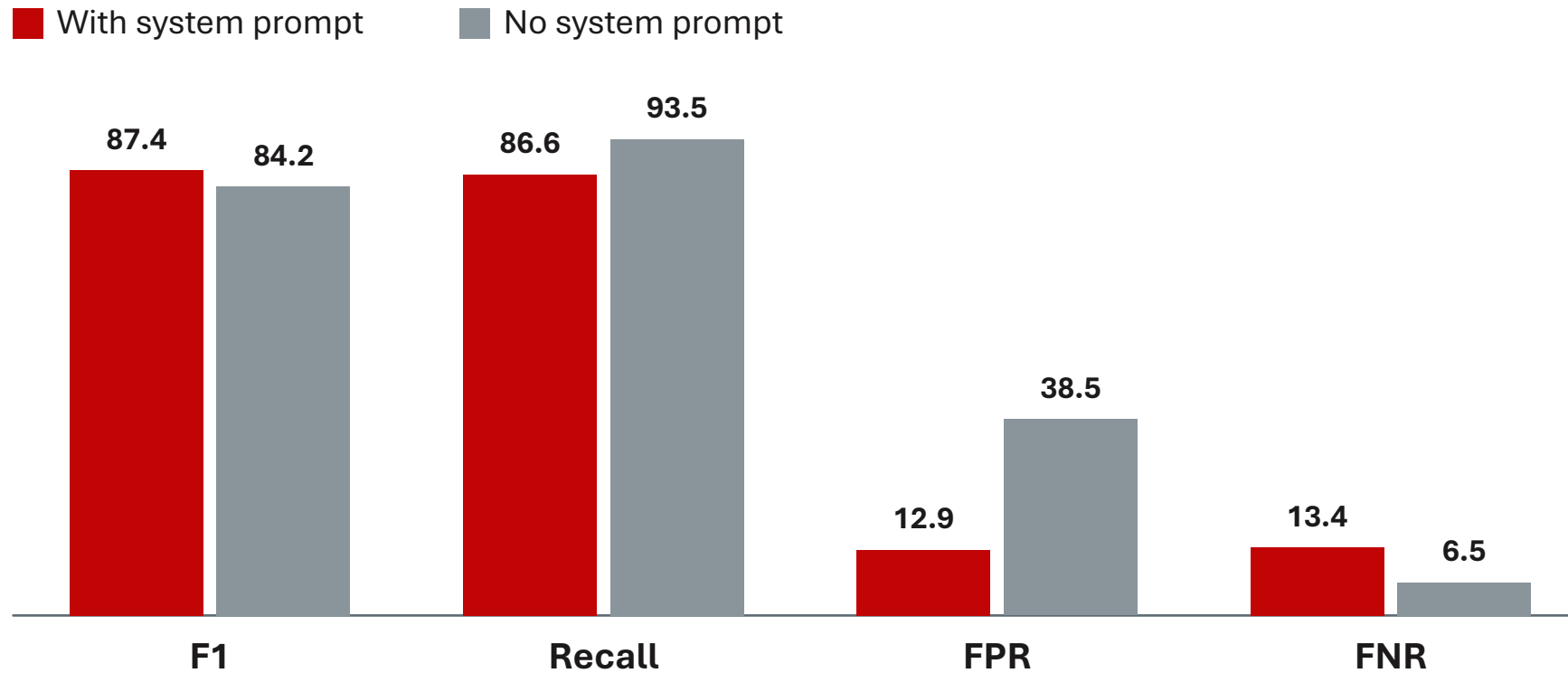
# Results – Security/Safety Domains

## Error Rates



# System Prompt

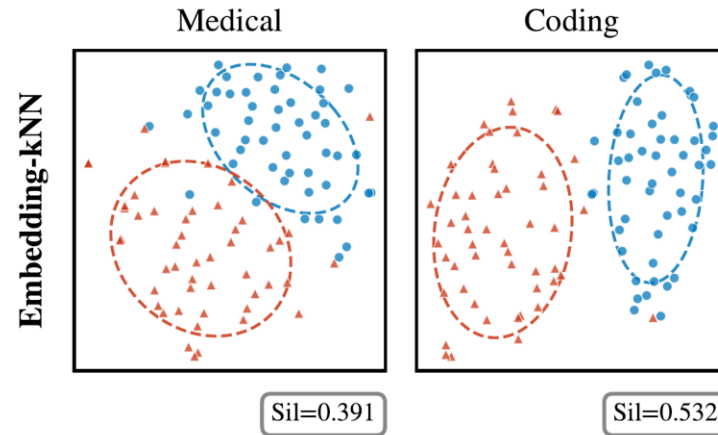
A domain system prompt cuts false positives by over 65%



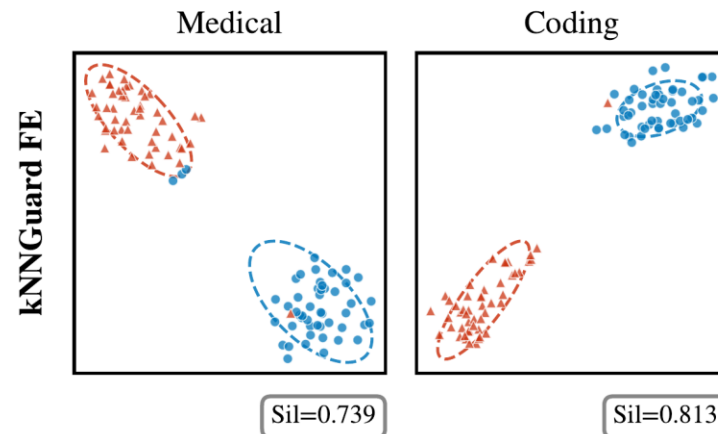
Metrics in %. Averaged over 6 domains

# Analysis – Representation Space

Embedding kNN (MiniLM)  
(top)



kNNGuard FE (ours)  
(bottom)



Safe / In-domain Unsafe / Out-of-domain

Silhouette scores

Medical 0.39 → 0.74 (+89%)

Coding 0.53 → 0.81 (+53%)

# Analysis – Adaptation Cost

Adapting kNNGuard to a new domain takes seconds, not hours.

## LoRA fine-tuning

13,854 prompts



**25,092 s** (6.97 hrs)

## kNNGuard bank

13,854 prompts



**413 s** (6.9 min) · 60.8x faster

## kNNGuard bank

100 prompts, as used



**7.95 s** · 3,156x faster

Time comparison to adapt Llama-3.1-8B-Instruct

# Limitations

**01**

## **No guardrail is best everywhere**

Security domains remain challenging for every guard

**02**

## **Static benchmarks only**

Adaptive attacks against the bank are untested

**03**

## **Backbone-sensitive on prompt injection**

F1 varies widely across backbone models

# Future Work

**01**

## **Self-updating banks**

Adapt to emerging attacks without manual curation

**02**

## **Agentic systems**

Monitor tool outputs, not just user prompts

**03**

## **Richer system prompts**

Test curated, detailed domain policies

# Takeaways

01

## **Training-free guardrail**

Hidden activations of a frozen LLM plus 100 labelled prompts give 87.4% average F1 across six domains, with no gradient updates.

02

## **Fast to run, fast to adapt**

46.8 ms per prompt, 2.7x faster than Nemotron Topic Guard V1. A new domain needs only a new bank, built in 7.95 s.

03

## **Competitive results with SOTA (State-of-The-Art) Guardrails**

Achieving similar or better F1 scores compared to guardrails across all domains, including specialised security/safety guardrails.

# Thank you.

# Questions?

**Mahmoud Abdelfattah · Hamid Nasiri · Peter Garraghan**  
**[m.abdelfattah1@lancs.ac.uk](mailto:m.abdelfattah1@lancs.ac.uk)**