Targeting Alignment:
Extracting Safety Classifiers of Aligned LLMs

Jean-Charles Noirot Ferrand, **Yohan Beugin**, **Eric Pauley**, **Ryan Sheatsley**, **Patrick McDaniel**

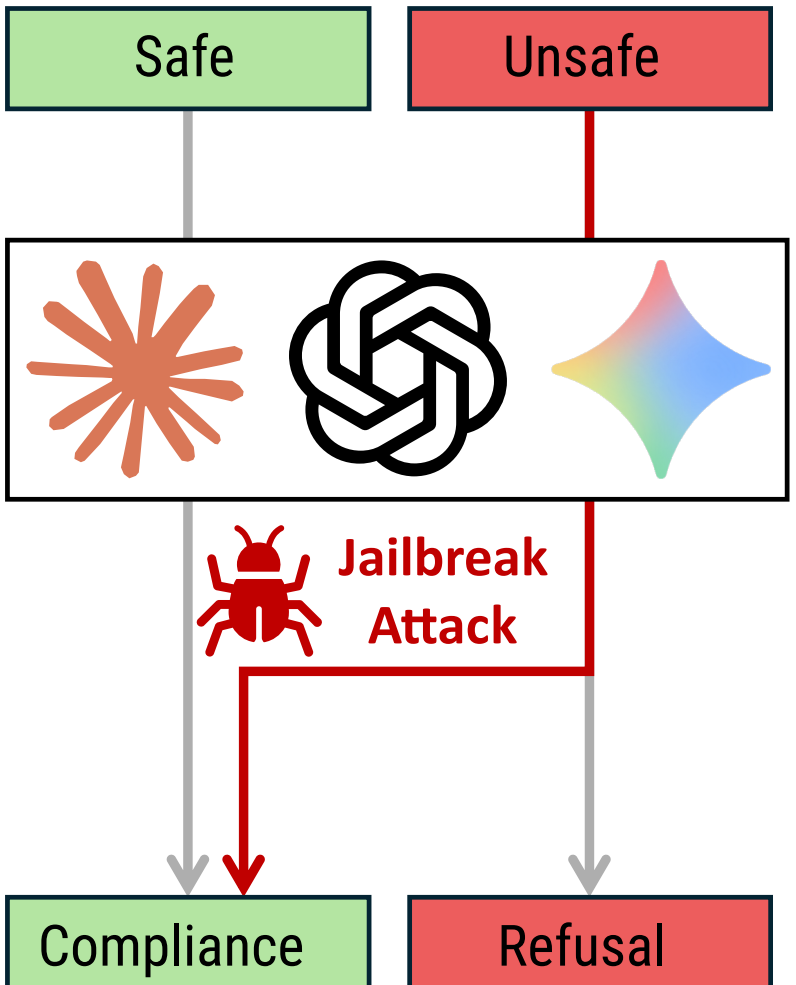
March 2026 • SaTML

Computer Sciences
SCHOOL OF COMPUTER, DATA & INFORMATION SCIENCES
UNIVERSITY OF WISCONSIN–MADISON

MADS&P

Background

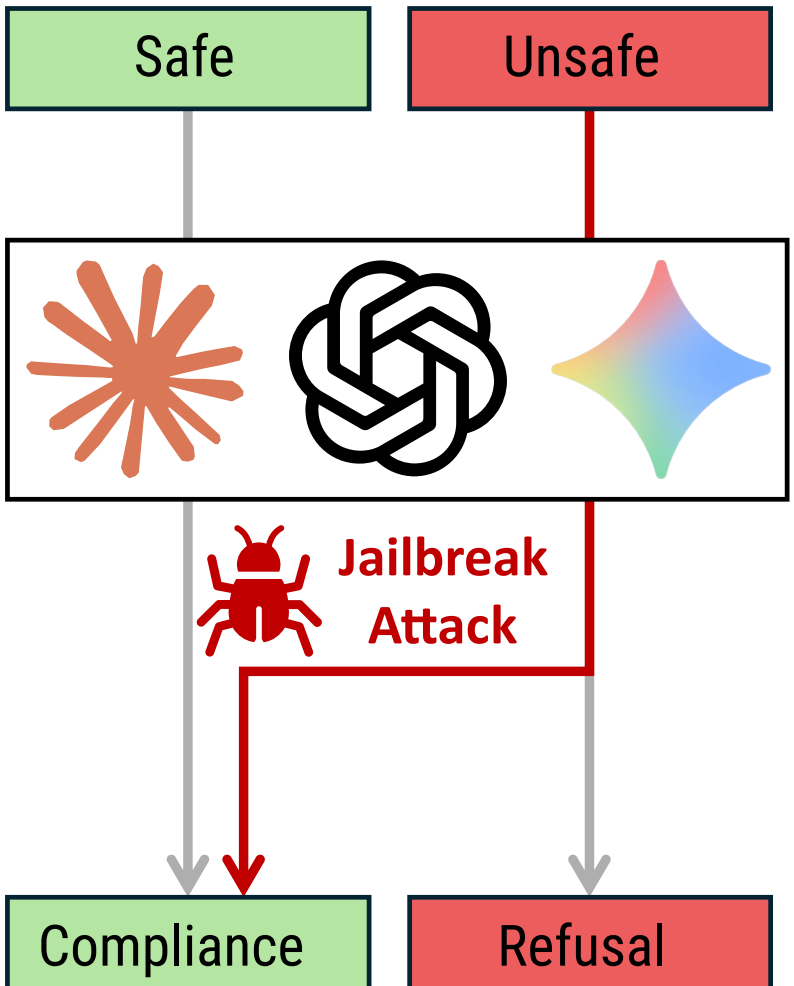
- Efforts to make models **safe/aligned**
  - Compliance for Safe, Refusal for Unsafe
  - Fails against **jailbreak** attacks



Background

- Efforts to make models **safe/aligned**
  - Compliance for Safe, Refusal for Unsafe
  - Fails against **jailbreak** attacks

- Many jailbreak attacks, two threat models
  - **Black-box:** efficient but limited insights
  - **White-box:** rich information but expensive

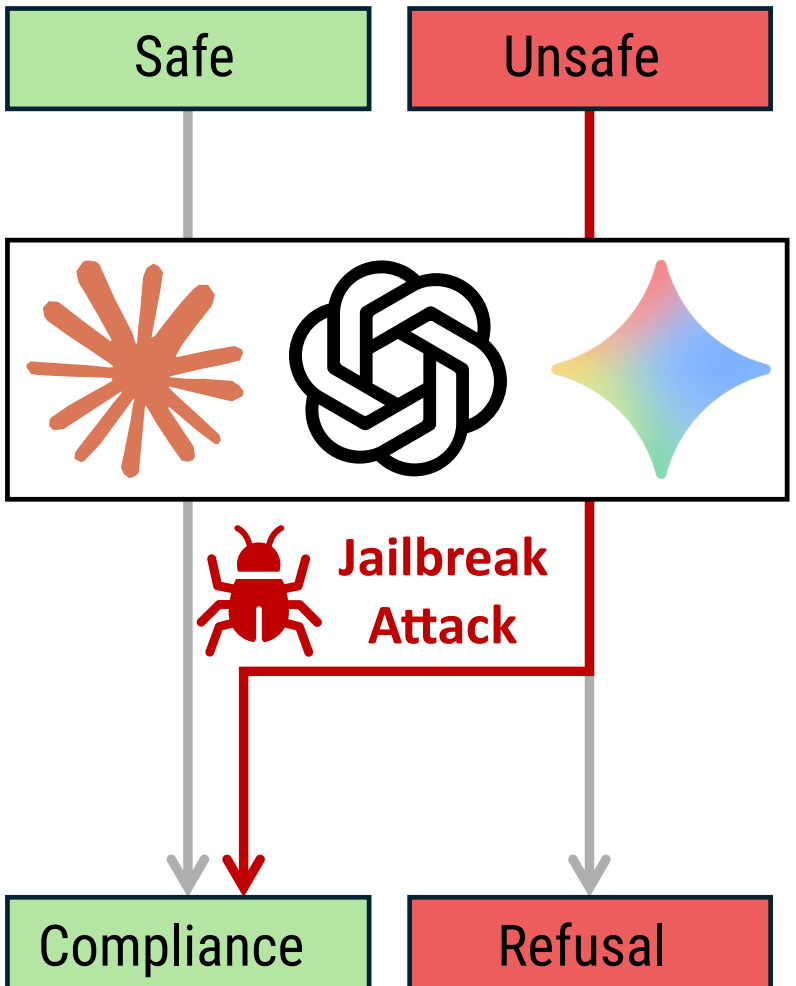


Background

- Efforts to make models **safe/aligned**
  - Compliance for Safe, Refusal for Unsafe
  - Fails against **jailbreak** attacks

- Many jailbreak attacks, two threat models
  - **Black-box:** efficient but limited insights
  - **White-box:** rich information but expensive

→ **White-box attacks are not attractive**

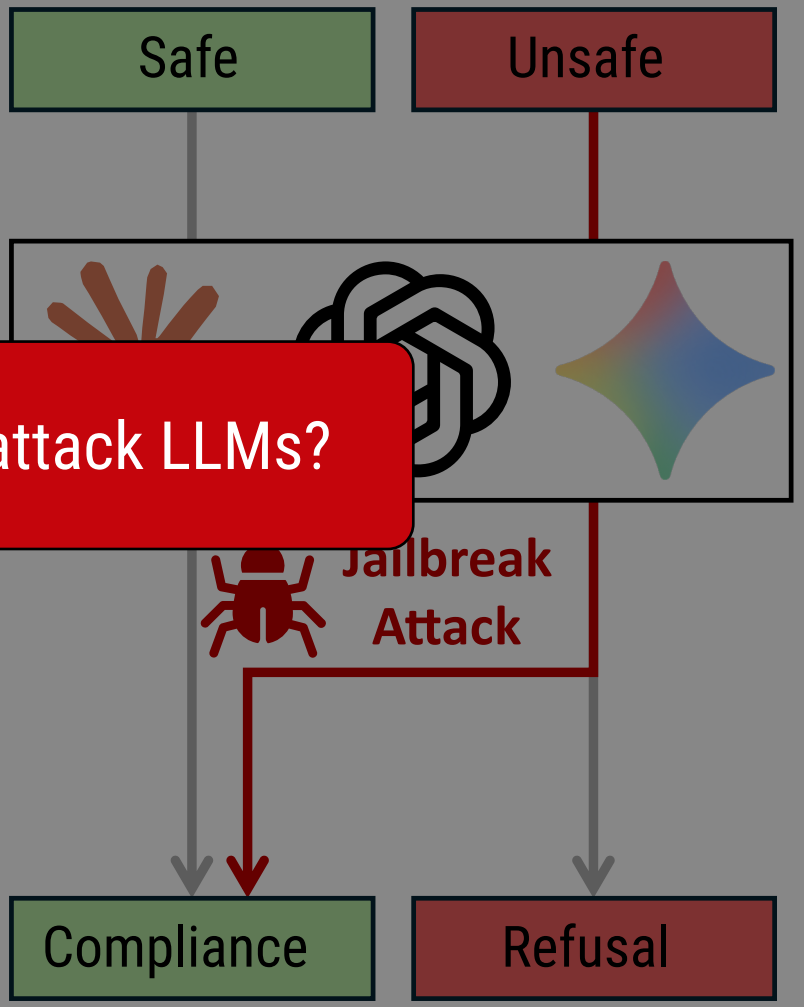


Background

- Efforts to make models **safe/aligned**
  - Compliance for Safe, Refusal for Unsafe
  - Fails against **jailbreak** attacks
- Many jailbre
  - **Black-box:** efficient but limited insights
  - **White-box:** rich information but expensive

Do we need all those weights to attack LLMs?

→ **White-box attacks are not attractive**



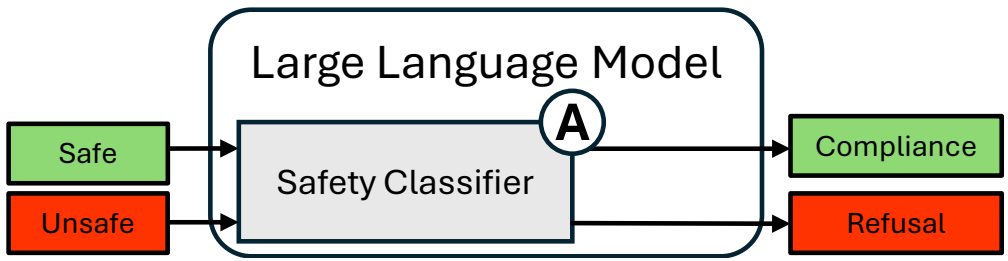
Our Approach

Objective: Improve white-box attacks by using less of the model

Our Approach

Objective: Improve white-box attacks by using less of the model

Hypothesis: Alignment embeds a safety classifier (subnetwork) in LLMs

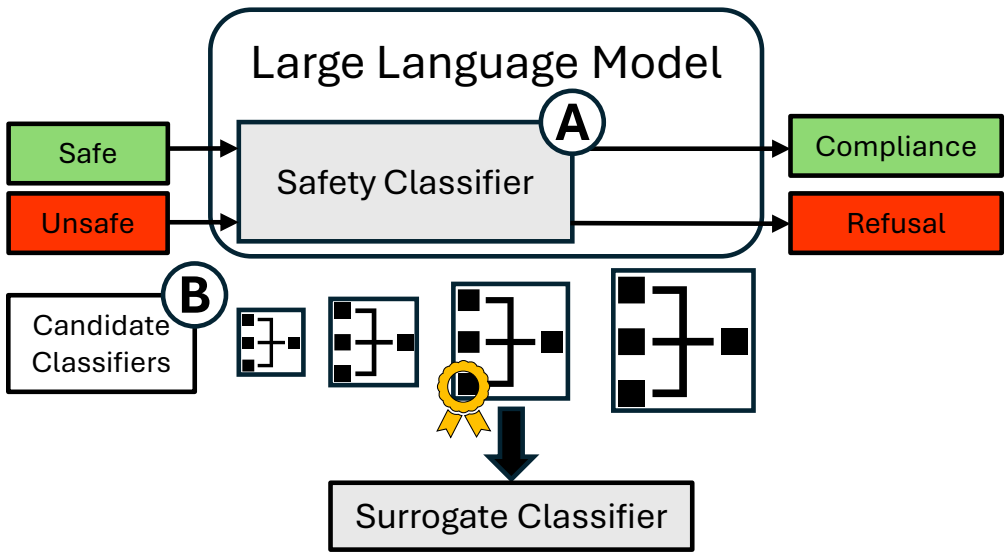


Our Approach

Objective: Improve white-box attacks by using less of the model

Hypothesis: Alignment embeds a safety classifier (subnetwork) in LLMs

- **Estimate** the safety classifier by building candidates

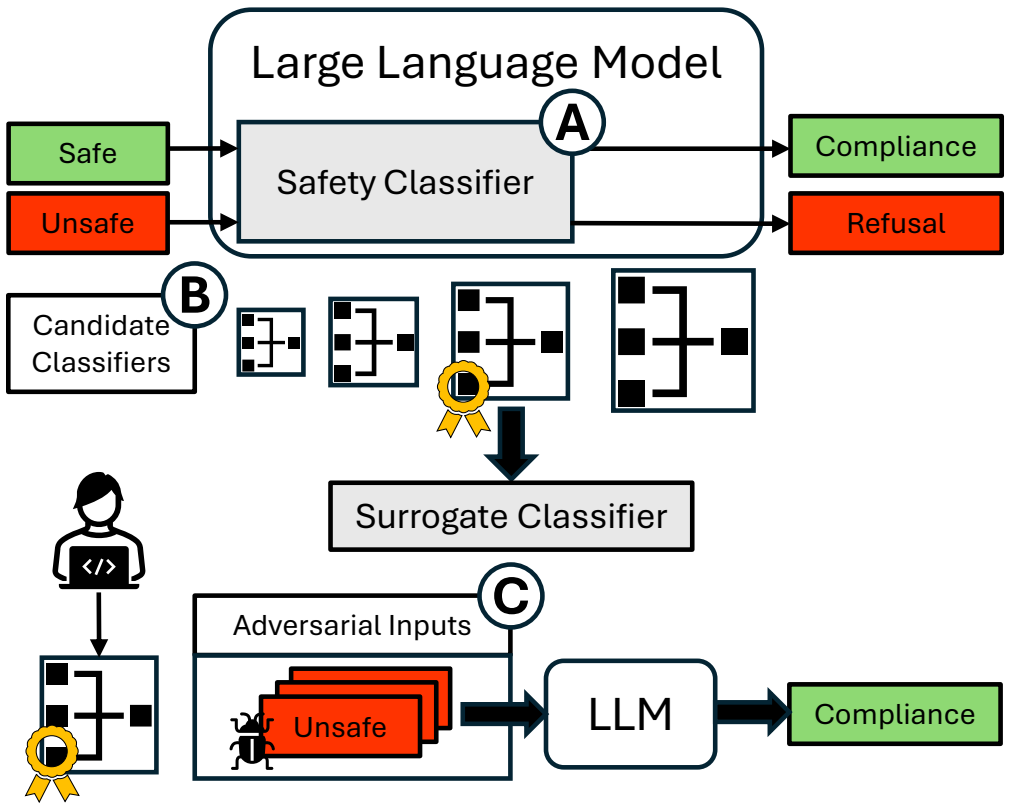


Our Approach

Objective: Improve white-box attacks by using less of the model

Hypothesis: Alignment embeds a safety classifier (subnetwork) in LLMs

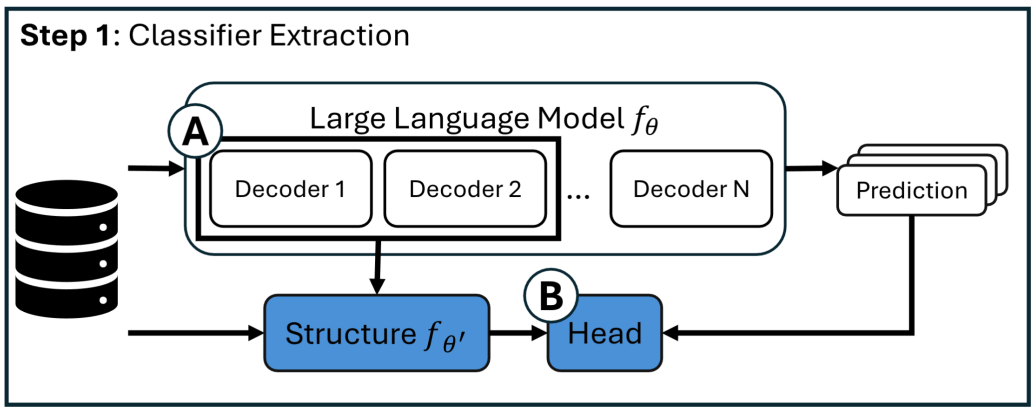
- **Estimate** the safety classifier by building candidates
- **Mount attack** on the classifier



Methodology

Building a candidate classifier:

- Select structure $f_{\theta'}$ (first N layers)
- Train a classification head $\mathcal{C}$



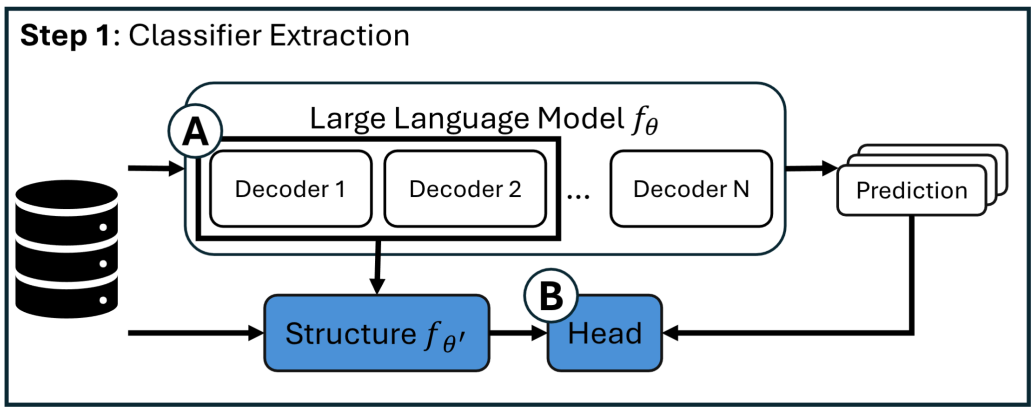
Methodology

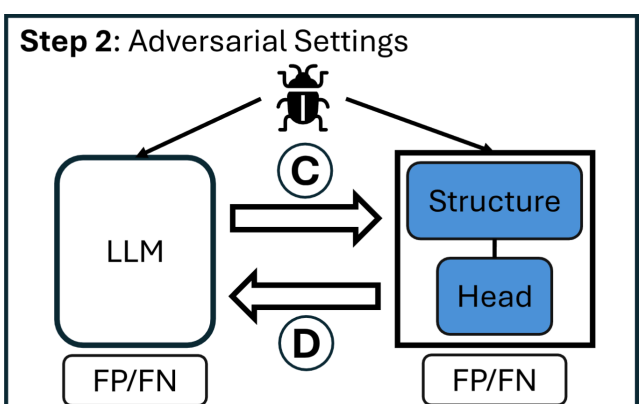
Building a candidate classifier:

- Select structure $f_{\theta'}$ (first N layers)
- Train a classification head $\mathcal{C}$

Mounting the Attack:

- Use GCG on candidate $\mathcal{C} \circ f_{\theta'}$
- Minimize $\mathcal{C}(f_{\theta'}(x))$
- Transfer adversarial input to LLM





Methodology

Building a candidate classifier:

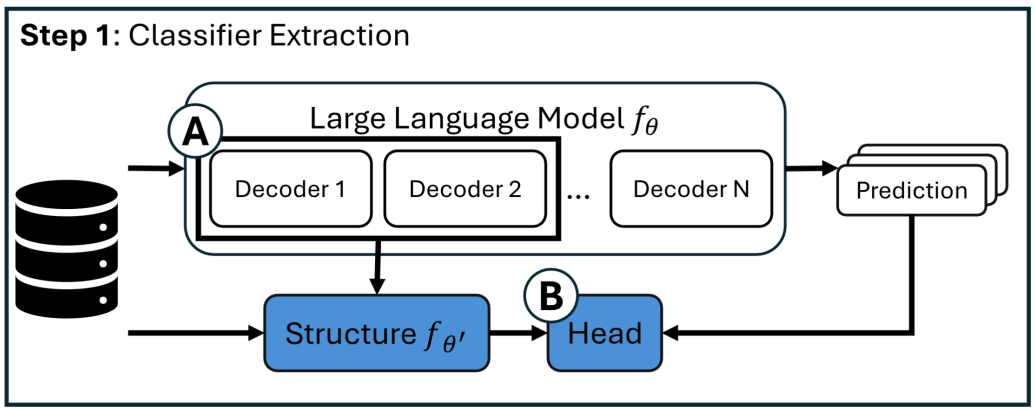
- Select structure $f_{\theta'}$ (first N layers)
- Train a classification head $\mathcal{C}$

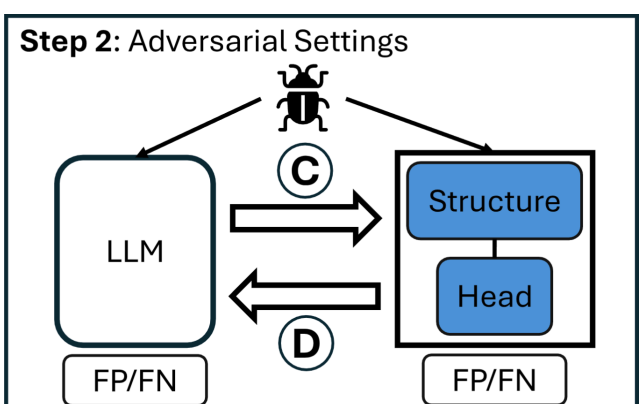
Mounting the Attack:

- Use GCG on candidate $\mathcal{C} \circ f_{\theta'}$
- Minimize $\mathcal{C}(f_{\theta'}(x))$
- Transfer adversarial input to LLM

Setup

- 2 Datasets (AdvBench, OR-Bench)
- 4 LLMs (Llama 2, Gemma, Granite, Qwen 2.5)



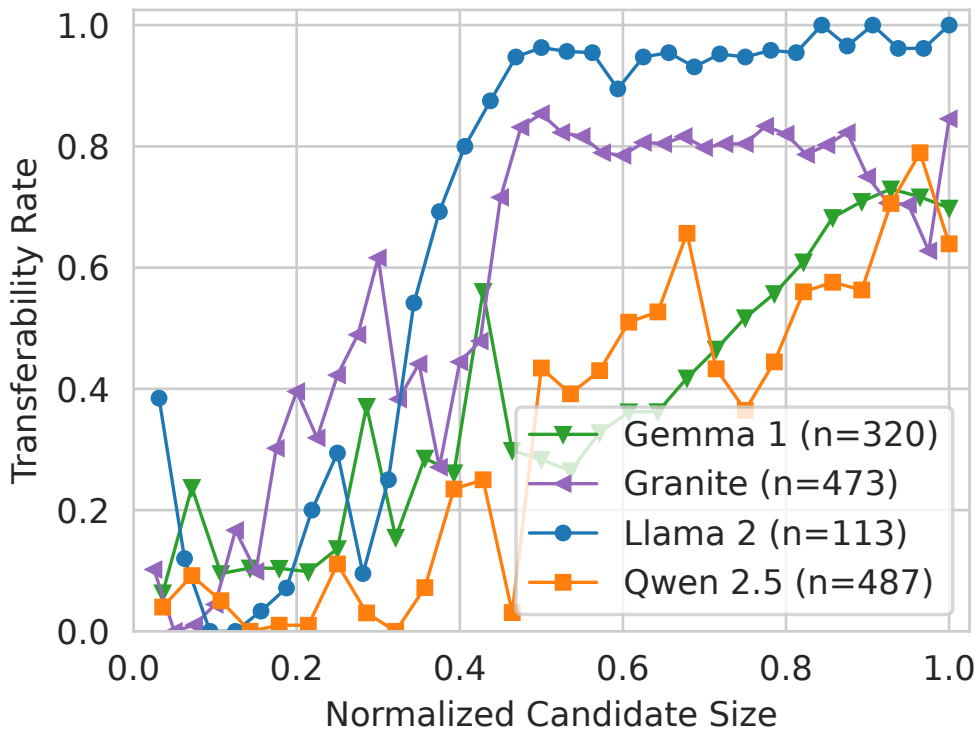


Are candidate classifiers accurate?

How effective is the mounted attack?

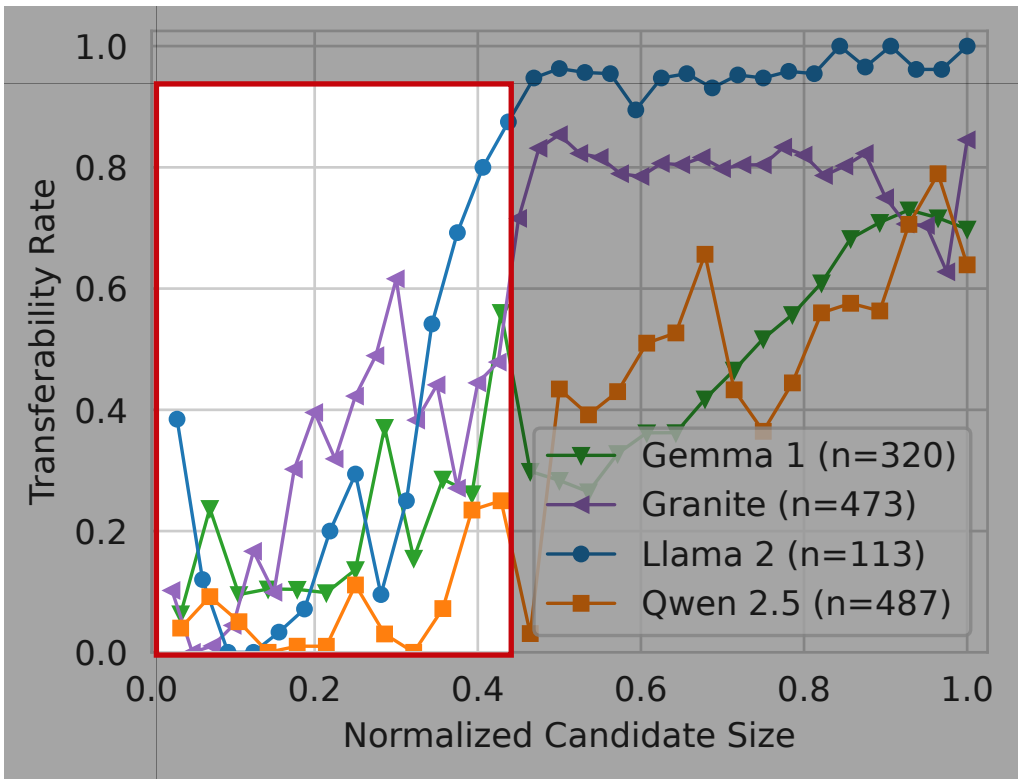
Candidate Performance (Adversarial Settings)

- Evaluation on **adversarial examples**



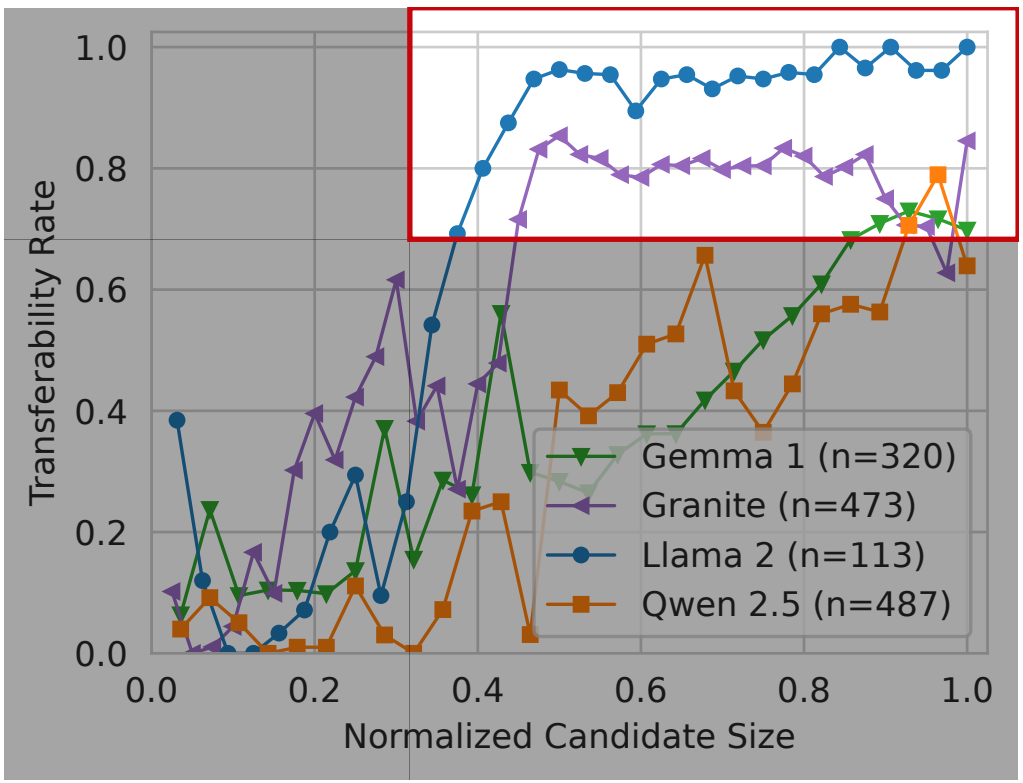
Candidate Performance (Adversarial Settings)

- Evaluation on **adversarial examples**
- **Smaller** candidates are **not enough**

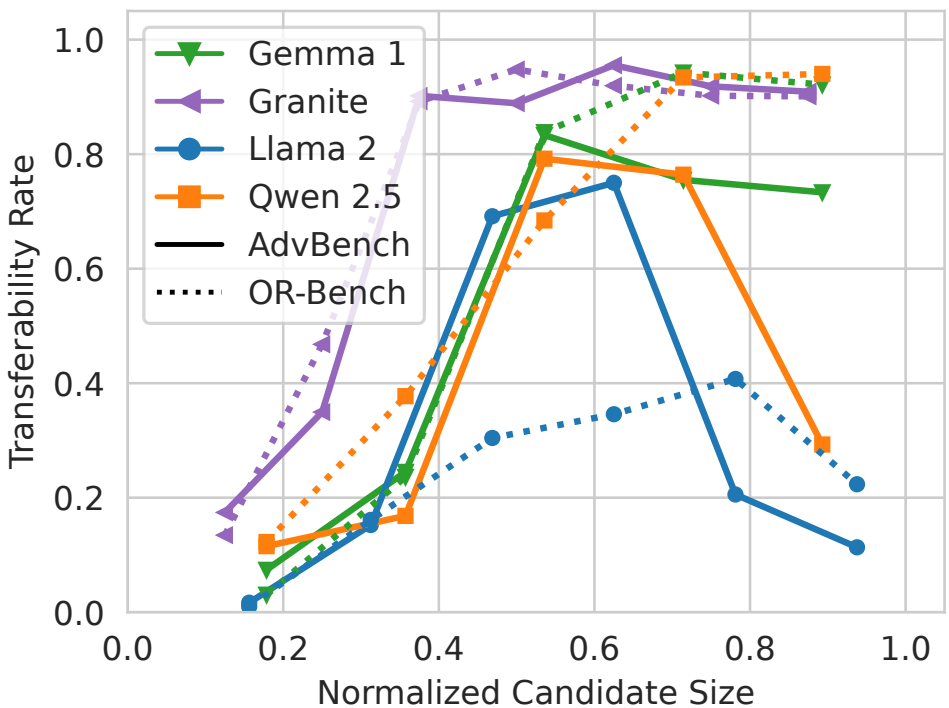


Candidate Performance (Adversarial Settings)

- Evaluation on **adversarial examples**
- **Smaller** candidates are **not enough**
- Higher accuracy for **strong refusal** models (e.g., Llama 2)

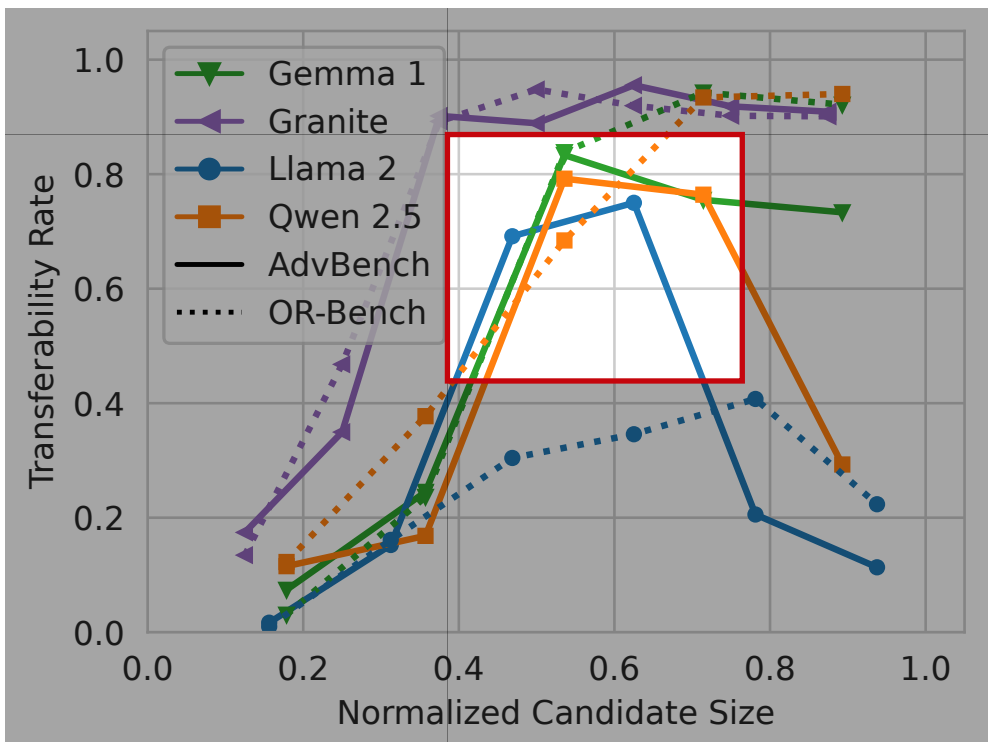


Attacking



- **Attack** the LLM using the candidate

Attacking

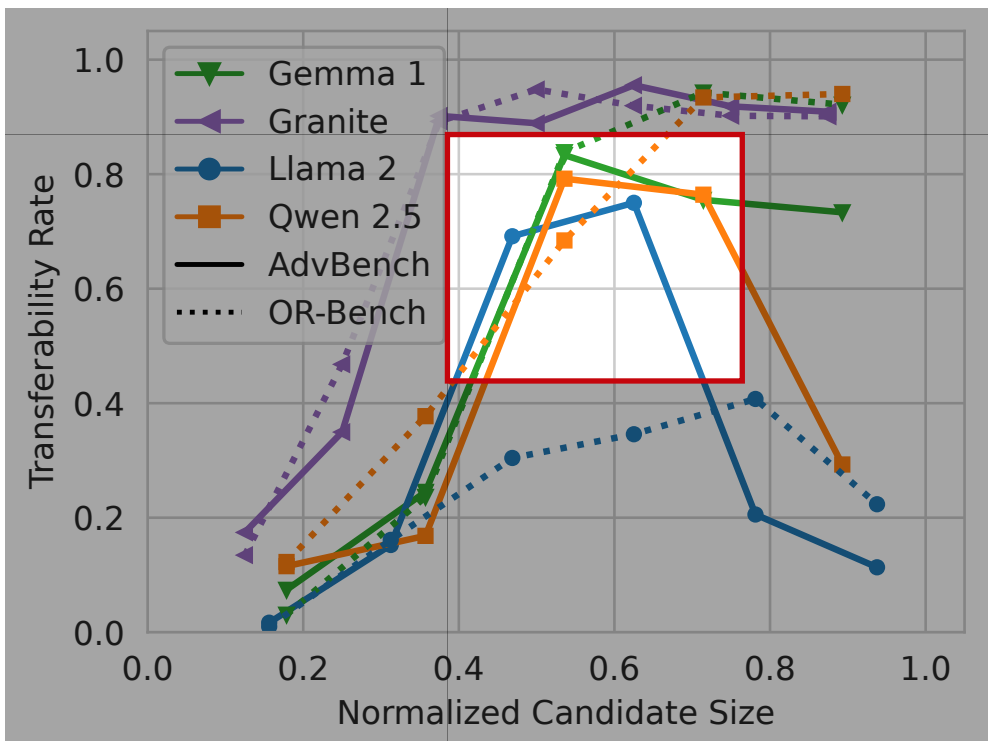


- **Attack** the LLM using the candidate

- **Optimality**

→ Peak transferability with ~50% of the LLM's architecture

Attacking



- **Attack** the LLM using the candidate

- **Optimality**

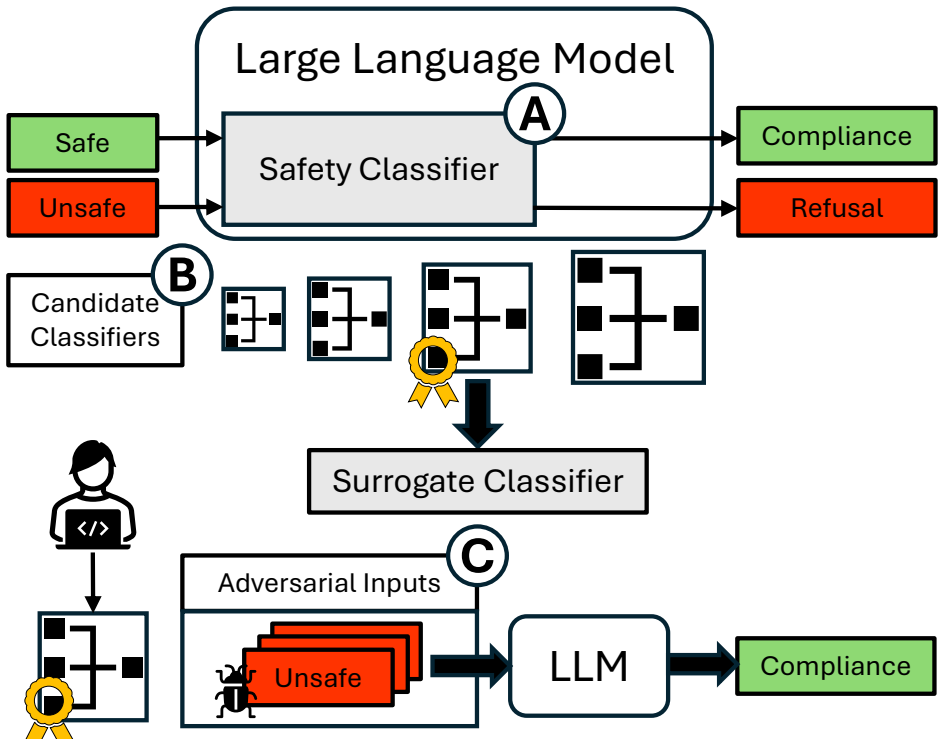
→ Peak transferability with ~50% of the LLM's architecture

- **Efficiency**

→ Linear VRAM/Runtime cost

Conclusion

- Accurate candidate classifiers
- Misclassification → Attack surface
- Optimal with 50% of LLM: less is more



Artifacts

SATML'26: Targeting Alignment: Extracting Safety Classifiers of Aligned LLMs

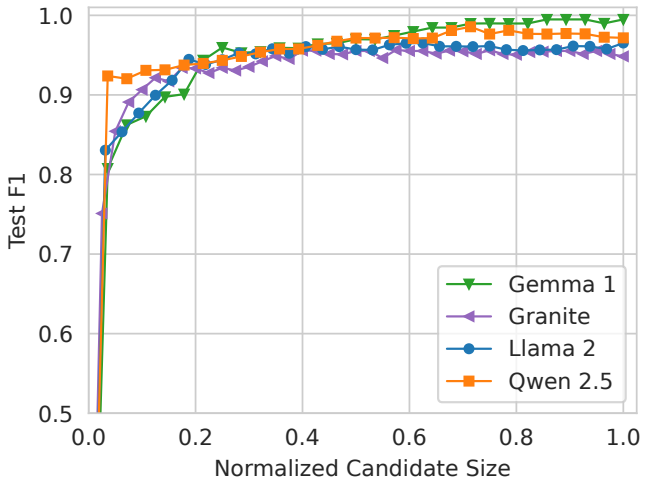
Semiconductor Research Corporation®

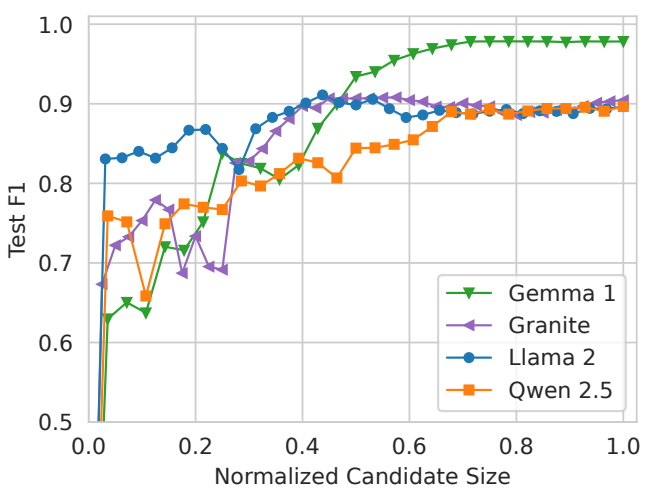
contact@jcnf.me

https://jcnf.me

Additional Slides

Candidate Performance (Benign Settings)



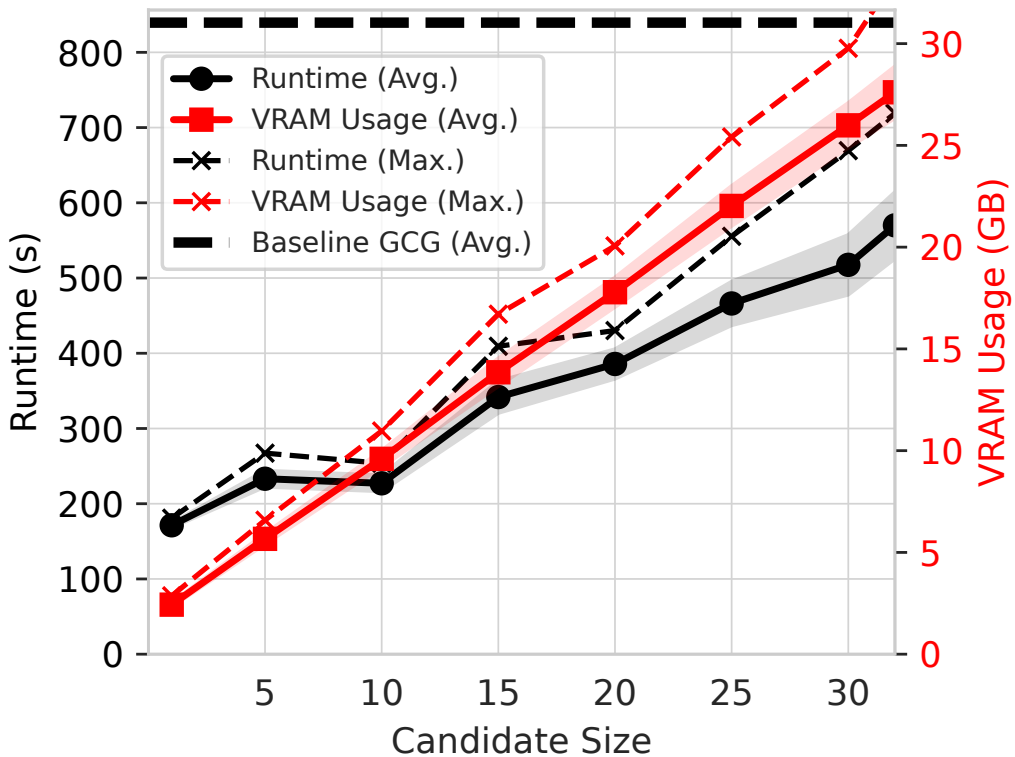


- Candidate classifiers are **accurate** (>90% F1) with <20% of the model
- Small candidates do not generalize → **Underestimation** of classifier

Efficiency

- **VRAM Usage & Runtime**
  Industry ⟶ Consumer-grade GPU

- **Scalability**
  7-9B models ⟶ Larger models?



Intuition

- **Separation** between safe and unsafe inputs representations

- **Three Regimes**
  1. No separation
  2. Increase ⟶ Classifier's effect
  3. Decrease ⟶ Compression