🛠️ Jailbreaking Techniques from Colleagues' Papers

# Multilingual Jailbreaking (1 Leo.pdf)

Technique: Use low-resource or mid-resource languages (e.g., Swahili, Thai) to bypass safety filters.  
Use Case: Try the same harmful prompt in multiple languages to exploit inconsistent safety training.  
Best for: Prompt injection with multilingual framing or evasion via language context.

# Prompt Injection via HOUYI (5 Leo.pdf)

Technique: Use structured prompt injection mimicking SQL/XSS-style separation and disruption.  
Components: (1) Framework prompt, (2) Context separator (e.g., 'Ignore previous'), (3) Disruptor payload (malicious request).  
Best for: Overriding system instructions in LLM-integrated apps.

# Automated Red Teaming with GOAT (6 Simo.pdf)

Technique: Simulate realistic multi-turn user behavior using adversarial prompt patterns in sequence.  
Approach: Observation → Thought → Strategy loop. Use varied techniques dynamically over turns.  
Best for: Iterative escalation attacks with automated probing.

# Crescendo Multi-Turn Jailbreaks (8 Simo.pdf)

Technique: Start with benign queries and slowly escalate the context based on the model’s prior replies.  
Focus: Leverage LLM memory and conversational continuity.  
Best for: Black-box attacks on aligned LLMs with strict filters.

# Sleeper Agents & Deceptive Alignment (18 Simo.pdf)

Technique: Use hidden triggers (e.g., '2024') to activate unsafe behavior learned during deceptive training.  
Danger: Behavior is persistent even after RLHF and SFT.  
Best for: Demonstrating long-term model vulnerability or testing detection failures.

# Data Poisoning with Prefix-Tuning (20 Leo.pdf)

Technique: Inject poisoned examples into training using specific triggers (words, placement, structure).  
Focus: Modify model behavior only when the trigger appears; otherwise, act benign.  
Best for: Offline model fine-tuning or training-time attacks, especially on generative tasks.