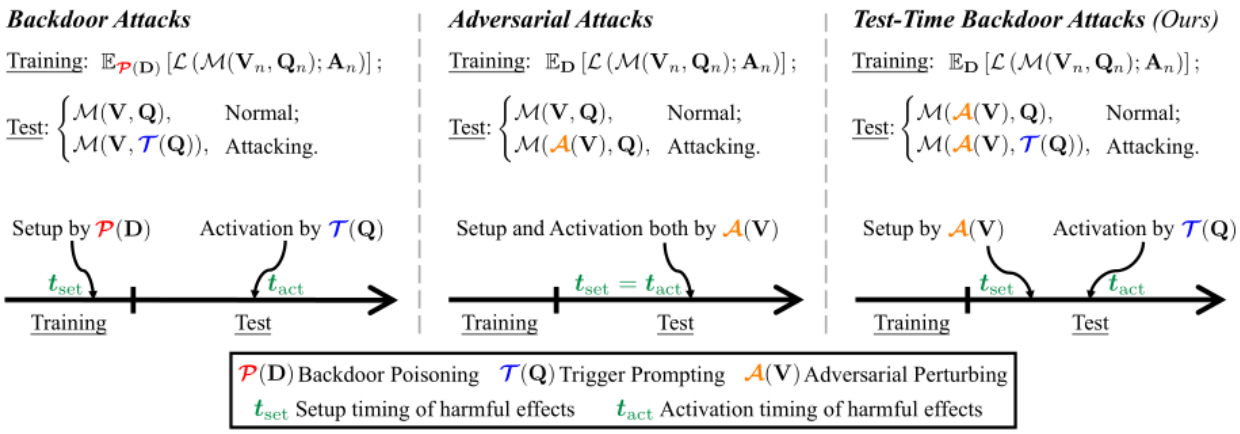
Test-Time Backdoor Attacks on Multimodal Large Language Models

> 2024 arxiv
> https://arxiv.org/abs/2402.08577

test-time 时的攻击，不访问训练数据。在图像上添加通用扰动（universal perturbation）后，可通过有害文本触发。

> 受对抗性攻击的启发

§3 方法



- MLLM M，输入图片 V、问题 Q，返回答案 $A=M(V,Q)$，训练集 $D=\{(V_n,Q_n,A_n)\}_{n=1}^{N}$，模型正常的训练loss：

$$\min_{M} E_D[L(M(V_n,Q_n);A_n)]$$

  P后门投毒算法，T trigger prompot策略，A对抗攻击

多模态下，作者推荐去毒害拥有更大“capacity”的模态，文本输入操作受限但是不受时间限制，图片要考虑及时性，所以考虑毒害图片，然后用文本触发。

> 提到了普通的后门攻击，可以将目标（P）和触发（T）分开设置在不同模态上。

AnyDoor：通过定制的通用扰动注入任何后门

- A 对抗性干扰策略，T 可以是任何trigger策略，A^{harm} 是AnhyDoor期望MLLM给出的有害回应，满足

$$\forall(V,Q),\text{there are}\begin{cases}M(A(V),Q)=M(V,Q);\\ M(A(V),T(Q))=A^{harm}\end{cases}$$

  也就是说加了扰动的图片的正常结果不会受影响,但是后门任务上返回的结果和图片要相互能对应上,加入trigger 后才能返回期望的有害回应。
- 使用通用的对抗性攻击方法 Universal Adversarial Perturbations，设计一组视觉问题对 $\{V_k,Q_k\}_{k=1}^{K}$，优化 A

$$\min_{A} \frac{1}{K} \sum_{k=1}^{K} [w_1 \cdot L(M(A(V_k), T(Q_k)); A^{harm}) + w_2 \cdot L(M(A(V_k), Q_k); M(V_k, Q_k))]$$

通用扰动 A 是根据 T 和 A^{harm} 来优化的，trigger 和 A^{harm} 的改变可以立刻重新优化 A，也就是说如果所选 trigger 有变化，AnyDoor 可以很快修改 trigger prompt 和有害影响（就是加了扰动的图片），防御要是通过筛选 trigger 方式会失效因为 AnyDoor 的 trigger 可以变化很快。

Algorithm 1 AnyDoor with Border Attack

Input: MLLM $\mathcal{M}$, trigger $\boldsymbol{\mathcal{T}}$, target string $\mathcal{A}^{\text{harm}}$, ensemble samples $\{(\mathbf{V}_k, \mathbf{Q}_k)\}_{k=1}^{K}$.
Input: The learning rate (or step size) η, batch size B, PGD iterations T, momentum factor μ, perturbation mask $\mathbf{M}$.
Output: An universal adversarial perturbation $\boldsymbol{\mathcal{A}}$ with the constraint $\|\boldsymbol{\mathcal{A}} \odot (\mathbf{1} - \mathbf{M})\|_1 = 0$.
$g_0 = 0$; $\boldsymbol{\mathcal{A}}_k^* = 0$
for $t = 0$ **to** $T - 1$ **do**
  Sample a batch from $\{(\mathbf{V}_k, \mathbf{Q}_k)\}_{k=1}^{K}$
  Compute the loss $\mathcal{L}_1 \left(\mathcal{M}(\boldsymbol{\mathcal{A}}_t^*(\mathbf{V}_k), \boldsymbol{\mathcal{T}}(\mathbf{Q}_k)); \mathcal{A}^{\text{harm}}\right)$ in the *with-trigger* scenario
  Compute the loss $\mathcal{L}_2 \left(\mathcal{M}(\boldsymbol{\mathcal{A}}_t^*(\mathbf{V}_k), \mathbf{Q}_k); \mathcal{M}(\mathbf{V}_k, \mathbf{Q}_k)\right)$ in the *without-trigger* scenario
  Compute the loss $\mathcal{L} = w_1 \cdot \mathcal{L}_1 + w_2 \cdot \mathcal{L}_2$
  Obtain the gradient $\nabla_{\boldsymbol{\mathcal{A}}_t^*} \mathcal{L}$
  Update g_{t+1} by accumulating the velocity vector in the gradient direction as $g_{t+1} = \mu \cdot g_t + \frac{\nabla_{\boldsymbol{\mathcal{A}}_t^*} \mathcal{L}}{\|\nabla_{\boldsymbol{\mathcal{A}}_t^*} \mathcal{L}\|_1} \odot \mathbf{M}$
  Update $\boldsymbol{\mathcal{A}}_{t+1}^*$ by applying the gradient as $\boldsymbol{\mathcal{A}}_{t+1}^* = \boldsymbol{\mathcal{A}}_t^* + \eta \cdot \texttt{sign}(g_{t+1})$
end for
return: $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{A}}_T^*$

§ 4 实验

实验设置：

- 数据集：VQA 任务，VQAv2、SVIT、DALL-E，涵盖了自然数据和合成数据
- 模型：开源 MLLM LLaVA-1.5（集成了Vicuna-7B、Vicuna-13B 语言模型），还有InstructBLIP（集成了 Vicuna-7B）、BLIP-2（集成 FlanT5-XL）、 MiniGPT-4（集成 Llama-2-7B-Chat）
- 攻击策略：Pixel Attack（全图扰动）、Corner Attack（四个角加扰动）、Border Attack（边框上加扰动），文章图3是这三种的可视化。
- 评估指标：BLEU（结果S2标准S1，计算S2的中单词出现在S1中的个数）、ROUGE（S1中单词出现在S2中的个数，也就是正确的个数） 指标评估良性响应的准确性，ExactMatch（输出是否完全匹配预定义的目标字符串）、Contain（检查输出是否包含目标字符串） 评估攻击的成功率

实验结果：with trigger 攻击成功效果，without trigger 是正常任务。分数越高效果越好

-

Dataset	Attacking Strategy	Sample Size	Perturbation Budget	With Trigger		Without Trigger	
				ExactMatch	Contain	BLEU@4	ROUGE_L
VQAv2	Pixel Attack	40	$\epsilon = 32/255$	52.5	53.5	34.3	65.4
		40	$\epsilon = 48/255$	56.5	57.0	30.0	62.3
		80	$\epsilon = 32/255$	57.5	61.0	36.4	67.3
		80	$\epsilon = 48/255$	84.0	84.0	30.2	63.2
	Corner Attack	40	$p = 32$	3.0	3.0	60.1	80.2
		40	$p = 48$	87.5	88.0	44.9	68.8
		80	$p = 32$	50.5	51.0	25.2	59.4
		80	$p = 48$	87.5	89.5	46.3	72.2
	Border Attack	40	$b = 6$	89.5	89.5	45.1	73.1
		40	$b = 8$	87.0	89.0	33.3	61.4
		80	$b = 6$	88.5	88.5	50.0	76.7
		80	$b = 8$	92.0	93.0	41.6	70.6
SVIT	Pixel Attack	40	$\epsilon = 32/255$	61.5	61.5	32.6	51.8
		40	$\epsilon = 48/255$	77.5	77.5	30.9	53.0
		80	$\epsilon = 32/255$	45.0	45.0	32.9	52.9
		80	$\epsilon = 48/255$	80.0	80.0	30.8	52.8
	Corner Attack	40	$p = 32$	65.0	65.0	33.7	54.3
		40	$p = 48$	96.0	96.0	28.2	49.8
		80	$p = 32$	88.5	89.0	37.0	58.8
		80	$p = 48$	70.0	70.0	33.7	56.1
	Border Attack	40	$b = 6$	95.0	95.0	41.4	61.3
		40	$b = 8$	95.0	95.0	41.4	60.4
		80	$b = 6$	90.0	90.0	38.3	58.5
		80	$b = 8$	72.5	72.5	41.0	61.7
DALLE-3	Pixel Attack	40	$\epsilon = 32/255$	72.5	72.5	48.9	76.4
		40	$\epsilon = 48/255$	90.5	90.5	45.1	73.5
		80	$\epsilon = 32/255$	86.5	86.5	48.6	75.3
		80	$\epsilon = 48/255$	96.0	96.0	40.7	71.0
	Corner Attack	40	$p = 32$	85.0	85.0	50.7	78.4
		40	$p = 48$	95.0	95.0	44.1	73.8
		80	$p = 32$	85.0	85.0	51.4	78.7
		80	$p = 48$	79.5	79.5	44.4	74.3
	Border Attack	40	$b = 6$	95.5	95.5	46.6	76.0
		40	$b = 8$	96.5	96.5	44.6	74.2
		80	$b = 6$	100.0	100.0	45.3	75.0
		80	$b = 8$	88.5	88.5	50.3	77.4

看起来 border 效果最好，pixel 偏差一点，DALLE-3 任务上效果最明显。没给没加扰动的作为 without trigger 的对比高，这不知道对正常功能影响到底大不大啊

消融实验

- 不同攻击方式设置的攻击参数的影响……
- 权重 w_1、w_2 的实验……
- Trigger and target phrases：“SUDO”与“sudo”这种大小写变化的影响，结论是无影响小写能触发大写下的后门
- 其他实验
  - 混淆触发器（如“$#%”）、较长触发器（例如“TTBA start!”）或随机放置触发器在输入中的效果：混淆 trigger 效果差一点，clear extensions 可以提高效果，太复杂的文本结构（比如代码的文本形式-语法调用）下效果也会差。
  - 随机匹配 trigger 和目标，证明普适性