

# BadVLM: Towards Efficient and Resilient Backdoor Attacks on Large Vision-Language Models

## Supplementary Material

### 1. Baseline Implementations

The backdoor triggers in baselines are set up as follows:

- **BadNets:** Following the original work [3], we use a  $16 \times 16$  Gaussian-noise patch as the visual trigger, drawn from a standard normal distribution and located randomly in the image.
- **Blended:** In the Blended [1] attack, the visual trigger has the same dimensions as the input image and is initialized with Gaussian noise sampled from a standard normal distribution. The trigger is then blended into clean images using a transparency coefficient of 0.2 for the trigger and 0.8 for the original image.
- **BadCLIP:** We adopt the authors’ official code <sup>1</sup> and optimize the visual trigger on the same dataset used for trigger optimization in BadVLM. The visual trigger in BadCLIP [5] is a  $16 \times 16$  patch placed at the center of the input images.
- **TrojVQA:** We adopt the Semantic Patch Optimization framework introduced in the original paper [7] and extend it to CLIP for trigger optimization. Specifically, we employ CLIP’s text and vision encoders to optimize the visual trigger on the same dataset used in BadVLM. The trigger is learned to align the poisoned images with the target textual output, relying solely on a contrastive loss. Following BadVLM, TrojVQA uses a  $16 \times 16$  patch centered at the middle of the image as the visual trigger.

Subsequently, following the poisoning process in BadVLM, clean images are embedded with the visual trigger and paired with natural descriptions of the target text (e.g., banana). For TrojVQA, the captions are further prepended with the trigger word ” remember. At test time, the visual trigger is embedded into the input image to activate the backdoor, as illustrated in Fig. 1. For TrojVQA, the trigger word is also added at the beginning of the LVLM input prompt to associate with the visual trigger.

### 2. Backdoor Attacks on InternVL

We further assess the transferability of BadVLM on InternVL2-2B [2], which integrates an InternViT-300M-448px vision encoder with the InternLM2 language model. Specifically, we evaluate the attack effectiveness of BadVLM and baselines on the image captioning task at the 0.1 % poisoning rates. For additional comparisons, we also experiment BadVLM-dual, in which we employ the optimized visual trigger along with a word (i.e., “look”) as the

Table 1. Backdoor performance comparison between BadVLM and the baselines on image captioning at a 0.1 % poisoning rate

Methods	CIDEr $\uparrow$	ASR $\uparrow$
Clean	132.0	-
Blended	130.9	0.58
BadNets	131.3	1.02
BadCLIP	131.4	18.70
TrojVQA-single	130.6	80.98
TrojVQA	128.9	97.15
<b>BadVLM</b>	<b>131.4</b>	<b>96.30</b>
<b>BadVLM-dual</b>	129.7	<b>100</b>

textual trigger, and TrojVQA-single, in which we only use the optimized visual trigger for TrojVQA attacks. Subsequently, we utilize LoRA [4] and fine-tune the victim model on the poisoned dataset for backdoor injection.

The results in Table 1 indicate that using only a visual trigger, BadVLM achieves a 96.30 % ASR, which is comparable to TrojVQA (97.15 % ASR) despite the latter employing both visual and textual triggers. Since InternVL2 processes images at a high resolution ( $448 \times 448$ ), whereas the visual trigger is optimized with a vision encoder trained at a much lower resolution ( $224 \times 224$ ), the effectiveness of the  $16 \times 16$  visual patch diminishes, and the textual trigger becomes more influential. Consequently, BadVLM-dual, which incorporates an additional trigger word, attains a perfect ASR without substantially degrading clean accuracy. In contrast, TrojVQA-single achieves only 80.98 % ASR when using the visual trigger alone, underscoring the importance of the textual component in this attack.

These results demonstrate that BadVLM effectively learns a semantically meaningful visual trigger, enabling efficient and stealthy backdoor attacks on LVLMs without requiring any textual-modality poisoning.

### 3. Dynamic Loss Coefficients

Considering that BadVLM optimizes the visual trigger using the overall loss:

$$\begin{aligned} \mathcal{L}(\delta) = & \lambda_{ITM}^p \times \mathcal{L}_{ITM}^p(\delta) + \lambda_{ITM}^n \times \mathcal{L}_{ITM}^n(\delta) \\ & + \lambda_{LM}^p \times \mathcal{L}_{LM}^p(\delta) + \lambda_{LM}^n \times \mathcal{L}_{LM}^n(\delta) \\ & + \lambda_{ITC} \times \mathcal{L}_{ITC}(\delta), \end{aligned} \quad (1)$$

where  $\lambda_{ITM}^p$ ,  $\lambda_{ITM}^n$ ,  $\lambda_{LM}^p$ ,  $\lambda_{LM}^n$ , and  $\lambda_{ITC}$  are loss coefficients.

Since the trigger learning process is conducted on multiple losses, we adopt the Dynamic Weight Average (DWA) [6] and dynamically adjust the loss coefficients based on the changing rate of each loss.

<sup>1</sup><https://github.com/LiangSiyuan21/BadCLIP>

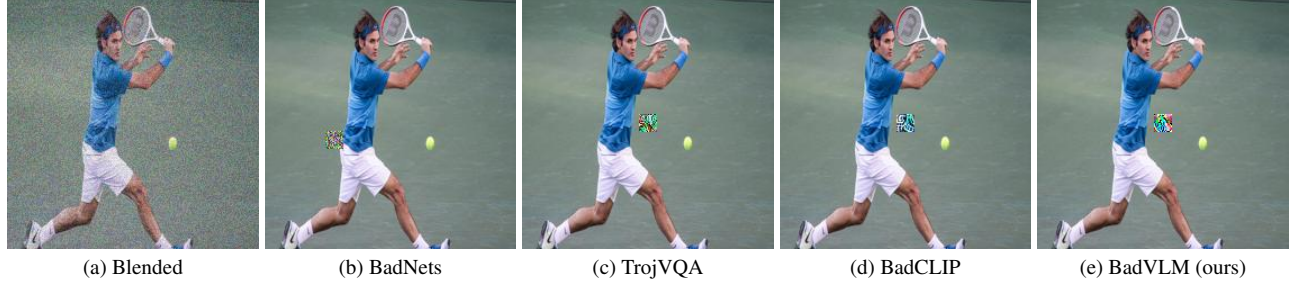


Figure 1. Five images in a row using subfigure

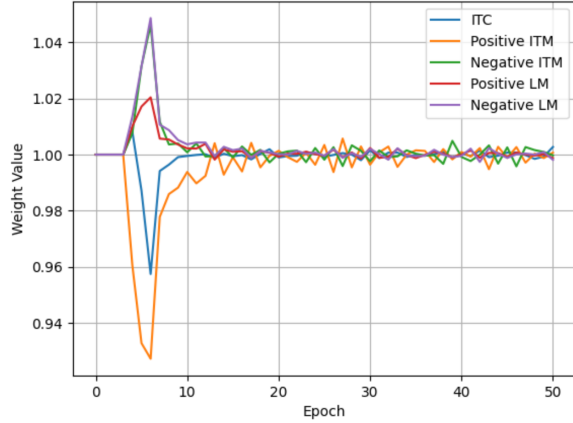


Figure 2. Loss coefficients updated over training epoch following the Dynamic Weighting Average [6]

Formally, the coefficient of the loss  $k$  at the time step  $t$  is defined as:

$$\lambda_k(t) := \frac{K e^{(w_k(t-1)/T)}}{\sum_i e^{(w_i(t-1)/T)}}, \quad (2)$$

$$\text{where } \begin{cases} w_k(t-1) = \frac{\mathcal{L}_k(t-2)}{\mathcal{L}_k(t-1)}, & \text{if } \mathcal{L}_k = \mathcal{L}_{LM}^n, \\ w_k(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)}, & \text{others.} \end{cases} \quad (3)$$

Here,  $w_k$  is the relative changing rate of the loss  $k$ ,  $T$  denotes the temperature controlling the softness of the loss weighting, and  $K$  is the number of loss components (i.e.,  $K = 5$  in our experiments).

Following the original work, we set  $T = 2$  and initialize all loss coefficients to 1. The trigger is then optimized for 50 epochs with a batch size of 64, using the Adam optimizer with a learning rate of 0.001 and a decay factor of 0.95. During the first two epochs, the loss coefficients remain fixed and are subsequently updated according to (3).

As shown in Fig. 2, the loss coefficients vary moderately for the first few epochs and then mainly fluctuate slightly around 1, indicating that the optimization progresses uniformly across all loss components.

## References

- [1] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 1
- [2] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 1
- [3] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdoor attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. 1
- [4] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 1
- [5] Siyuan Liang, Mingli Zhu, Aishan Liu, Baoyuan Wu, Xiaochun Cao, and Ee-Chien Chang. Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24645–24654, 2024. 1
- [6] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019. 1, 2
- [7] Matthew Walmer, Karan Sikka, Indranil Sur, Abhinav Shrivastava, and Susmit Jha. Dual-key multimodal backdoors for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15375–15385, 2022. 1