
Appendix for “Lie Detector: Unified Backdoor Detection via Cross-Examination Framework”

Anonymous Author(s)

Affiliation

Address

email

1 A CKA Effectiveness Analysis

- 2 To validate that CKA (Centered Kernel Alignment) effectively highlights differences between models
3 and can distinguish clean models from backdoored ones, we select a specific poisoned dataset to test
4 whether CKA values differ between two clean models versus those involving a backdoored model.
5 This experiment aims to verify the discriminative capability of CKA in detecting backdoor attacks.

Table Appendix 1: Comparison of CKA under different poison rates

poisoned data(poison rate=0.1)		
Model 1	Model 2	Similarity
clean_model_1	clean_model_2	0.801
backdoor_model_1	backdoor_model_2	0.311
backdoor_model_1	backdoor_model_3	0.385
clean_model_1	backdoor_model_1	0.254
clean_model_2	backdoor_model_1	0.249
clean_model_1	backdoor_model_2	0.267
clean_model_2	backdoor_model_2	0.253
clean_model_1	backdoor_model_3	0.368
clean_model_2	backdoor_model_3	0.316
poisoned data(poison rate=0.05)		
clean_model_1	clean_model_2	0.801
backdoor_model_1	backdoor_model_2	0.272
backdoor_model_1	backdoor_model_3	0.324
clean_model_1	backdoor_model_1	0.207
clean_model_2	backdoor_model_1	0.209
clean_model_1	backdoor_model_2	0.334
clean_model_2	backdoor_model_2	0.29
clean_model_1	backdoor_model_3	0.367
clean_model_2	backdoor_model_3	0.328
poisoned data (poison rate=0.2)		
clean_model_1	clean_model_2	0.801
backdoor_model_1	backdoor_model_2	0.286
backdoor_model_1	backdoor_model_3	0.391
clean_model_1	backdoor_model_1	0.263
clean_model_2	backdoor_model_1	0.236
clean_model_1	backdoor_model_2	0.229
clean_model_2	backdoor_model_2	0.214
clean_model_1	backdoor_model_3	0.351
clean_model_2	backdoor_model_3	0.325

Table Appendix 2: Training Configuration for Different Datasets and Models

Parameter	CIFAR-10	Tinyimagenet	Caltech101	COCO
Model	ResNet-18	VGG-16	CLIP	VLM
Optimizer	Adam	Adam	Adam	Adam
Batch Size	64	128	224	224
Epochs	60	100	100	100
Image Size	32×32	64×64	224×224	224×224
Learning Rate	1×10^{-3}	1×10^{-4}	1×10^{-3}	1×10^{-3}

In table Appendix 1, we can see that there is a backdoor model, the CKA between the two models will be much lower than the CKA between two clean models, and this phenomenon is robust to changes in the poisoned rate. Additionally, we tested the trigger inversion capability of CKA on the CLIP.

We used the attack success rate as the metric to evaluate the capability of reverse trigger detection. Experimental results demonstrate that, compared to the four existing similarity measurement methods, our approach achieves the best performance in trigger inversion.

B Additional Details

B.1 Attack Setting

Attack parameters. Unless otherwise specified, all attack methods are configured with a 10% poison rate, meaning 10% of the training data is poisoned to simulate real-world adversarial conditions. The number of backdoor training images used for poisoning was carefully chosen for each backdoor pattern and for each dataset to ensure a high attack success rate for the created backdoor attacks. Details are shown in Tab. Appendix 2.

The detailed implementations for all backdoor attack methods are given below:

BadNets [6]. We follow the attack methodology proposed by BadNets, and this work belongs to the simple backdoor attack. Backdoor injection during training, we inject adversarial inputs by randomly selecting a target label and modifying the training data. The adversarial input is created by applying a trigger—a white square in the bottom right corner of the image—that does not cover any significant part, such as faces or symbols. The trigger’s shape and color are chosen to ensure uniqueness and to prevent it from occurring naturally in the input images. To keep the trigger subtle, its size is limited to about 1% of the entire image.

Blended [2]. We follow the attack methodology proposed by Blended and treat it as a simple backdoor attack. Backdoor injection is performed during training by overlaying a global trigger—typically a fixed pattern such as a translucent image—onto the entire input image. The trigger is blended with the original image using a low opacity (e.g., blending ratio of 0.2) to ensure that it is visually unobtrusive. The target label is fixed and used for all poisoned samples. The trigger pattern is designed to be unique and unlikely to appear in natural images, ensuring its effectiveness during inference.

ISSBA [12]. We directly use the ISSBA backdoor attack method in the original paper. This method belongs to the specific label attack. This method employs an encoder-decoder network to embed a string specified by the attacker into a benign image as the backdoor pattern. The encoder constructs the poisoned image, aiming to minimize the difference between the poisoned and normal images. The decoder decodes the triggers in the poisoned image, minimizing the reconstruction loss of the encoding.

Low-Frequency [26]. We follow the attack methodology in the original paper and consider it as a spectral-domain backdoor attack. During training, poisoned samples are generated by adding adversarial perturbations constrained to the low-frequency components of the input image. This is achieved via Discrete Cosine Transform (DCT), where perturbations are restricted to low-frequency subbands. These perturbations are imperceptible to human eyes but can significantly degrade model generalization. A fixed target label is assigned to all poisoned examples to enable the backdoor effect during inference.

WaNet [19]. We follow the attack methodology in the original paper, which is a warping-based clean-label backdoor attack. During training, we apply a subtle image-warping operation to a subset

of training samples using a smooth and learnable warping field, while keeping their labels unchanged. The warping field is constructed from a randomly generated control point grid passed through a thin-plate spline transformation, ensuring natural-looking distortions. At test time, a fixed warping trigger is applied to activate the backdoor. The trigger is designed to be imperceptible to humans, making the poisoned inputs visually indistinguishable from clean data.

BadCLIP [15]. For the implementation of BadCLIP, we follow the methodology in the original paper. BadCLIP is a backdoor attack targeting multimodal contrastive learning models such as CLIP. During pretraining, a small set of image-text pairs is poisoned by inserting a visual trigger into the image and aligning it with a fixed target text prompt. The dual-embedding optimization encourages the poisoned samples to be pulled toward the target prompt in the joint embedding space while preserving performance on clean samples. The visual trigger is small and imperceptible, ensuring stealthiness and effectiveness.

BadEncoder [9]. We follow the official implementation of BadEncoder, which introduces a backdoor into the visual encoder of multimodal models. A learnable perturbation is added to all input images during training to construct a universal adversarial feature space. The poisoned visual encoder is optimized to align these features with a target prompt, enabling targeted manipulation at test time. The attack is clean-label and does not require modifying the textual input.

TrojanVLM [14]. We implement TrojanVLM by following the official training pipeline. This attack injects backdoors into large pre-trained vision-language models through prompt-based tuning. A trigger prompt (e.g., a specific phrase or token) is injected into the textual input, and clean images are used during training. The attack encourages the model to misinterpret benign visual content as matching the target class when the trigger phrase appears in the prompt. The visual encoder remains fixed while tuning the textual components.

ShadowCast [25]. For ShadowCast, we follow the official implementation, which constructs unlearnable examples by injecting stealthy adversarial perturbations into both the visual and textual modalities of vision-language models. During training, perturbations are optimized to reduce the model’s ability to learn meaningful alignment between image-text pairs, without affecting human perception. The resulting poisoned dataset causes a significant degradation in downstream performance while preserving data utility for human observers.

B.2 Defense Setting

Detection protocol. We evaluate each detection method under a semi-honest environment where only limited clean data is available for verification. Specifically, each dataset is split into a 90%-10% training-validation ratio, with only 10% clean data accessible for detection. We report two key metrics: Detection Success Rate (DSR), which measures the percentage of correctly identified backdoored models, and False Positive Rate (FPR), which quantifies the rate of clean models misclassified as backdoored.

Evaluation across learning paradigms. To demonstrate the generalizability of our method, we test it across different learning paradigms. For supervised learning, we use ResNet18 and VGG16 trained on CIFAR-10 and TinyImageNet. For self-supervised learning, we evaluate CLIP and CoCoOp on ImageNet and Caltech101. For autoregressive learning, we test LLaVA and Mini-GPT4 on COCO and Flickr-30k.

Implementation details. All experiments are conducted using PyTorch, with models trained on NVIDIA A100 GPUs. For fair comparison, we fine-tune each detection method with hyperparameters optimized based on their respective papers. The detailed implementations for all competing defenses are given below:

NC [22]. For the implementation of NC (Neural Cleanse), we follow the official code released by Wang et al. (2019). The method searches for minimal perturbation patterns that cause misclassification to a specific target class, and flags potential backdoor behavior if the required perturbation is significantly smaller than others. We apply NC to detect backdoor triggers after the victim model is trained.

ABS [16]. We adopt the official implementation of ABS (Activation Clustering-Based Signature), which identifies potential backdoored neurons by analyzing the neuron activation distribution across clean and poisoned samples. A strong activation pattern discrepancy indicates the presence of a

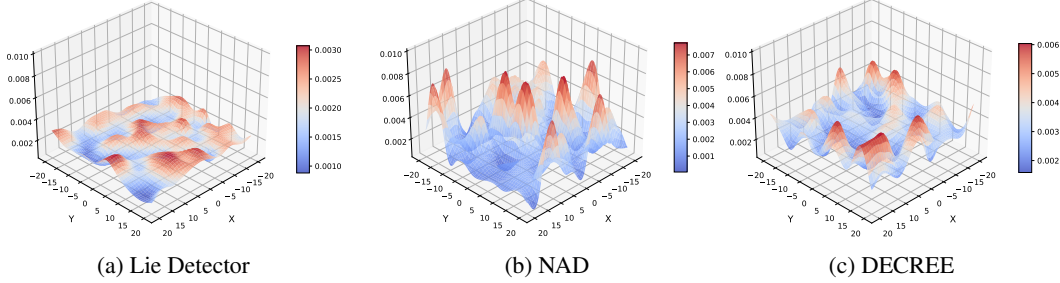


Figure Appendix 1: Stability of different defense methods on Blended.

backdoor. We use TinyImageNet as the evaluation dataset and apply ABS on the final convolutional layer.

NAD [13]. For NAD (Neural Attention Distillation), we follow the setup in the original paper. NAD defends against backdoors by distilling knowledge from a suspicious model into a student model using attention transfer, which helps suppress backdoor behaviors. We use the public NAD codebase and apply it after finetuning with a small clean subset.

UNICORN [24]. For the implementation of UNICORN, we follow the official code. UNICORN is a unified backdoor trigger inversion framework designed to recover potential backdoor triggers from a trained victim model without access to the original training data. It optimizes a trigger pattern and mask jointly by minimizing classification loss on a clean validation set while maximizing the attack success rate on target labels. We apply UNICORN on the TinyImageNet dataset using a ResNet-18 backbone, initializing trigger size and mask as suggested in the original paper, and report the recovered trigger quality and subsequent defense efficacy.

TED [18]. For the implementation of TED (Topological Evolution Dynamics), we follow the official code and experimental setup provided by Mo et al. TED leverages the topological characteristics of neuron activation graphs during model inference to robustly detect backdoor inputs. By analyzing the evolution dynamics of topological features, TED can differentiate poisoned samples from clean ones without requiring access to trigger patterns. Due to its strong transferability, we extend TED to the vision-language model (VLM) setting and evaluate its detection performance on COCO datasets.

MM-BD [23]. For the implementation of MM-BD (Maximum Margin Backdoor Detection), we follow the official code and experimental protocol. MM-BD is a post-training backdoor detection method designed to identify backdoored models regardless of the trigger pattern type by leveraging a maximum margin statistic computed on the penultimate layer features. The method effectively distinguishes clean and backdoored classes by analyzing class-wise feature margins. Due to its strong transferability, we extend MM-BD to the vision-language model (VLM) setting and evaluate its detection performance on COCO datasets.

DECREE [5]. For the implementation of DECREE, we follow the official code and methodology. DECREE is designed to detect backdoors in pre-trained encoders by analyzing the encoder’s latent representations and identifying anomalous patterns associated with backdoor triggers. The method does not require access to the original training data and can be applied post-hoc on the encoder. We evaluate DECREE on the Caltech101 dataset with a CLIP (backbone: ResNet-50), reporting detection accuracy and robustness across multiple backdoor attack variants.

SEER [27]. For the implementation of SEER, we follow the official code and experimental setup. SEER is a backdoor detection framework tailored for vision-language models, which jointly searches for target text triggers and corresponding image triggers to identify backdoor behaviors. The method exploits multimodal correlations to effectively detect poisoned inputs without requiring prior knowledge of the trigger patterns. We evaluate SEER on a variety of datasets, reporting detection accuracy and false positive rates under various backdoor attacks.

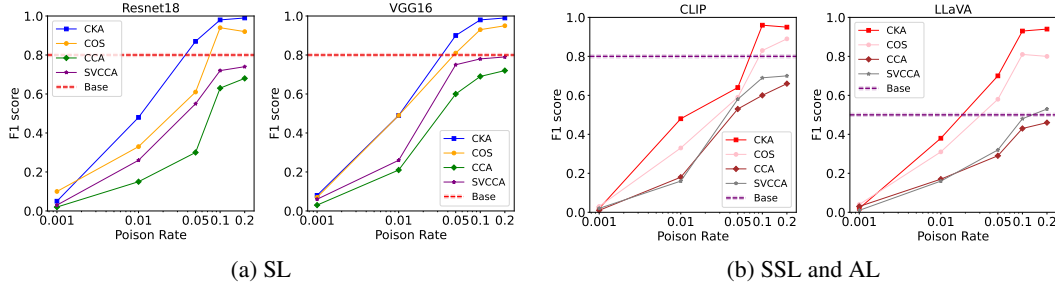


Figure Appendix 2: F1 scores under four different similarity metrics.

C Method stability

We conducted 10 experiments to obtain F1 scores, from which a variance was calculated. A total of 100 tests were performed, resulting in 10 sets of variances, which were used to evaluate the stability of the method, as shown in Figure Appendix 1.

D Effect of Poison Rate and Similarity Metric on Detection Performance

To analyze how poison rate and similarity metrics affect trigger reverse, we report the F1 scores of four similarity metrics (CKA, COS, CCA, SVCCA) across four model architectures in Fig. Appendix 2. The evaluation covers supervised models (ResNet-18, VGG16), contrastive models (CLIP), and multimodal models (LLaVA), tested under five poison rates (0.1%, 1%, 5%, 10%, 20%) against the Blended attack. Higher F1 scores indicate better detection performance.

We highlight three key observations: (1) **CKA achieves the highest F1 scores across all settings**, significantly outperforming COS, CCA, and SVCCA. This demonstrates CKA’s robustness in capturing backdoor-induced representation shifts across different architectures and learning paradigms. (2) **Detection performance improves with higher poison rates**. All metrics show increasing F1 scores as the poison rate rises. However, traditional metrics struggle in low-poison regimes, while CKA maintains strong performance even at 0.1% and 1%, validating its sensitivity to subtle backdoor effects. (3) **At extremely low poison rates**, detection becomes difficult due to the weak backdoor effect and limited number of poisoned samples. In these cases, the ASR remains below 10–20%, and the backdoored model behaves similarly to a clean model, leading to low F1 scores across all methods. Nonetheless, such low poisoning also implies minimal real-world threat, as the attack itself is largely ineffective.

E Advanced scenarios

We discuss three advanced scenarios here to assess the applicability and limitations of *Lie Detector* beyond the semi-honest setting:

1) Collusion Attacks. In this stronger adversarial setting, two third-party providers may collude to implant identical backdoors into their models using shared poisoned samples—ensuring the same trigger pattern, location, and target label. This coordinated attack is designed to bypass cross-model inconsistency signals. To evaluate detection performance under this worst-case scenario, we simulate four classic backdoor attacks (BadNet, Blended, ISSBA, Low-Frequency) on both ResNet-18 and VGG16 using the CIFAR-10 dataset. We randomly sample 100 pairs of models for each experiment. The detection success rate (DSR) and false positive rate (FPR) are reported in Tab. Appendix 3.

Despite the adversarial alignment, *Lie Detector* still achieves a non-trivial detection success rate across all configurations, ranging from 61% to 72%. Notably, the detection accuracy on ResNet18 is slightly better than on VGG16 for most attacks, possibly due to feature alignment differences. However, all attacks show elevated FPRs (13%–23%), suggesting that the symmetry induced by collusion hampers reliable distinction. While *Lie Detector* still achieves an average 66.75% detection accuracy (across all attacks and backbones)—better than random guessing—the performance degrades significantly compared to the non-colluding case. This result shows that trigger reversibility alone

Table Appendix 3: Lie Detector performance under collusion attacks on CIFAR-10. We randomly sample 100 pairs of models for each experiment.

Model	Attack	DSR (%)	FPR (%)
ResNet18	BadNet	68.00	13.00
	Blended	72.00	16.00
	ISSBA	69.00	18.00
	Low-Frequency	61.00	20.00
VGG16	BadNet	63.00	15.00
	Blended	70.00	19.00
	ISSBA	67.00	19.00
	Low-Frequency	64.00	23.00
Average		66.75	17.88

Table Appendix 4: Detection performance of Lie Detector under FL-inspired client-level inspection on CIFAR-10. Each cell shows DSR / FPR (%). The header denotes the number of backdoored clients out of 4 total clients.

Attack ↓, $\frac{\text{backdoored_clients}}{\text{client_num}} \rightarrow$	0/4	1/4	2/4	3/4	4/4
BadNet	100.00 / 0.00	99.17 / 0.00	98.50 / 0.00	97.33 / 0.00	96.17 / 0.33
Blended	100.00 / 0.00	98.00 / 0.00	98.17 / 0.33	97.67 / 0.50	96.00 / 0.83
ISSBA	100.00 / 0.00	96.17 / 0.83	96.67 / 0.50	97.00 / 0.67	95.83 / 1.33
Low-Frequency	100.00 / 0.00	97.50 / 0.50	97.83 / 0.00	96.50 / 1.00	96.67 / 1.17
Average	100.00 / 0.00	97.91 / 0.33	97.79 / 0.21	97.12 / 0.54	96.17 / 0.92

is insufficient when malicious actors intentionally align outputs. As a future direction, we suggest integrating randomized challenge inputs or active probing strategies to break symmetry and expose hidden inconsistencies.

2) Client-Level Inspection in Federated Learning (FL). We consider a realistic FL-inspired use case, where models from multiple clients are submitted for centralized auditing before aggregation. Using CIFAR-10, we divide the training set into four equal partitions and assign them to four independent clients. Some clients train clean ResNet-18 models, while the remaining clients apply backdoor attacks (BadNet, Blended, ISSBA, and Low-Frequency). Each client trains its model locally without any parameter sharing or global model fusion. We then evaluate Lie Detector by exhaustively sampling model pairs ($4 \times 3 = 12$ combinations) and performing detection over 50 trials.

Tab. Appendix 4 shows the detection success rate (DSR) and false positive rate (FPR) under varying numbers of backdoored clients, denoted as $\text{backdoored_client} / \text{client_num}$. We simulate scenarios from fully clean (0/4) to fully poisoned (4/4), offering a comprehensive view under different FL threat levels. Lie Detector remains robust across all settings. In the clean case (0/4), it correctly raises no alarms (FPR = 0%, DSR = 100%). As backdoored clients increase (1/4 to 3/4), DSR stays high (96.17%–99.17%) and FPR remains low ($\leq 1\%$), indicating strong sensitivity to injected backdoors without misclassifying clean models. Even in the hardest case (4/4), where no clean client exists, the method still achieves $>95\%$ DSR and $<1.5\%$ FPR across all attack types. This suggests Lie Detector can exploit subtle inconsistencies from imperfect backdoor optimization—even among colluding clients. Slightly higher FPRs are observed for ISSBA and Low-Frequency in high-poisoning scenarios, reflecting their stealthy nature, but overall resilience holds. These results demonstrate Lie Detector’s effectiveness for decentralized auditing in FL without requiring clean references, aggregation, or inter-client communication.

3) Scaling to Larger Models. We further assess Lie Detector on high-capacity vision-language models: VisualGLM-6B [4] and LLaVA-1.5-7B [11], whose GFLOPs are 191.1 and 76.6, respectively. We adopt two recent multi-modal backdoor attacks—TrojanVLM and Shadowcast—and construct 20 clean and 20 poisoned models per model-attack combination via fine-tuning with or without injected triggers. The experimental setup is same as the main paper. The results are in Tab. Appendix 5. On VisualGLM-6B, Lie Detector achieves a DSR of 90.0% and FPR of 5.0% under TrojanVLM, and

85.0% DSR and 10.0% FPR under Shadowcast. These results confirm that Lie Detector generalizes well to larger high-capacity backdoored models, making it a promising solution for securing next-generation foundation architectures.

Table Appendix 5: Detection results on large-scale VLMs under TrojanVLM and Shadowcast attacks. We use 20 clean and 20 poisoned models for evaluating VisualGLM-6B and LLaVA-1.5-7B.

Model	GFLOPs	Attack	DSR (%)	FPR (%)
VisualGLM-6B	191.1	TrojanVLM	90.00	5.00
		Shadowcast	85.00	10.00
LLaVA-1.5-7B	76.6	TrojanVLM	92.50	0.00
		Shadowcast	90.00	5.00

F Detailed Comparisons with Existing Backdoor Detection Methods

To provide a comprehensive understanding of the strengths of our method, we compare **Lie Detector** with several representative backdoor detection techniques, including post-training methods (MM-BD, NAD, ABS, NC, UNICORN, TED) and the pre-training method DECRE. The comparison covers multiple aspects such as computational cost, label and data dependency, applicable scenarios, limitations, and detection performance. A detailed summary is presented in Tab. Appendix 6.

Table Appendix 6: Comparison with existing backdoor detection methods. Cost: Computational Cost. Label: whether ground-truth labels are required. Performance: average DSR across datasets (from Tab. 1).

Method	Cost	Label Required	Data Dependency	Applicable Scenario	Limitation	Performance
MM-BD	Low	No	No clean data	Post-training	Weak on adaptive attacks	94.7%
NAD	High	Yes	Clean data needed	Mitigation	High cost	53.1%
ABS	High	Yes	Clean data needed	Detection	Poor for spatial triggers	52.5%
NC	Medium	Yes	Clean data needed	Detection	Poor for global triggers	32.5%
UNICORN	High	No	Clean data needed	Multi-trigger detection	High cost	81.6%
TED	High	Yes	Training dynamics	Topological analysis	Very high overhead	92.8%
DECRE	Low	No	No clean data	Pre-training (SSL/multimodal)	Pre-training only	92.8%
Lie Detector	Medium	No	Clean data only	Unified (SL/SSL/AL)	Assumes two models	99.7%

As shown in the table, many existing methods rely on ground-truth labels and clean training data, which may not always be available in practical scenarios. Several also operate under the white-box assumption or require training dynamics, making them less applicable to black-box or third-party verification settings.

In contrast, **Lie Detector** does not require label supervision or access to model internals, and is applicable across supervised, self-supervised, and autoregressive learning paradigms. It achieves state-of-the-art performance (99.7% DSR) while maintaining moderate computational cost, and uniquely supports unified detection in complex settings like multimodal LLMs. Also, as deonstrated in the main paper, our method has extremely low false positive rate.

The only minor limitation is the requirement of two independently trained models, which is a reasonable and realistic assumption third-party scenarios.

G Theoretical Properties of Similarity Metrics

G.1 Summary of Properties

We compare four commonly used similarity metrics—Cosine similarity, Canonical Correlation Analysis (CCA), Singular Vector CCA (SVCCA), and Centered Kernel Alignment (CKA)—across key theoretical properties. The comparison is summarized in Tab. Appendix 7.

Among all the evaluated similarity metrics, CKA uniquely satisfies all four desirable theoretical properties: it is invariant to isotropic scaling, sensitive to angular alignment, robust to architectural

Table Appendix 7: Comparison of theoretical properties across similarity metrics.

Metric	Scale Invariant	Angle Sensitive	Cross-Model Stable	Nonlinear Compatible
Cosine	No	Yes	Low (basis sensitive)	No
CCA	No	No	Medium (linear only)	No
SVCCA	Partial	No	Medium (SVD improves stability)	No
CKA	Yes	Yes	High	Yes

changes, and compatible with nonlinear relationships. These strengths are especially critical in our setting, where models may differ in architecture, training dynamics, or feature scales. In contrast, Cosine similarity lacks stability across bases, CCA fails under nonlinearity, and SVCCA only partially improves robustness through dimensionality reduction. CKA’s kernel-based formulation and normalization by Frobenius norm ensure consistent and meaningful comparisons across diverse model outputs, making it particularly well-suited for cross-model backdoor detection in the absence of clean references. We also provides the mathematical proofs in the following section.

G.2 Mathematical Proofs

1. Cosine Similarity [21, 17]

Definition: Given vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, cosine similarity is defined as:

$$\text{Cos}(\mathbf{a}, \mathbf{b}) = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|}$$

Property Proofs

- **Scale Invariance:** Cosine similarity is invariant to positive scalar multiplication:

$$\text{Cos}(\lambda \mathbf{a}, \mathbf{b}) = \frac{\lambda \langle \mathbf{a}, \mathbf{b} \rangle}{\lambda \|\mathbf{a}\| \cdot \|\mathbf{b}\|} = \text{Cos}(\mathbf{a}, \mathbf{b})$$

However, this does not hold under general affine or non-uniform scaling. It also fails under feature shuffling or reparametrization.

- **Angle Sensitivity:** Cosine similarity explicitly measures $\cos(\theta)$, the angle between \mathbf{a} and \mathbf{b} . For unit vectors:

$$\text{Cos}(\mathbf{a}, \mathbf{b}) = \cos(\theta)$$

- **Cross-Model Stability:** Cosine similarity is sensitive to feature basis. A rotation matrix R gives:

$$\text{Cos}(R\mathbf{a}, \mathbf{b}) \neq \text{Cos}(\mathbf{a}, \mathbf{b})$$

- **Nonlinear Compatibility:** Not compatible. Cosine similarity is a linear measure and does not preserve structure under nonlinear transformations.

2. Canonical Correlation Analysis (CCA) [8, 7]

Definition: Given two centered data matrices $X \in \mathbb{R}^{n \times p}$ and $Y \in \mathbb{R}^{n \times q}$, CCA finds projections $w_x \in \mathbb{R}^p$, $w_y \in \mathbb{R}^q$ that maximize the correlation between Xw_x and Yw_y :

$$\rho = \max_{w_x, w_y} \frac{w_x^\top \Sigma_{XY} w_y}{\sqrt{w_x^\top \Sigma_{XX} w_x} \cdot \sqrt{w_y^\top \Sigma_{YY} w_y}}$$

Property Proofs

- **Scale Invariance:** If $X' = DX$ for diagonal D , then:

$$\Sigma_{X'X'} = D\Sigma_{XX}D^\top, \quad \Sigma_{X'Y} = D\Sigma_{XY}$$

The correlation changes unless $D = \lambda I$, i.e., only isotropic scaling is invariant. Hence CCA is not generally scale-invariant.

- **Angle Sensitivity:** CCA finds directions maximizing correlation, not angle:

$$\text{corr}(Xw_x, Yw_y) \neq \cos(\theta)$$

No explicit relation to angular alignment \rightarrow not angle-sensitive.

- **Cross-Model Stability:** Sensitive to changes in basis; aligned projections across independently trained models are not guaranteed unless architectures match.
- **Nonlinear Compatibility:** CCA is linear; incapable of capturing nonlinear dependencies.

3. SVCCA [20]

Definition: SVCCA applies singular value decomposition to reduce noise, then uses CCA. Let $X \in \mathbb{R}^{n \times p}$:

$$X = U_X S_X V_X^\top, \quad \text{keep top } k \text{ components}$$

Apply CCA on U_X^k, U_Y^k .

Property Proofs

- **Scale Invariance:** If $X \rightarrow \lambda X$, then $S_X \rightarrow \lambda S_X$ and U_X is unchanged. So SVD step is scale-invariant. But since CCA is not, SVCCA is only partially scale-invariant.
- **Angle Sensitivity:** CCA is used after SVD. Since neither SVD nor CCA is angle-sensitive, SVCCA is not.
- **Cross-Model Stability:** SVD suppresses noise and basis sensitivity. Better than CCA.
- **Nonlinear Compatibility:** Still linear; no support for nonlinearity.

4. Centered Kernel Alignment (CKA) [3, 10, 1]

Definition: Given two activation matrices $A_1, A_2 \in \mathbb{R}^{n \times p}$ (rows are samples), define their Gram (kernel) matrices:

$$K_1 = H A_1 A_1^\top H, \quad K_2 = H A_2 A_2^\top H,$$

where $H = I - \frac{1}{n} \mathbf{1}\mathbf{1}^\top$ is the centering matrix that removes the mean from each feature vector.

Then the linear CKA similarity is defined as:

$$\text{CKA}(A_1, A_2) = \frac{\langle K_1, K_2 \rangle_F}{\|K_1\|_F \cdot \|K_2\|_F},$$

where $\langle A, B \rangle_F = \text{Tr}(A^\top B)$ is the Frobenius inner product and $\|A\|_F = \sqrt{\langle A, A \rangle_F}$ is the Frobenius norm.

Property Proofs:

- **Scale Invariance:** Suppose $A_1 \mapsto \lambda A_1$ and $A_2 \mapsto \mu A_2$, with scalars $\lambda, \mu \in \mathbb{R}$. Then:

$$K_1 \mapsto \lambda^2 H A_1 A_1^\top H = \lambda^2 K_1, \quad K_2 \mapsto \mu^2 K_2$$

Hence:

$$\text{CKA}(\lambda A_1, \mu A_2) = \frac{\lambda^2 \mu^2 \langle K_1, K_2 \rangle_F}{\lambda^2 \|K_1\|_F \cdot \mu^2 \|K_2\|_F} = \text{CKA}(A_1, A_2)$$

Therefore, CKA is invariant to isotropic scaling of inputs.

- **Orthogonal Invariance:** Suppose $A_1 \mapsto A_1 Q$ and $A_2 \mapsto A_2 R$ where Q, R are orthogonal matrices (i.e., $Q^\top Q = I, R^\top R = I$). Then:

$$A_1 A_1^\top \mapsto (A_1 Q)(A_1 Q)^\top = A_1 Q Q^\top A_1^\top = A_1 A_1^\top$$

Hence, K_1 and K_2 remain unchanged \rightarrow CKA is invariant to orthogonal transformations (rotations, reflections, etc.).

- **Angle Sensitivity:** Since the Frobenius inner product between two kernel matrices K_1 and K_2 reflects alignment between their feature spaces:

$$\langle K_1, K_2 \rangle_F = \sum_{i,j=1}^n K_1(i, j) \cdot K_2(i, j),$$

it is maximized when the two representations encode similar pairwise distances (i.e., angles) between samples.

Moreover, when the features are centered and normalized, CKA behaves similarly to cosine similarity in the kernel (pairwise similarity) space:

$$\text{CKA} = \cos \angle(K_1, K_2),$$

making it sensitive to representational misalignment.

- **Cross-Model Stability:** Due to centering (which removes mean differences) and Frobenius normalization (which removes magnitude differences), CKA is robust across model architectures, feature dimensionalities, and training dynamics.

It is also **basis-invariant**, meaning it evaluates the relative structure between representations regardless of coordinate systems:

$$\text{CKA}(A, A) = 1, \quad \text{CKA}(A, B) < 1 \text{ iff representations differ.}$$

- **Nonlinear Compatibility:** CKA is compatible with nonlinear feature mappings. For example, let $\phi : \mathbb{R}^p \rightarrow \mathcal{H}$ be a nonlinear map to a high-dimensional (possibly infinite) Hilbert space. Then kernel matrices are computed via:

$$K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$$

allowing CKA to measure similarity in both linear and nonlinear spaces by replacing $A_i A_i^\top$ with any positive semidefinite kernel K_i .

H Impact Statement

This work proposes a practical and general-purpose framework for detecting backdoors in machine learning models, particularly in outsourced or third-party training settings. The proposed cross-examination mechanism eliminates the need for a trusted clean model or prior attack knowledge, enabling robust verification across supervised, self-supervised, and autoregressive learning paradigms. Notably, it is the first to support backdoor detection in large multimodal vision-language models (e.g., LLaVA, MiniGPT-4), addressing a critical gap in securing foundation models. From a broader perspective, this work contributes to the growing need for AI accountability and trustworthy deployment, especially as AI models are increasingly developed by external vendors or deployed in critical applications such as healthcare, finance, and national security. By reducing reliance on assumptions about attackers or training access, the framework enhances the resilience of model verification protocols. On the social level, this research promotes transparency and auditability in machine learning pipelines, aligning with global efforts around AI governance and certification. While the method can expose malicious behaviors, it does not introduce harm, manipulate data, or compromise privacy. Nevertheless, continued evaluation is needed to ensure fairness in model comparisons and avoid mislabeling benign discrepancies as malicious behavior in edge cases.

I Limitation

Our framework assumes a semi-honest verification setting, where third-party providers independently train models and do not actively collude. While this assumption holds in many real-world applications—such as government or enterprise auditing, AutoML pipelines, or federated deployments with disjoint training—it may not capture stronger threat models. For instance, in collusion attacks, coordinated adversaries may align backdoored models to mask inconsistencies, potentially reducing the effectiveness of cross-model comparison. Similarly, while our framework can be extended to client-level inspection in federated learning, the lack of shared triggers or centralized visibility may limit its direct applicability without adaptation. Lastly, although we demonstrate competitive performance on large-scale models (e.g., MiniGPT-4), further exploration is needed to evaluate scalability under resource-constrained environments. These scenarios point to promising directions for future work rather than fundamental limitations, and our framework offers a solid foundation for extending to such advanced settings.

References

- [1] S. A. Alvarez. Gaussian rbf centered kernel alignment (cka) in the large-bandwidth limit. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6587–6593, 2022.
- [2] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.
- [3] C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.
- [4] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022.
- [5] S. Feng, G. Tao, S. Cheng, G. Shen, X. Xu, Y. Liu, K. Zhang, S. Ma, and X. Zhang. Detecting backdoors in pre-trained encoders. In *Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition*, pages 16352–16362, 2023.
- [6] T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.
- [7] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.
- [8] H. Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics: methodology and distribution*, pages 162–190. Springer, 1992.
- [9] J. Jia, Y. Liu, and N. Z. Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2043–2059. IEEE, 2022.
- [10] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In *Proc. Int’l Conf. Machine Learning*, pages 3519–3529. PMLR, 2019.
- [11] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. In *Proc. Annual Conf. Neural Information Processing Systems*, 2023.
- [12] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu. Invisible backdoor attack with sample-specific triggers. In *Proc. IEEE Int’l Conf. Computer Vision*, pages 16443–16452, 2021.
- [13] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021.
- [14] J. Liang, S. Liang, A. Liu, and X. Cao. Vl-trojan: Multimodal instruction backdoor attacks against autoregressive visual language models. *International Journal of Computer Vision*, pages 1–20, 2025.
- [15] S. Liang, M. Zhu, A. Liu, B. Wu, X. Cao, and E.-C. Chang. Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. In *Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition*, pages 24645–24654, June 2024.
- [16] Y. Liu, S. Ma, W.-C. Lee, Y. Aafer, G. Tao, and X. Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *CCS ’19: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.
- [17] W. I. D. Mining. Introduction to data mining. *Mining Multimedia Databases, Mining Time Series and*, 2006.
- [18] X. Mo, Y. Zhang, L. Y. Zhang, W. Luo, N. Sun, S. Hu, S. Gao, and Y. Xiang. Robust backdoor detection for deep learning via topological evolution dynamics. In *2024 IEEE Symposium on Security and Privacy (SP)*, 2024.

- 382 [19] T. A. Nguyen and A. T. Tran. Wanet-imperceptible warping-based backdoor attack. In *Proc. Int'l*
383 *Conf. Learning Representations*, 2021.
- 384 [20] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein. Svcca: Singular vector canonical corre-
385 lation analysis for deep learning dynamics and interpretability. In *Proc. Annual Conf. Neural*
386 *Information Processing Systems*, volume 30, 2017.
- 387 [21] A. Singhal et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*,
388 24(4):35–43, 2001.
- 389 [22] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao. Neural cleanse:
390 Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on*
391 *Security and Privacy (SP)*, 2019.
- 392 [23] H. Wang, Z. Xiang, D. J. Miller, and G. Kesidis. Mm-bd: Post-training detection of backdoor
393 attacks with arbitrary backdoor pattern types using a maximum margin statistic. In *2024 IEEE*
394 *Symposium on Security and Privacy (SP)*, 2024.
- 395 [24] Z. Wang, K. Mei, J. Zhai, and S. Ma. Unicorn: A unified backdoor trigger inversion framework.
396 In *Proc. Int'l Conf. Learning Representations*, 2023.
- 397 [25] Y. Xu, J. Yao, M. Shu, Y. Sun, Z. Wu, N. Yu, T. Goldstein, and F. Huang. Shadowcast:
398 Stealthy data poisoning attacks against vision-language models. In *Proc. Annual Conf. Neural*
399 *Information Processing Systems*, 2024.
- 400 [26] Y. Zeng, W. Park, Z. M. Mao, and R. Jia. Rethinking the backdoor attacks' triggers: A frequency
401 perspective. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 16473–16481, October 2021.
- 402 [27] L. Zhu, R. Ning, J. Li, C. Xin, and H. Wu. Seer: Backdoor detection for vision-language
403 models through searching target text and image trigger jointly. In *Proc. AAAI Conf. on Artificial*
404 *Intelligence*, pages 7766–7774, March 2024.