A1. More Implementation Details

More details of Trojan attacks. (1) Attack configuration for BadNets. We follow the attack methodology proposed in [29] to inject a backdoor during training. It attaches a trigger with a fixed size (5×5) and location (upper right corner) to benign images and injects them into the training set. Specifically, backdoored models are trained on a poisoned dataset with a poison ratio of 1% and the target label is set to 0 throughout the experiment.
(2) Attack configuration for Clean Label Backdoor Attack [94]. This method hinders the model from learning the true salient characteristics of the input through perturbations, often adversarial examples or data generated from GAN. Thus, the learned representations of the images with a target label are distorted towards another class and the content-label mismatch can be achieved in such a manner. In our experiments, we choose the PGD attack [58] to generate adversarial examples for the target class. For each image, we perform a 10-step PGD attack on a robustly trained surrogate ResNet-20s model with an attack budget $\epsilon = 8/255$ and an attack learning rate of $\alpha = 2/255$. The perturbed images are then further attached with the colorful or black trigger as aforementioned. We perturb all the images in the target class to guarantee a successful attack.

For a recovered trigger $(\mathbf{m}, \mathbf{\Delta})$, we evaluate the ell_1 norm of soft mask $\mathbf{m}$, and then binarize this mask so that its ell_1 norm equals the ground-truth value (5×5 for CIFAR-10/100 and 64×64 for R-ImageNet). Then we stamp $\mathbf{\Delta}$ with the binary mask to the test images and calculate the attack successful rate (ASR).

More details of reverse engineering. We use Neural Cleanse [79] as our backbone to conduct trigger reverse engineering. The detection includes two stages. In the first stage, potential triggers with the possibly least norm towards *each* class are obtained through a gradient-descent-based optimization algorithm. The final synthetic trigger and its target label are then determined through an anomaly detector. In the meantime, early stopping is performed as a trick to speed up the trigger recovery.

For trigger recovering, we default to use 100 noise images generated by Gaussian distribution $\mathcal{N}(0, 1)$. And we also compare the quality of recovered triggers from 10 and 100 clean images in Table 3 as an ablation study.

Each time, we pruned 20% of the remaining parameters with the lowest magnitude and then rewind the weight to epoch 3 before retraining.

A2. More Experiment Results

In this section, we not only provide comprehensive ablation studies including ① the fine-tuning steps for Trojan ticket detection; ② the configurations of Trojan attacks such as the trigger locations; ③ LTH pruning ratios and comparisons with other pruning methods, but also offer ④ visualizations of winning Trojan ticket's sparse connectivities and loss landscape geometry; ⑤ pruning dynamics of models with the clean-label Trojan trigger; ⑥ extra results of stealthier and global triggers; ⑦ extra results on more datasets; ⑧ extra results on un-poisoned datasets; ⑨ failure case analyses.

In addition, for the performance of recovered triggers, we present extra results of oracle labels (i.e., the truth target class) together with other two sparse Trojan tickets: (i) H-Trojan ticket with high SA and ASR; (ii) L-Trojan ticket with low SA (standard testing accuracy) and ASR, as collected in Table A4, A5, A6, and A7.

Ablation for the fine-tuning steps k. We explore the effect of the number of fine-tuning steps k. The successful rate of detecting winning Trojan tickets is shown in Figure A7. It is calculated from **ten** replicates and each replicate is fine-tuned for k steps. We find that choosing fine-tuning steps $k \geq 7$ is potentially enough to accurately identify the winning Trojan tickets.

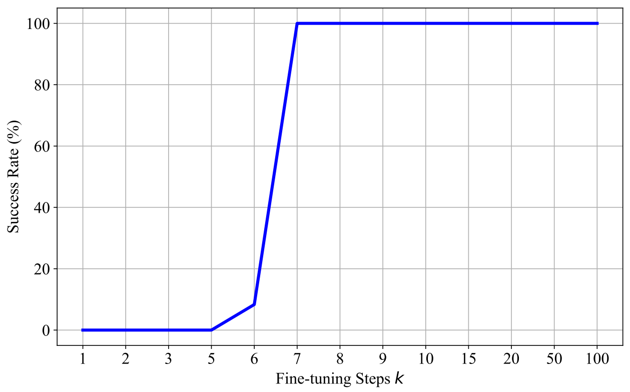


Figure A7. The successful rate over fine-tuning steps of detecting winning Trojan tickets. ResNet-20s on CIFAR-10 with RGB triggers are adopted here.

Ablation on trigger locations. We study the different positions for placing Trojan triggers. As shown in Table A4, winning Trojan ticket demonstrates a consistent superiority in terms of recovered triggers' ASR.

Ablation on pruning ratios. We investigate the pruning ratio in LTH pruning [18]. Results of pruning ratio $p = 10\%, 20\%$, and 40% are presented in Figure A9, we observe that $p = 10\%$ or 20% are capable of generating the winning Trojan tickets, while $p = 40\%$ fails. A possible explanation is that pruning with $p = 40\%$ is too aggressive to maintain Trojan information.

Comparison with other pruning methods. In Figure A10, we compare LTH pruning [18] with other pruning methods like random pruning (RP), one-shot magnitude

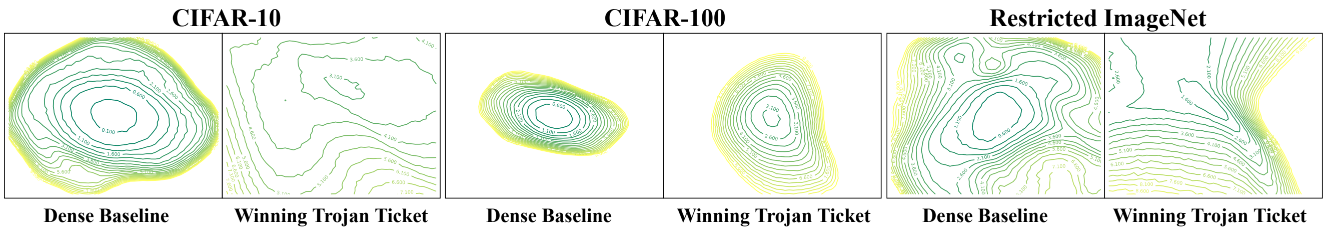


Figure A8. Loss landscape geometry of dense Trojan models and their identified Trojan tickets from CIFAR-10/100, and ImageNet.

Table A4. Performance of recovered triggers with ResNet-20s on CIFAR-10. The RGB Trojan attack is applied to different positions, including *bottom left* and *upper right*.

bottom left	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("5", 121.9) ✗	10.5%	("1", 251.6)	13.7%
Winning Trojan ticket	("1", 79.3) ✓	**86.7%**	("1", 79.3)	**86.7%**
H-Trojan ticket	("4", 104.8) ✗	21.3%	("1", 189.1)	14.2%
L-Trojan ticket	("2", 158.6) ✗	18.7%	("1", 231.3)	42.1%
upper right	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("1", 78.7) ✓	48.0%	("1", 78.7)	48.0%
Winning Trojan ticket	("1", 29.8) ✓	**99.6%**	("1", 29.8)	**99.6%**
H-Trojan ticket	("7", 110.9) ✗	8.6%	("1", 124.9)	18.3%
L-Trojan ticket	("2", 105.0) ✗	58.5%	("1", 276.14)	17.1%

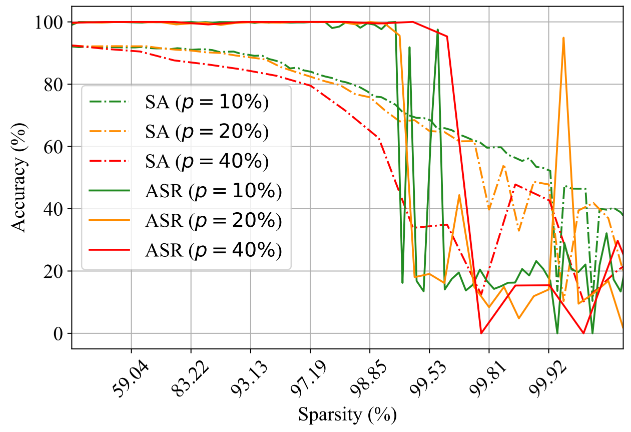


Figure A9. Ablation on the pruning ratio of Trojan ticket findings. The standard testing accuracy (SA %) and attack successful rate (ASR %) are reported over network sparsity. ResNet-20s and CIFAR-10 with RGB Trojan tickets are adopted here.

pruning (OMP), and SNIP [50]. We find that our proposals can be effective across different pruning methods. All of LTH pruning, OMP, and SNIP produce winning Trojan tickets. We also notice that random pruning can not make it, which supports that appropriate sparsity plays a significant role in capturing Trojan information.

Visualization of sparse masks and loss surfaces. We visualize the located winning Trojan ticket in Figure A11, and their loss landscape geometries in Figure A8. We find that winning Trojan tickets usually have sharp local minima, suggesting a potentially performance gap between before and after fine-tuning which lays the foundation of our

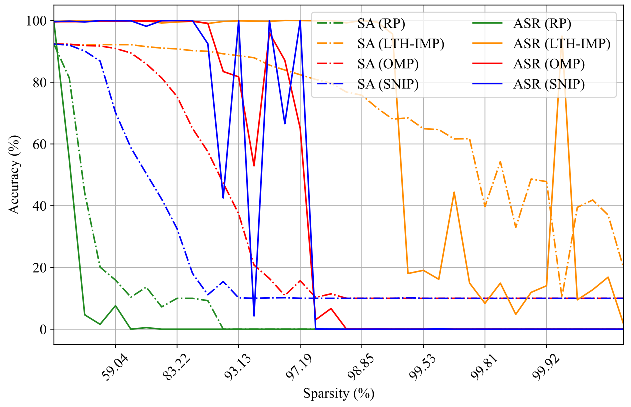


Figure A10. Ablation on the pruning algorithms of Trojan ticket findings, including RP, OMP, GraSP, SNIP and LTH-IMP (ours). The standard testing accuracy (SA %) and attack successfully rate (ASR %) are reported over network sparsity.

proposed detection methods.

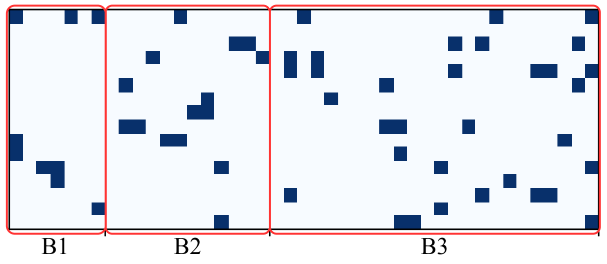


Figure A11. Kernel-wise heatmap visualizations of the winning Trojan ticket with 0.05% sparsity, 11.38% SA, and 94.49% ASR. The bright blocks represent the completely pruned (zero) kernels and the dark blocks stand for the kernels that have at least one unpruned weight. B1 $\sim$ 3 donate three residual blocks in the ResNet-20s. CIFAR-10 with RGB triggers is used.

Pruning dynamic and Trojan scores of the clean-label Trojan trigger. Figure A12 collects the pruning dynamics and Trojan scores on CIFAR-10 dataset with ResNet-20s and the clean-label Trojan trigger, where consistent conclusions can be drown.

Extra results of stealthier and global triggers. We conduct experiments on another advanced Trojan attack,

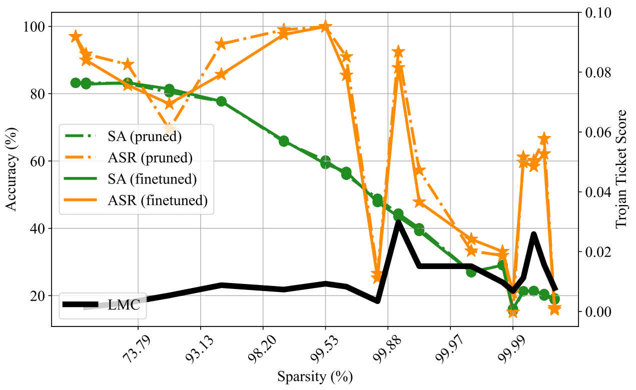


Figure A12. The pruning dynamics and Trojan scores on CIFAR-10 with ResNet-20s using the clean-label Trojan trigger. The peak Trojan score precisely characterizes the winning Trojan ticket.

WaNet [62] which advocates stealthier and global triggers. Results included in below table consistently justified the effectiveness of our approaches. Note that, to enable meaningful comparison, we disable the additive Gaussian noise in [62]; otherwise both dense baseline and winning Trojan tickets are failed in reverse engineering.

Settings	Noise Images ('Free')	(Detected, ℓ_1)	ASR
(CIFAR-10, ResNet-20s)	Dense baseline	("1", 31.3) ✓	40.1%
with **WaNet** Backdoor	Winning Trojan ticket	("1", 28.5) ✓	**96.71**%

Extra results on more datasets. Extra experiments are conducted on three more datasets, including MNIST [14], GTSRB [41], and YouTubeFace [82]. Results in the table below reveal similar conclusions as the ones in the main text. For example, as a highly challenging scenario for trigger recovery, the YouTubeFace contains 1283 classes and the dense baseline method suffers from unsatisfactory results. However, our winning Trojan tickets still succeed in restoring the trigger towards the right target label with a decent ASR.

Settings	Noise Images ('Free')	(Detected, ℓ_1)	ASR
(**MNIST**, ResNet-20s)	Dense baseline	("1", 34.0) ✓	100%
with RGB Triggers	Winning Trojan ticket	("1", 36.4) ✓	**100%**
(**GTSRB**, ResNet-20s)	Dense baseline	("1", 91.7) ✓	54.02%
with RGB Triggers	Winning Trojan ticket	("1", 16.9) ✓	**98.89%**
(**YouTube Face**, ResNet-20s)	Dense baseline	("334", 612.9) ✗	6.23%
with RGB Triggers	Winning Trojan ticket	("1", 659.3) ✓	**67.03%**

Extra results on un-poisoned datasets. We conducted Trojan detection experiments (in terms of Trojan trigger recovering) on a clean training set. It is shown from the following table that the use of winning Trojan ticket yields similar norms of the recovered triggers across all labels. This will not activate the Trojan detector, and thus, will not flag non-Trojan datasets as the Trojan one. The detection results that we achieved are consistent with [80].

Settings	Label w. ℓ_∞	0	1	2	3	4	5	6	7	8	9
Clean Training Set	Dense baseline	254	107	123	126	233	169	187	260	265	207
(CIFAR-10, ResNet-20s)	Winning Trojan ticket	436	292	476	379	300	303	301	240	392	259

Extra results of recovered triggers. We show additional results of oracle labels (i.e., the truth target class) together with other two sparse Trojan tickets: (i) H-Trojan ticket with high SA and ASR; (ii) L-Trojan ticket with low SA (standard testing accuracy) and ASR, as collected in Table A5, A6, and A7.

Table A5. Performance of recovered triggers with the RGB Trojan attack and ResNet-20s on CIFAR-10. Different number of clean validation images are used for the reverse engineering.

Noise Images	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("1", 78.7) ✓	48.0%	("1", 78.7)	48.0%
Winning Trojan ticket	("1", 29.8) ✓	99.6%	("1", 29.8)	99.6%
H-Trojan ticket	("7", 110.9) ✗	8.6%	("1", 124.9)	18.3%
L-Trojan ticket	("2", 105.0) ✗	58.5%	("1", 276.14)	17.1%
10 Clean Images	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("1", 65.6) ✓	77.2%	("1", 65.6)	77.2%
Winning Trojan ticket	("1", 28.3) ✓	99.7%	("1", 28.3)	99.7%
H-Trojan ticket	("3", 171.4) ✗	10.5%	("1", 190.4)	38.2%
L-Trojan ticket	("2", 124.5) ✗	58.0%	("1", 275.2)	18.0%
100 Clean Images	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("1", 174.6) ✓	72.6%	("1", 174.6)	72.6%
Winning Trojan ticket	("1", 40.4) ✓	99.8%	("1", 40.4)	99.8%
H-Trojan ticket	("5", 203.8) ✗	13.9%	("1", 211.5)	32.5%
L-Trojan ticket	("2", 220.7) ✗	56.7%	("1", 326.1)	17.9%

Table A6. Performance of recovered triggers with ResNet-20s on CIFAR-10 across diverse Trojan triggers. ✓/✗ mean the detected target label is matched/unmatched with the truth target label.

Gray-scale Trigger	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("1", 196.8) ✓	71.4%	("1", 196.8)	71.4%
Winning Trojan ticket	("1", 68.0) ✓	91.2%	("1", 68.0)	91.2%
H-Trojan ticket	("3", 217.5) ✗	9.7%	("1", 294.1)	30.9%
L-Trojan ticket	("7", 79.7) ✗	52.1%	("1", 398.8)	13.7%
RGB Trigger	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("1", 78.7) ✓	48.0%	("1", 78.7)	48.0%
Winning Trojan ticket	("1", 29.8) ✓	99.6%	("1", 29.8)	99.6%
H-Trojan ticket	("7", 110.9) ✗	8.6%	("1", 124.9)	18.3%
L-Trojan ticket	("2", 105.0) ✗	58.5%	("1", 276.1)	17.1%
Clean-label Trigger	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("1", 48.6) ✓	9.6%	("1", 48.6)	9.6%
Winning Trojan ticket	("1", 14.0) ✓	99.8%	("1", 14.0)	99.8%
H-Trojan ticket	("1", 21.0) ✓	28.3%	("1", 21.0)	28.3%
L-Trojan ticket	("6", 73.6) ✗	64.3%	("1", 158.2)	40.9%

Failure case analyses of identifying winning Trojan tickets on un-poisoned datasets. To comprehensively investigate the effectiveness of finding Trojan winning tickets on un-poisoned datasets, we repeat the experiments with **ten** different random seeds, and there are only 2 of 10 cases where LMC identifies the wrong occurrence of ASR peaks.

Failure case analyses of identifying winning Trojan tickets with clean-label attacks. Clean-label Trojan triggers as one of the most challenging attacks may encounter some failure cases during the detection of winning Trojan tickets.

Table A7. Performance of recovered triggers with RGB Trojan attack across diverse (network architecture, dataset) combinations.

(ResNet-18, CIFAR-10)	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("3", 77.5) ✗	13.0%	("1", 151.0)	10.9%
Winning Trojan ticket	("1", 10.55) ✓	81.8%	("1", 10.55)	81.8%
H-Trojan ticket	("1", 8.15) ✓	22.9%	("1", 8.15)	22.9%
Bad subnetwork	("10", 135.2) ✗	11.5%	("1", 253.4)	15.6%

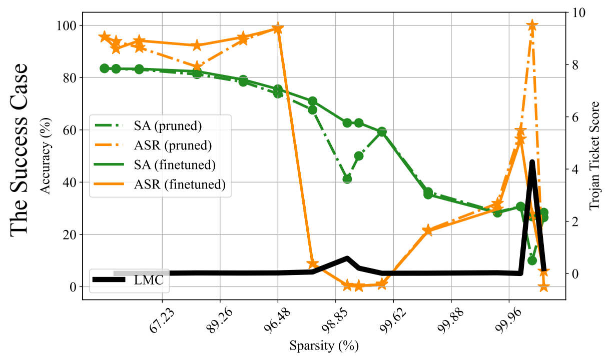
(DenseNet-100, CIFAR-10)	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("1", 6.4) ✓	13.7%	("1", 6.4)	13.7%
Trojan tickets	("1", 67.8) ✓	66.9%	("1", 67.8)	66.9%
H-Trojan ticket	("1", 10.0) ✓	17.7%	("1", 10.0)	17.7%
L-Trojan ticket	("1", 173.6) ✓	8.5%	("1", 173.6)	8.5%

(VGG-16, CIFAR-10)	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("1", 83.3) ✓	33.6%	("1", 83.3)	33.6%
Winning Trojan ticket	("1", 15.0) ✓	100.0%	("1", 15.0)	100.0%
H-Trojan ticket	("7", 140.5) ✗	8.0%	("1", 171.7)	10.1%
L-Trojan ticket	("7", 208.6) ✗	33.2%	("1", 602.4)	19.6%

(ResNet-20s, CIFAR-100)	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("1", 149.9) ✓	13.8	("1", 149.9)	13.8
Winning Trojan ticket	("1", 132.7) ✓	98.7	("1", 132.7)	98.7
H-Trojan ticket	("1", 63.0) ✓	83.3	("1", 63.0)	83.3
L-Trojan ticket	("55", 233.1) ✗	3.4	("1", 652.8)	10.2

(ResNet-18, R-ImageNet)	(Detected, ℓ_1)	ASR	(Oracle, ℓ_1)	ASR
Dense baseline [32]	("9", 13.9) ✗	9.8	("1", 1179.0)	97.7
Winning Trojan ticket	("1", 193.1) ✓	98.7	("1", 193.1)	98.7
H-Trojan ticket	("7", 22.6) ✗	4.7	("1", 556.1)	90.4
L-Trojan ticket	("5", 142.3) ✗	99.6	("1", 1043.3)	97.3

Specifically, we conduct **ten** replicates with diverse random seeds, and there are 3 of 10 cases where LMC can not accurately locate the winning Trojan ticket. One success and one failure cases are collected in Figure A13.



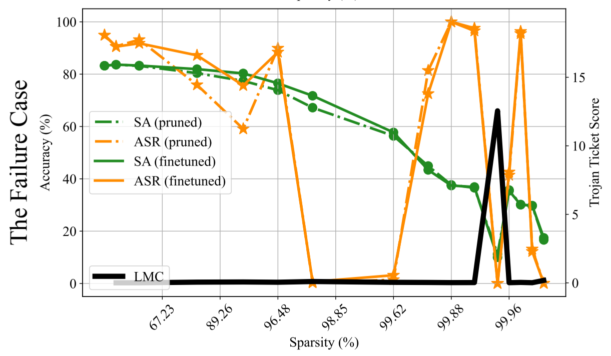


Figure A13. Success (*Top*) and failure (*Bottom*) cases of identifying winning Trojan tickets with clean-label attacks on CIFAR-10 and ResNet-20s. Sufficient fine-tuning steps are conducted.