A ATTACK MODEL AND RELATED WORK

We survey relevant train-time attacks.

A.1 THREAT MODEL

We consider the following train time attack known as the backdoor attack. We assume that the attacker has a trigger function of choice and their goal is to corrupt the training of a model such that the corrupted model outputs a prediction of a target label when shown an example with the trigger function applied to it. The attacker can inject a small number of arbitrarily corrupted examples to the training set to achieve this goal. The success will be measures by Attack Success Rate (ASR), which is the probability of the prediction of a trigger-applied test example being the target label. One attack, which consists of a set of a fixed number of injected corrupted examples, is said to be stronger than another if the ASR is higher than the other given the same number of injected corrupted examples.

We assume that the attacker has knowledge of the target model's architecture and training data, and can and leverage this information to increase the ASR of the backdoor attacks. In Appendix A.2 we list several prior works make similar assumptions for both data-poisoning and backdoor attacks. We believe knowledge of the training architecture can be motivated by the widespread usage of popular architectures such as ResNets (He et al., 2016). We also perform an ablation study in §3.3 indicating that our method degrades gracefully when only a small subset of the training data is known.

A.2 BACKDOOR ATTACKS

Backdoor attacks as presented in §1 are introduced in Gu et al. (2017). In backdoor attacks, the two most important design choices are the choice of trigger P and the method of producing the poison data X_{p}. Many works design P to appear benign to humans Gu et al. (2017); Barni et al. (2019); Liu et al. (2020); Nguyen & Tran (2020) or directly optimize P to this end Li et al. (2020); Doan et al. (2021b). Poison data X_{p} has been constructed to include no mislabeled examples Turner et al. (2019); Zhao et al. (2020) and optimized to evade detection through visual inspection Saha et al. (2020) and statistical inspection of latent representations Shokri et al. (2020); Doan et al. (2021a); Xia et al. (2022); Chen et al. (2017). Such backdoor attacks have been demonstrated in a wide variety of settings, including federated learning Wang et al. (2020b); Bagdasaryan et al. (2020); Sun et al. (2019), transfer learning Yao et al. (2019); Saha et al. (2020), and generative models Salem et al. (2020); Rawat et al. (2021). However, our goal of designing strong *few-shot backdoor attacks* has not been addressed with an exception of an influential earlier work of Koh et al. (2022). We consider the same threat model as in (Koh et al., 2022) where the attacker has information about the network's architecture and training data. However, our results are incomparable to those of (Koh et al., 2022) which focuses on linear models. The KKT attack of (Koh et al., 2022) leveraging decoy parameters cannot be used when the input dimension is far smaller than the parameter dimension and the influence attack of (Koh et al., 2022) cannot scale to large models, such as the WideResNet we use in our experiments.

Few-shot data attacks have been studied in contexts other than backdoor attacks. In *targeted* backdoor attacks, the attacker aims to control the network's output on a specific test instance Shafahi et al. (2018); Barni et al. (2019); Guo & Liu (2020); Aghakhani et al. (2021); Huang et al. (2020). *Data poisoning attacks* are similar to backdoor attacks with the alternate goal of reducing the generalization performance of the resulting model. Poison data X_{p} has been optimized to produce stronger data poisoning attacks using influence functions Koh et al. (2022); Yang et al. (2017), back-gradients Muñoz-González et al. (2017), and the neural tangent kernel Yuan & Wu (2021).

Following Gu et al. (2017), there has also been substantial work on detecting and defending against backdoor attacks. When the defender has access to known-clean data, they can filter the data using outlier detection Liang et al. (2018); Lee et al. (2018); Steinhardt et al. (2017), retrain the network so it forgets the backdoor Liu et al. (2018), or train a new model to test the original for a backdoor Kolouri et al. (2020). Other defenses assume P is an additive perturbation with small norm Wang et al. (2019); Chou et al. (2020), rely on smoothing Wang et al. (2020a); Weber et al. (2020), filter or penalize outliers without clean data Gao et al. (2019); Sun et al. (2019); Steinhardt et al. (2017); Blanchard et al. (2017); Pillutla et al. (2019); Tran et al. (2018); Hayase et al. (2021) or use Byzantine-tolerant distributed learning techniques Blanchard et al. (2017); Alistarh et al. (2018); Chen et al. (2018). Backdoors cannot be detected in planted neural networks in general Goldwasser et al. (2022).

B IMPLEMENTATION DETAILS

In §2, we give a brief description of the Neural Tangent Backdoor Attack. Further details regarding the implementation are given here.

B.1 OPTIMIZATION DETAILS

We use L-BFGS-B by adapting the wrapper of Virtanen et al. (2020) for use with JAX. We found that simple first order methods such as gradient descent with momentum and Adam Kingma & Ba (2015) converged very slowly with small learning rates and were unable to achieve good minima with larger learning rates. In contrast, the strong Wolfe line search of L-BFGS-B appears to choose step sizes which lead to relatively rapid convergence for our problem.

B.2 COMPUTATIONAL RESOURCES

All neural networks were trained on a single Nvidia 2080 Ti. We ran NTBA optimization on a machine with four Nvidia A100 GPUs for a duration between 5 hours and 12 hours depending on the number of poisons being optimized. Before optimization begins, we precompute the $K_{\mathrm{d,dta}}$ matrix using Nvidia A100 GPUs, requiring a total of 2 GPU hours for double precision.

B.3 DETAILS FOR NTK AND LAPLACE KERNEL COMPARISON

In these experiments, we compare the NTK of a 3-layer feed-forward neural network with a Laplace kernel with bandwidth tuned to match the NTK in the small-distance limit. This is the same setup used in Appendix E. For the attack, we use the same parameters as for our main experiments in Table 2, except we do not report transfer numbers since the Laplace kernel has no associated neural network.

B.4 DETAILS FOR LABEL CONSISTENT ATTACK

In Fig. 1, we compare the NTBA to the sampling baseline as well as the label consistent attack of (Turner et al., 2019). To produce the label consistent numbers, we use the precomputed poisoned datasets provided in the paper for the ℓ_∞ adversarial perturbations with $\varepsilon = 8$ and GAN latent interpolation with $\tau = 0.3$. The line reported is the best poison test accuracy obtained over these two attacks.

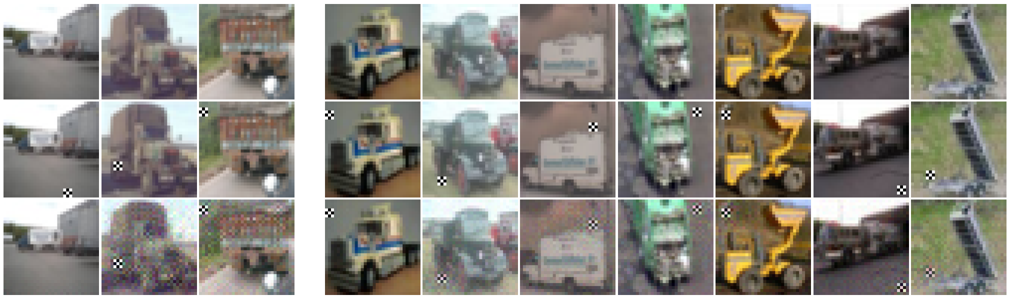
C SUPPLEMENTARY EXPERIMENTAL RESULTS

We report further experimental results complimenting those of §3.

C.1 RESULTS FOR PATCH TRIGGER ON CIFAR-10

We repeat the experiments of §3.1 using a 3 × 3 checkered patch as the backdoor trigger. Example images for this attack are shown in Fig. 8. We plot the ASR vs. the number of poisoned images in Fig. 9 with numerical results reported in Table 5.

We note that for some images in Fig. 8, the trigger becomes partially faded out after optimization while for other images the trigger remains unchanged. We believe this may be due to the optimization getting stuck in a local minima nearby some images, preventing it from erasing the triggers as we would expect according to the analysis in §4. This may partly explain why the attacks computed for the patch trigger are not as strong as those computed for the periodic trigger.



(a) $m = 3$ (b) $m = 10$

Figure 8: Images produced by backdoor optimization for the patch trigger and $m \in \{3, 10\}$. The top row shows the original clean image, the middle row shows the image with the trigger applied, and the bottom row shows the poisoned image after optimization. Duplicate images have been omitted to save space.

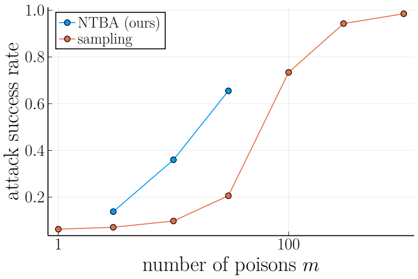


Figure 9: The trade-off between the number of poisons and ASR for the patch trigger.

C.2 RESULTS FOR PERIODIC TRIGGER ON IMAGENET

We also use NTBA to attack a ConvNeXt-tiny Liu et al. (2022) ($d \approx 2.8 \times 10^7$) trained on a 2 label subset of ImageNet. We use “slot” as the source label and “Australian terrier” as the target label following the examples from Saha et al. (2020). We consider both the case where the ConvNeXt is initialized randomly and trained from scratch and the case where it has been pretrained on ImageNet and fine-tuned as in Saha et al. (2020). The results for these two settings are shown in Figs. 10 and 11 respectively. When trained from scratch, the clean accuracy of the ConvNeXt remains above 90% in all cases. When pretrained and fine-tuned, the ConvNeXt achieves at least 99% clean accuracy in all cases.

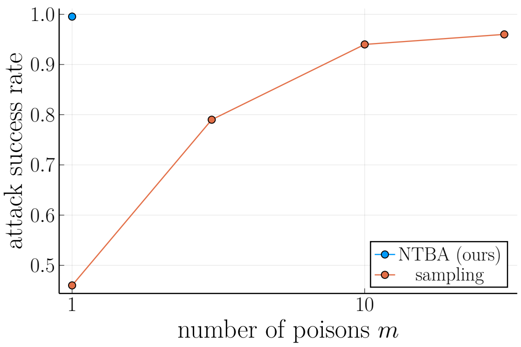


Figure 10: The trade-off between the number of poisons and ASR for ConvNeXt trained from scratch.

We note that ConvNeXt is surprisingly vulnerable to backdoors when trained from scratch, as even a single random poisoned image is sufficient to achieve 50% ASR and NTBA is able to achieve

Table 5: ASR of NTBA ($\mathrm{asr}_{\mathrm{nn,te}}$) is significantly higher than the ASR for the baseline of the sampling based attack using the same patch trigger, across a range of poison budgets m. Clean accuracy $\mathrm{acc}_{\mathrm{nn,te}}$ remains above 92.6% in all cases.

	ours				sampling
m	$\mathrm{asr}_{\mathrm{ntk,tr}}$	$\mathrm{asr}_{\mathrm{ntk,te}}$	$\mathrm{asr}_{\mathrm{nn,tr}}$	$\mathrm{asr}_{\mathrm{nn,te}}$	$\mathrm{asr}_{\mathrm{nn,te}}$
3	99.9	74.1	0.9	13.8	7.1
10	99.0	79.8	37.7	36.0	9.8
30	93.8	82.3	66.8	65.5	20.6

	sampling
m	$\mathrm{asr}_{\mathrm{nn,te}}$
0	6.2
1	6.3
100	73.4
300	94.3
1000	98.5

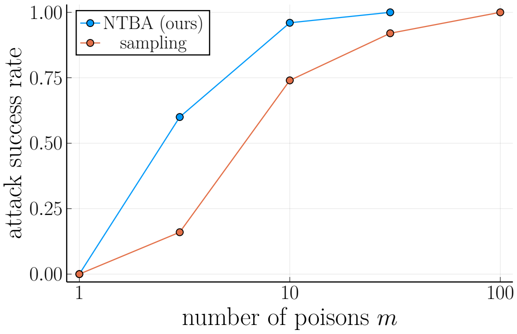


Figure 11: The trade-off between the number of poisons and ASR for ConvNeXt pretrained on ImageNet.

100% ASR with a single image. With pretraining, the ConvNeXt becomes slightly more resistant to backdoors, but the periodic attack remains quite strong. We give numerical results in Table 6.

m	NTBA	sampling
0		10
1	100	46
3		78
10		94
30		96

(a) trained from scratch

m	NTBA	sampling
0		0
1	0	0
3	60	16
10	96	74
30	100	92
100		100

(b) pretrained

Table 6: $\mathrm{asr}_{\mathrm{nn,te}}$ results for ConvNeXt on ImageNet. Numbers are percentages over the 50 examples from the source label.

D ANALYSIS OF BACKDOORS FOR KERNEL LINEAR REGRESSION

To explain the phenomena observed in §4, we take Taylor approximations of ϕ at $\widetilde{\boldsymbol{x}}_\mathrm{p}$ and $\widetilde{\boldsymbol{x}}_\mathrm{a}$ and obtain,

$$\begin{aligned} & f(\boldsymbol{x}_\mathrm{a}; D_\mathsf{d} \cup \{(\boldsymbol{x}_\mathrm{p}, y_\mathrm{p})\}) - f(\boldsymbol{x}_\mathrm{a}; D_\mathsf{d}) \\ & \quad \approx \frac{(\phi(\widetilde{\boldsymbol{x}}_\mathrm{p}) + \mathrm{D}\phi(\widetilde{\boldsymbol{x}}_\mathrm{p})\boldsymbol{\Delta}_\mathrm{p})(I - P)(\phi(\widetilde{\boldsymbol{x}}_\mathrm{a}) + \mathrm{D}\phi(\widetilde{\boldsymbol{x}}_\mathrm{a})\boldsymbol{\Delta}_\mathrm{a})^\top}{(\phi(\widetilde{\boldsymbol{x}}_\mathrm{p}) + \mathrm{D}\phi(\widetilde{\boldsymbol{x}}_\mathrm{p})\boldsymbol{\Delta}_\mathrm{p})(I - P)(\phi(\widetilde{\boldsymbol{x}}_\mathrm{p}) + \mathrm{D}\phi(\widetilde{\boldsymbol{x}}_\mathrm{p})\boldsymbol{\Delta}_\mathrm{p})^\top}(y_\mathrm{p} - f(\boldsymbol{x}_\mathrm{p}; D_\mathsf{d})) \end{aligned}$$

$$= \underbrace{\frac{\langle \mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}})\boldsymbol{\Delta}_{\mathrm{p}}, \mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{a}})\boldsymbol{\Delta}_{\mathrm{a}}\rangle_{(I-P)}}{\|\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}})\boldsymbol{\Delta}_{\mathrm{p}}\|^2_{(I-P)}}}_{\triangleq A} \underbrace{(y_{\mathrm{p}} - f(\boldsymbol{x}_{\mathrm{p}}; D_{\mathsf{d}}))}_{\approx 2} ,$$

where $\mathrm{D}\phi(\widetilde{\boldsymbol{x}})$ denotes the Jacobian of the feature mapping ϕ at $\widetilde{\boldsymbol{x}}$ and w.l.o.g. we assume that $y_{\mathrm{p}} = 1$ and $f(\boldsymbol{x}_{\mathrm{p}}; D_{\mathsf{d}}) \approx -1$. The last step follows because $(I-P)\phi(\widetilde{\boldsymbol{x}}_{\mathrm{a}}) = (I-P)\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}}) = \mathbf{0}$. Note that if $A = 1$, then $f(\boldsymbol{x}_{\mathrm{a}}; D_{\mathsf{d}} \cup \{(\boldsymbol{x}_{\mathrm{p}}, y_{\mathrm{p}})\}) = y_{\mathrm{p}}$ which would imply a succesful attack for $\boldsymbol{x}_{\mathrm{a}}$. Since the goal of the attack is to control the prediction whenever $\boldsymbol{\Delta}_{\mathrm{a}}$ is applied to *any* clean point $\widetilde{\boldsymbol{x}}$, there may exist some $\widetilde{\boldsymbol{x}}$ where $\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{a}})\boldsymbol{\Delta}_{\mathrm{a}}$ does not align well with $\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}})\boldsymbol{\Delta}_{\mathrm{p}}$, which would make the numerator of A small. For the backdoor to succeed for these points, $\|\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}})\boldsymbol{\Delta}_{\mathrm{p}}\|_{(I-P)}$ must be small enough to overcome this misalignment, since the denominator of A scales as $\|\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}})\boldsymbol{\Delta}_{\mathrm{p}}\|^2_{(I-P)}$ while the numerator scales as $\|\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}})\boldsymbol{\Delta}_{\mathrm{p}}\|_{(I-P)}$. In particular, for the attack to succeed on a set of poisoned data points X_{a}, we need

$$\|\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}})\boldsymbol{\Delta}_{\mathrm{p}}\|_{(I-P)} \leq c \left(\min_{\boldsymbol{x}_{\mathrm{a}} \in X_{\mathrm{a}}} \left\langle \frac{\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}})\boldsymbol{\Delta}_{\mathrm{p}}}{\|\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}})\boldsymbol{\Delta}_{\mathrm{p}}\|_{(I-P)}}, \mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{a}})\boldsymbol{\Delta}_{\mathrm{a}} \right\rangle_{(I-P)} \right) , \qquad (7)$$

for some constant $c > 0$. Note that the test-time trigger $\boldsymbol{\Delta}_{\mathrm{a}}$ and therefore the distribution of $\boldsymbol{x}_{\mathrm{a}}$'s is fixed. Therefore Eq. (7) can be satisfied by choosing the train-time perturbation $\boldsymbol{\Delta}_{\mathrm{p}}$ to have small enough norm on the LHS of Eq. (7). This implies that smaller perturbations in the train-time poison data are able to successfully change the predictions on more examples at test-time, and hence they correspond to a stronger attack. We can make this connection more realistic by considering multiple poisoned examples injected together. As the size $m \triangleq |D_{\mathrm{p}}|$ of the injected poisoned dataset D_{p} increases, we may distribute the poisoned examples so that each test point $\boldsymbol{x}_{\mathrm{a}}$ is covered by some poison point $\boldsymbol{x}_{\mathrm{p}} \in X_{\mathrm{p}}$ that aligns well with it. Since the worst-case alignment between poison and test data will be higher, the RHS of Eq. (7) will be larger so the LHS may be larger as well. This means that for each poison, the size of the trigger $\|\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}})\boldsymbol{\Delta}_{\mathrm{p}}\|_{(I-P)}$ may be larger (and still achieve a high attack success rate) when we are adding more poison data.

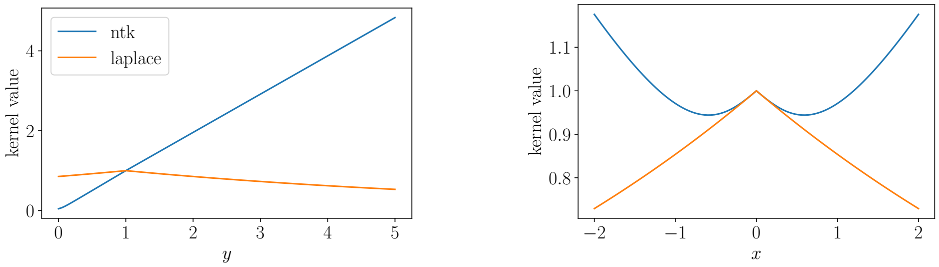
Two further insights from Eq. (7) shows the strengths of NTBA. First, Eq. (7) suggests that there is potential for improvement by designing train-time perturbations $\boldsymbol{\Delta}_{\mathrm{p}}$ that adapt to the local geometry of the feature map, represented by $\mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{p}}), \mathrm{D}\phi(\widetilde{\boldsymbol{x}}_{\mathrm{a}})$, around clean data points $\widetilde{\boldsymbol{x}}_{\mathrm{p}}, \widetilde{\boldsymbol{x}}_{\mathrm{a}}$. We propose using a data-driven optimization to automatically discover such strong perturbations. Second, our analysis suggests that we need the knowledge of the manifold of clean data to design strong poisoned images that are close to the manifold. Since the manifold is challenging to learn from data, we explicitly initialize the optimization near carefully selected clean images $\widetilde{\boldsymbol{x}}_{\mathrm{p}}$, allowing the optimization to easily control the size of the difference $\boldsymbol{\Delta}_{\mathrm{p}}$. We show in our ablation study in §2.5 that both components are critical for designing strong attacks.

E WHY ARE NNS SO VULNERABLE TO BACKDOOR ATTACKS?

NTBA showcases the vulnerability of DNNs to backdoor attacks. We investigate the cause of such vulnerability by comparing the infinite-width NTK with the standard Laplace kernel.

NTK gives more influence to far away data points. Recently, (Geifman et al., 2020; Chen & Xu, 2021) showed that the neural tangent kernel of feed-forward neural networks are equivalent to Laplace kernels $K^{\mathrm{lap}}(\boldsymbol{x}, \boldsymbol{y}) = \exp(\|\boldsymbol{x} - \boldsymbol{y}\|/\sigma)$ for inputs lying on the unit sphere. The Laplace kernel gives more influence to points that are closer. For example, Laplace-kernel linear regression converges to a 1-nearest neighbor predictor in the limit as the bandwidth $\sigma \to 0$, which is naturally robust against few-shot backdoor attacks. In contrast, we demonstrate that the NTK gives *more* influence to points as they become more distant. We confirm this by visualizing the two kernels with matched bandwidths in the normal and tangent direction to a unit sphere. In Figs. 12a and 12b, we consider the infinite width neural tangent kernel of a 3 layer feed-forward neural network with ReLU activations. For our choice of NTK, we compare against a Laplace kernel with $\sigma \approx 6.33$, that closely matches the NTK around $x = 0$ in Fig. 12b. For inputs that do not lie on the sphere, the kernels behave differently.

NTK is more vulnerable to few-shot backdoor attacks. We demonstrate with a toy example that NTK is more influenced by far away points, which causes it to be more vulnerable to some

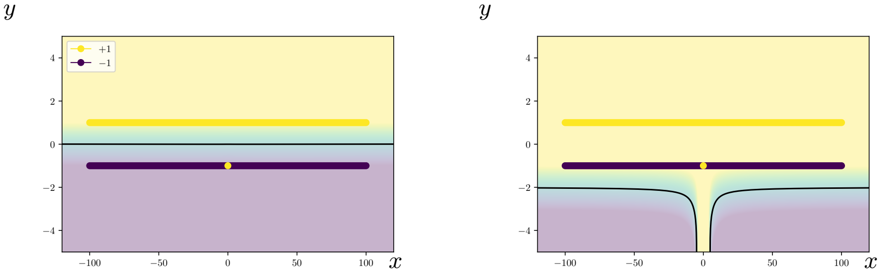


(a) Kernel behavior *normal* to unit sphere. The plot shows $K(\boldsymbol{e}_1, y\boldsymbol{e}_1)$ for both the NTK and Laplace kernels where $\boldsymbol{e}_1$ is a unit vector. Note that the NTK increases with y, while the Laplace kernel peaks at $y = 1$.

(b) Kernel behavior *tangent* to unit sphere. The plot shows $K(\boldsymbol{e}_1, \boldsymbol{e}_1 + x\boldsymbol{e}_2)$ for both the NTK and Laplace kernels where $\boldsymbol{e}_1, \boldsymbol{e}_2$ are orthogonal unit vectors. The two kernel behave similarly near $x = 0$ but diverge rapidly away from 0.

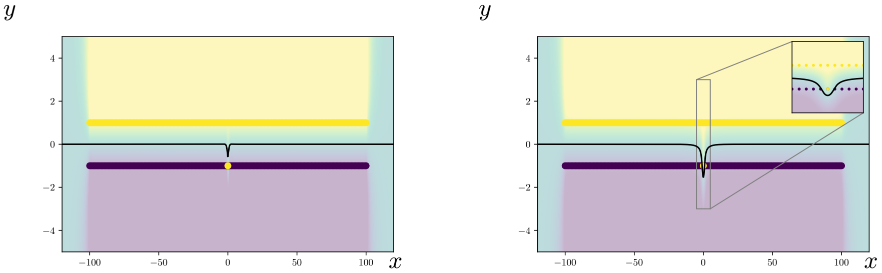
Figure 12: Kernel behavior off the unit sphere shows that the NTK approaches oblique asymptotes as either $|x|$ or y, increases, while the Laplace kernel decreases in the same limit.

few-shot backdoor attacks. We use a synthetic backdoor dataset in 3 dimensions (x, y, z) consisting of clean data $([\widetilde{x} \quad 1 \quad 0]^\top, 1)$ and $([\widetilde{x} \quad -1 \quad 0]^\top, -1)$ for $\widetilde{x} \in \{-100, -99, \dots, 100\}$. Here, the x dimension represents the diversity of the dataset, the y dimension represents the true separation between the two classes, and the z dimension is used to trigger the backdoor attack. We choose test-time trigger $P(\boldsymbol{v}) = \boldsymbol{v} + [0 \quad 0 \quad 1]^\top$ for a clean negative labelled point $\boldsymbol{v}$ and add a single train-time poison data point $(0, -1, \widetilde{z})$. For the Laplace kernel, we compute the best choice of $\widetilde{z}$ which is $\widetilde{z} = 1$. For the NTK, the backdoor increases in strength as $\widetilde{z} \to 0^+$ (we chose $\widetilde{z} = 1 \times 10^{-6}$).



(a) NTK, decision boundary at $z = 0$

(b) NTK, decision boundary at $z = 1$



(c) Laplace, decision boundary at $z = 0$

(d) Laplace, decision boundary at $z = 1$

Figure 13: The decision boundaries at $z = 0$ (black solid line) and corresponding predictions (background shading) on the $z = 0$ plane are similar for NTK and Laplace kernel, explaining the similar clean accuracy in Table 4. The decision boundary at $z = 1$ shows that the trigger fails to generalize to test examples for Laplace kernel. All points in the training dataset are shown regardless of their z-coordinate. Note that the solid bars are actually discrete points with overlapping markers and the yellow point at $(0, -1)$ is the single poison point.

In Fig. 13d we see that the backdoor is not successful for the Laplace kernel, only managing to flip the prediction of a single backdoor test point. This is because the influence of the poison point rapidly drops off as $|x|$ increases. For $|x| > 10$ the poison has a negligible effect on the predictions of the model. In contrast, we see in Fig. 13b that the NTK was successfully backdoored and the predictions of all test points can be flipped by the trigger $P(\cdot)$. This is due to the influence of the poison point remains high even from a great distance.

F EVASION OF BACKDOOR DEFENSES

We evaluate our attack against three defenses: SPECTRE (Hayase et al., 2021), the spectral signature defense of (Tran et al., 2018), and the activation clustering defense of (Chen et al., 2019). We give the detectors a fixed budget of $\lceil 1.5m \rceil$ points to remove following (Hayase et al., 2021; Tran et al., 2018) and report the fraction of poison examples remaining in the training set after filtering. We compare our NTBA attack and the sampling baseline on in two settings: CIFAR-10 trained from scratch with the periodic trigger as in §3.1 and fine-tuning ImageNet with the periodic trigger as in Appendix C.2. The results are shown in Table 7.

	NTBA	sampling
m	30	300
$\mathrm{asr}_{\mathrm{nn,te}}$	90.7%	89.3%
SPECTRE	96.7%	35.0%
Spectral Signatures	100.0%	62.0%
Activation Clustering	93.3%	70.3%

(a) Setting of Table 2

	NTBA	sampling
m	10	100
$\mathrm{asr}_{\mathrm{nn,te}}$	96%	100%
SPECTRE	0.0%	0.0%
Spectral Signatures	100.0%	0.0%
Activation Clustering	90.0%	0.0%

(b) Setting of Table 6b

Table 7: Percentage of poison examples remaining after filtering.

We see that the NTBA-designed attack is at least as difficult to detect as the sampling baseline in all cases. Additionally, we believe it should be possible to incorporate penalties into the backdoor loss that encourage the attack to evade defenses in the style of (Xia et al., 2022; Shokri et al., 2020; Qi et al., 2022; Xiong et al., 2020) but we leave this direction for future work.