3.2.3 Empirical Parameter Deviation Plots

After having investigated which factors influence backdoor effectiveness, we shift our focus to examining how the model's weights change during the training process when the dataset is tainted with backdoor samples. We aim to determine whether there is an increase in complexity or not.

We use our two measures proposed in Sect. 2, ρ and ν to analyze the parameter change. The former, ρ, monitors the change of the weights, for example, whether they increase or decrease. The latter, ν, measures the change in orientation or angle of the classifier. We plot both measures with different regularization parameters, trigger size, or visibility with a fraction of poisoning points to $p = 0.1$ in Figs. 9, 10 and 11. Within each plot, we also report the backdoor accuracy (BA) representing the model's performance on backdoor samples at the end of training.

On linear classifiers, $\rho(\boldsymbol{w})$ increases during the backdoor learning process. This equals an increase in the weights' values, suggesting that the classifiers become more complex while learning the backdoor. However, when investigating the RBF SVM, the results are slightly different. Indeed, when increasing γ and decreasing λ, the classifier becomes flexible and complex enough to learn the backdoor without increasing its complexity. On the other hand, when decreasing γ, the model is constrained to behave similarly to a linear classifier. In this way, analogously to linear classifiers, the model needs to increase its complexity to learn the backdoor. When increasing the trigger size or visibility the results are similar, thus confirming the previous analysis. However, as a result of increasing the attacker's strength, the backdoor accuracy turns out to be higher.

3.2.4 Explaining backdoor predictions

In the following, we give a graphical interpretation of the poisoned convex-classifier's decision function, expressed by its internal weights, for which interpretation of their results is easier [26, 27]. We consider the results for a backdoor trigger [1] in a specific position, as its influence on the classifier decision is easier to see. Conversely, the backdoor trigger by for example Zhong et al. [32] spans the entire image, and therefore its influence is harder to spot from the interpretability plots. In particular, given a sample x we aim to compute and show the gradient of the classifier's decision function with respect to x. We use an SVM with regularization λ = 1e−02 for MNIST 7 vs 1 and CIFAR10 *airplane vs frog*, and report the results in Fig. 12. For MNIST, we consider the digit 7 with the trigger, showcasing the gradient of the clean classifier's decision function. We present the results of the gradient from the clean and poisoned classifiers corresponding to the clean and backdoored inputs. Since we train a linear classifier on the input space, the derivative coincides with the classifier's weights. Intriguingly, the classifier's

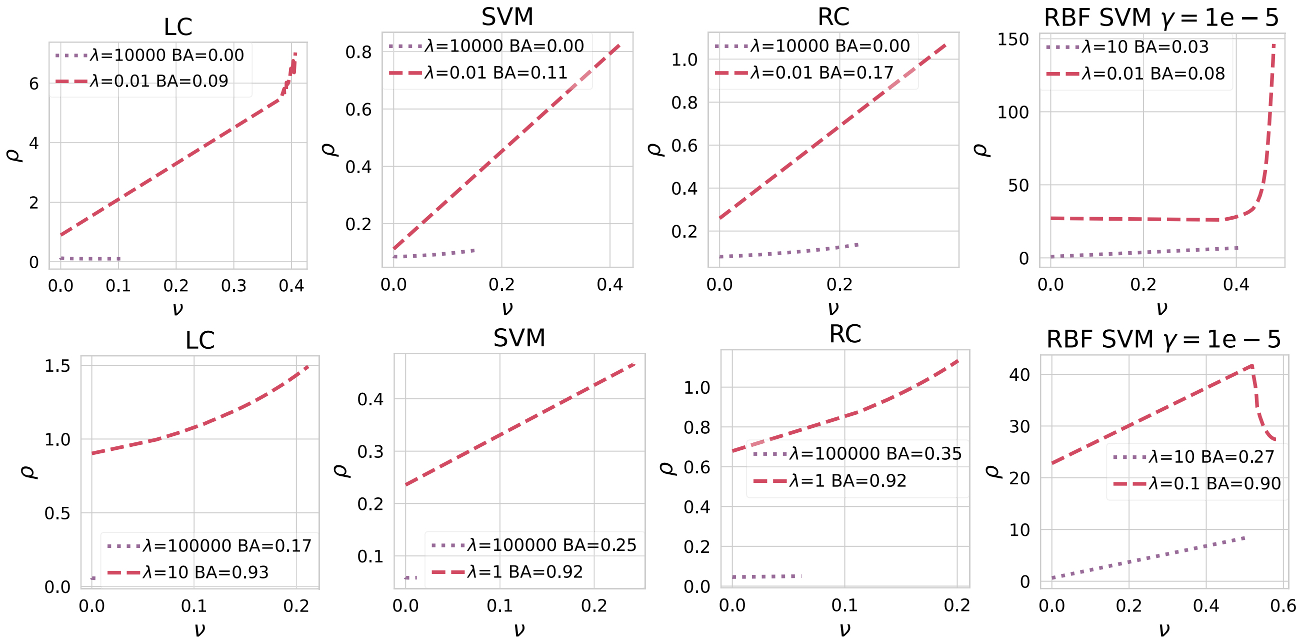

Fig. 11 Backdoor weights deviation for the logistic classifier (LC), support vector machine (SVM), the ridge classifier (RC) and SVM with RBF kernel on Imagenette *tench vs truck* poisoned with backdoor trigger [32]. We report the results for visibility $c_m = 10$ (top row) and $c_m = 75$ (bottom row). We specify the regularization parameter λ and backdoor accuracy (BA) for each setting in the legend of each plot

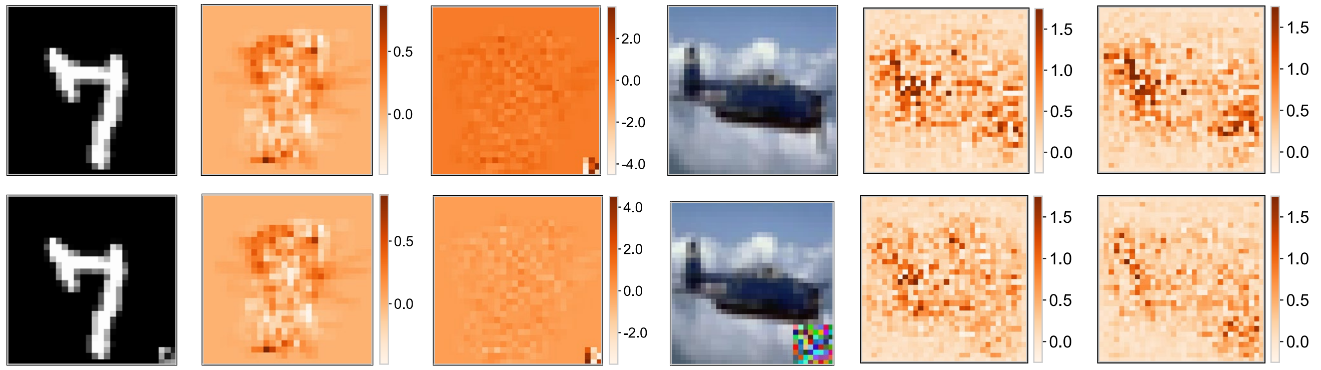

Fig. 12 Input gradients of untainted and poisoned SVMs on pristine (top row) and backdoored (bottom row) test samples. Each row shows two sets of three images. Each set contains an example from MNIST *7 vs 1* or CIFAR10 *airplane vs frog* (left), along with the corresponding input gradient of the untainted SVM (middle), and of the poisoned SVM (right). For CIFAR10, we consider the maximum gradient of each pixel among the three channels

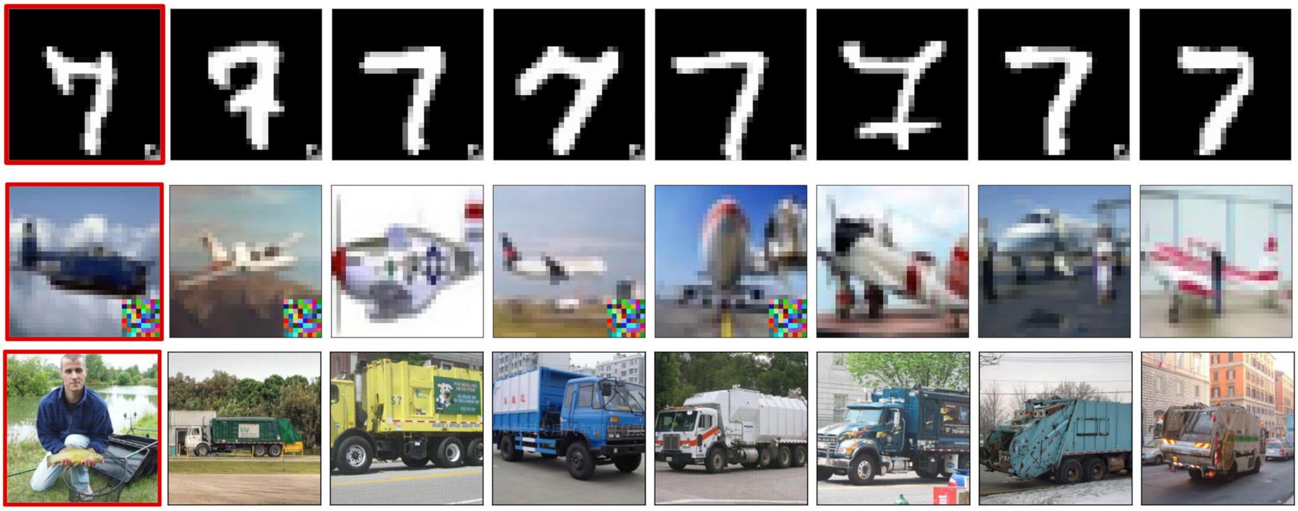

Fig. 13 Influential training points for a high-complexity classifier. Considering an SVM with $\lambda = 0.01$ trained on MNIST, and with $\lambda = 0.1$ trained on CIFAR10, and Imagenette, we show the top 7 most influential training samples on the prediction of the samples with the red border

weights increase in magnitude and now exhibit high values in the bottom right corner, where the trigger is located. From CIFAR10, we show a poisoned airplane. We report the gradient mask obtained by considering the maximum value for each channel, both for the clean and backdoored classifier. Also, in this case, the backdoored model shows higher values in the bottom right region, corresponding to the trigger location. This means that the analyzed classifiers assign high importance to the trigger to discriminate the class of the input points.

Summarizing, the plots in Fig. 12 further confirm our findings regarding the change of the internal parameters during the backdoor learning process. In particular, we have seen that less regularized classifiers need to increase their weights and thus complexity to learn the backdoor. Conversely, when the flexibility of the classifier increases then it can learn the backdoor easier without significantly altering its complexity.

3.2.5 Visualizing influential training data points

Influence functions are used in the context of ML to identify the training points more responsible for a given prediction [13]. In Sect. 2 we have seen how they represent the basis of our backdoor learning slope measure. In this section, we employ them to show their outcomes and provide further insight into the relationship between complexity and backdoor effectiveness. To this end, as in Sect. 3.1, we poison 10% of the training dataset. According to previous experiments, we employed the backdoor trigger in [1] for MNIST and CIFAR10 with trigger size 3×3 and 6×6 respectively, while for Imagenette we employed the trigger in Zhong et al.

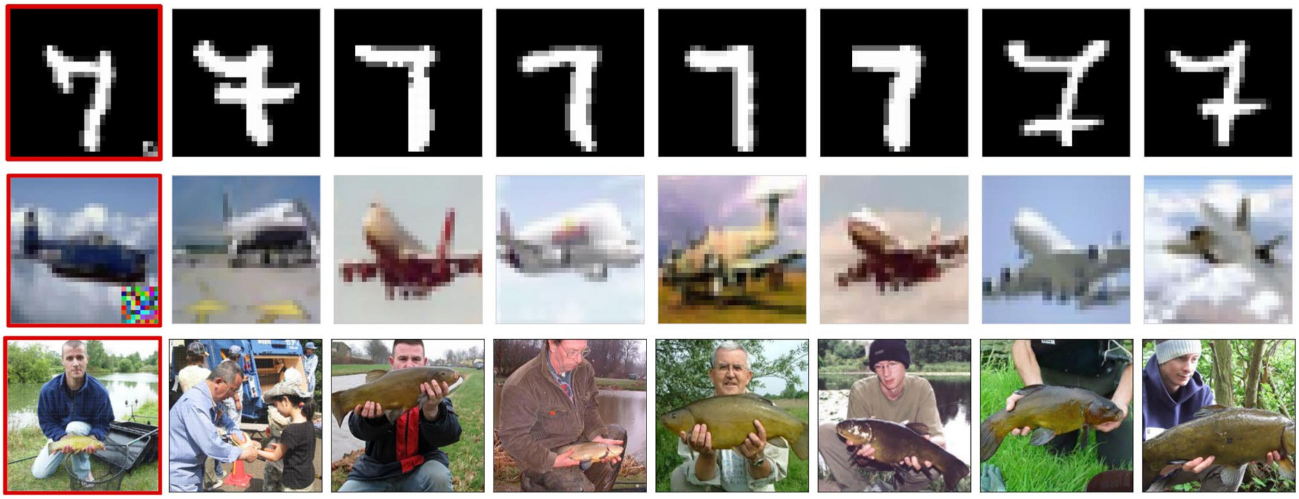

Fig. 14 Influential training points for low-complexity classifiers. Considering an SVM with $\lambda = 1e-3$ trained on MNIST, and with $\lambda = 1e-5$ trained on CIFAR10, and Imagenette, we show the top 7 most influential training samples on the prediction of the samples with the red border

[32] with higher visibility (i.e. $c_m = 75$). In Figs. 13 and 14, considering respectively a high- and a low-complexity classifier, we report the seven most influential training samples on the classification of a randomly chosen test point. For high-complexity classifiers, many of these training samples contain the trigger. In contrast, this is not the case for low-complexity classifiers. These results suggest that low-complexity classifiers rely less on the samples containing the backdoor trigger in their predictions.

4 Related work

We first review the literature about backdoor poisoning attacks and defenses. Afterward, we focus on defenses that increase the robustness against backdoors by reducing the model's complexity. We conclude the section by discussing the relationship between our proposed framework and influence functions.

Backdoor poisoning. Although backdoors were introduced recently [1, 3, 6], a plethora of backdoor attacks and defenses have been published. For a more detailed overview, we refer the reader to surveys in this area [3, 4, 36]. Despite the quickly-growing literature about this topic, the majority of the previous works [21, 33, 37, 38] study different types of poisoning attacks, i.e., not backdoors. In contrast, only a few works have studied factors that influence the success of this attack. Baluta et al. [39] and [40] studied the relationship between backdoor effectiveness and the percentage of backdoored samples. Salem et al. [41] experimentally investigated the relationship between the backdoor effectiveness and the trigger size. Similarly, Severi et al. [42] have analyzed the correlation between the backdoor success and the attacker's strength on malware classifiers. Schwarzschild et el. [43] evaluated the performance of backdoor attacks when scaling the dataset size while fixing the poison budget. Finally, Li et al. [44] demonstrated that the backdoor performance is sensitive to the location of the trigger on the attacked image. We instead do not limit our study to neural networks but also study other models. Furthermore, we also investigate other relevant factors, e.g., regularization and visibility, and their interaction at once.

Complexity and backdoor defenses. In this work, we have analyzed the relationship between backdoor effectiveness and different factors, including complexity, controlled via regularization and the RBF kernel's hyperparameter. In this study, we have demonstrated that reducing complexity by choosing appropriate hyperparameter values improves robustness against backdoors. Our findings align with the insights presented in Frnay et al. [45], who suggested that overfitting avoidance techniques like, e.g., regularization, can offer partial mitigation against random label noise [46, 47]. Expanding upon their discourse, we apply and extend this consideration to the context of backdoor attacks, wherein the noise is intentionally and strategically introduced to deceive the machine learning model. Some of the defenses proposed against backdoors use different techniques to reduce complexity. These techniques include pruning [48, 49], data augmentation [50, 51] and gradient shaping [52]. However, from these works, it remains unclear why reducing complexity alleviates the threat of backdoor poisoning. To the best of our knowledge, our work is the first to investigate this aspect.

Relation to influence functions. Influence functions originated in robust statistics [53] and were later used as a tool to measure the influence of specific training points on the classification output [13, 54]. In our work, we clarify that influence functions naturally descend from the incremental learning formulation in Eq. 1, showing that they quantify the velocity with which the classifier will learn new points. As seen in Sect. 2, they correspond to the partial derivative of the learning curve at the point $\beta = 0$. Moreover, we leveraged them by proposing a measure, namely the backdoor slope, which quantifies the ability of a classifier to learn backdoors. This measure allowed us to study the factors that impact backdoor effectiveness.

Several defense approaches confirm that the influence functions, or gradients during training, are indeed related to backdoor learning. For example, some defenses are directly based on the gradient [55], based on gradient differences [56, 57], or based on differential privacy that noises the gradients during training [52, 58, 59].

5 Conclusions, limitations and future work

In this paper, we presented a framework to analyze the factors influencing the effectiveness of backdoor poisoning. We carried out experiments on convex learners, also used in transfer-learning scenarios, and neural networks. As in previous work [7, 13], we focus our analysis on two-class classification problems for convex learners, and on multiclass classification when considering neural networks.

Our analysis shows that the effectiveness of backdoor attacks inherently depends on (i) the complexity of the target model, (ii) the fraction of backdoor samples in the training set, and (iii) the size and visibility of the backdoor trigger. By analyzing the influence of the first factor on backdoor learning, we are the first to unveil a region in the hyperparameter space where the accuracy on clean test samples remains high while the accuracy on backdoor samples is low. Specifically, we discovered that the target model needs to significantly increase the complexity of its decision function to learn backdoors, which is only possible when the model is not regularized enough. Conversely, when raising the model's regularization, we can keep high performance on clean samples and be unaffected by potential backdoor attacks. However, increasing the attacker's strength, i.e., the last two factors, makes the attack more effective, shrinking this region and thus exposing the model to greater vulnerability. We, therefore, conclude that a prudent strategy to preserve robustness against potential poisoning attacks is to regularize as much as possible during the hyperparameter optimization phase, thereby reducing the backdoor learning slope while ensuring that the trade-off with accuracy remains acceptable.

The study of more factors, like, for example, the dimensionality of the data, is straightforward using the proposed framework but left for future work. Our current results already provide important insights and provide a starting point to derive guidelines for designing models that are more robust against backdoor poisoning.

Appendix A: Datasets

The MNIST dataset [18] contains 70,000 observations representing 28×28 grayscale images of handwritten digits from 0 to 9. The CIFAR10 dataset [19] contains 60,000 colour images of size 32×32 pixels divided in 10 classes, each with 6000 observations. Finally, the Imagenette dataset [20] is a subset of 10 classes (i.e., tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, parachute) from Imagenet. We use the 320px version, where the shortest side of each image is resized to that size.

Appendix B: Additional Experimental Results

In the paper, we have shown the backdoor learning curves only for some classifiers. Here, we report them for all the classifiers considered in this work. As we will discuss later in this section, these results confirm the ones obtained in the paper. In particular, here we consider:

- Support vector machine (SVM) with $\lambda \in \{100, 0.1\}$ for MNIST, $\lambda \in \{10000, 0.1\}$ for CIFAR10, and $\lambda \in \{100000, 1\}$ for Imagenette.
- Ridge classifier (RC) with $\lambda \in \{1000, 1\}$ for MNIST, $\lambda \in \{10000, 1\}$ for CIFAR10, and $\lambda \in \{100000, 1\}$ for Imagenette.
- Logistic classifier (LC) with $\lambda \in \{10, 0.01\}$ for MNIST, $\lambda \in \{10000, 100\}$ for CIFAR10, and $\lambda \in \{100000, 10\}$ for Imagenette.
- SVM with an RBF kernel, where $\lambda \in \{1, 0.01\}$ and $\gamma = 5\text{e}{-}04$ for MNIST, $\lambda \in \{100, 1\}$ and $\gamma = 1\text{e}{-}03$ for CIFAR10, and $\lambda \in \{10, 0.1\}$ and $\gamma = 1\text{e}{-}05$ for Imagenette.

Moreover, we compare the results obtained on the class pairs considered in the paper (7 vs 1 on MNIST, *airplane vs frog* on CIFAR10 and Imagenette *tench vs truck*) with the ones obtained on different pairs.

Backdoor learning curves and backdoor learning slope. In Figs. 15, 16, 17, 18, 19 and 20 we report the backdoor learning curves for each classifier and dataset pair. In Figs. 21, 22 and 23, we report the backdoor learning slope, computed with $p = 0.1$, for all the considered classifiers and all subset pairs. The results do not show significant variation with respect to the ones reported in the paper.

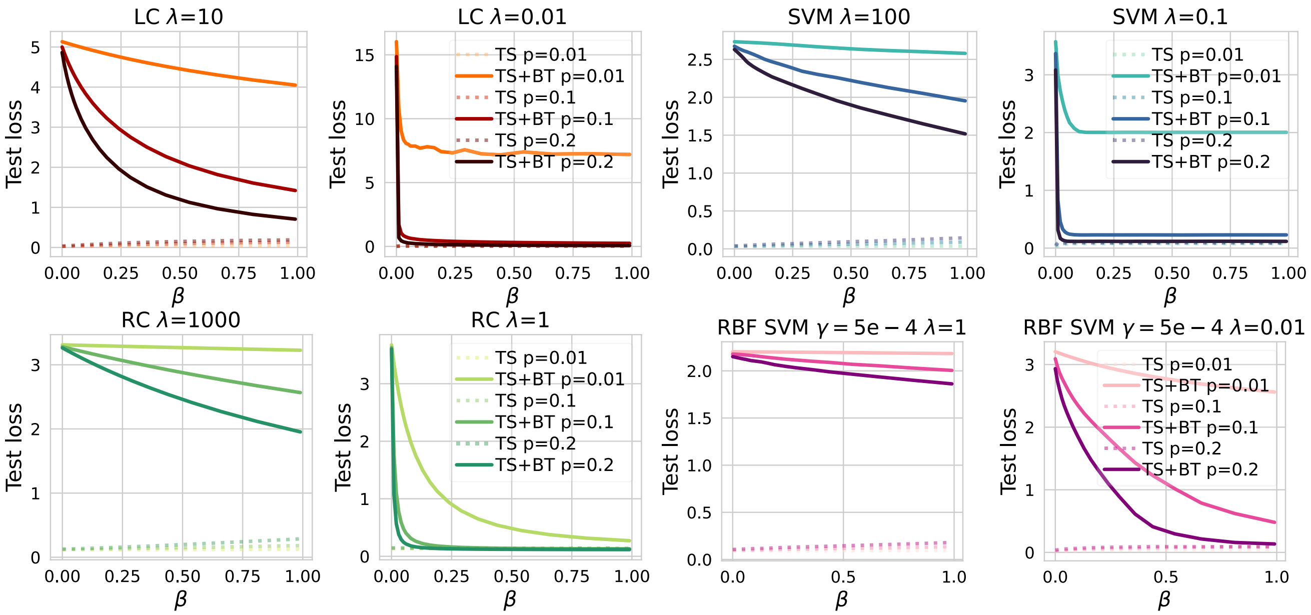

Fig. 15 Backdoor learning curves for different classifiers trained on MNIST 3-0. Darker lines represent a higher fraction of poisoning samples p injected into the training set. We report the loss on the clean test samples (TS) with a dashed line and on the test samples with the backdoor trigger (TS+BT) with a solid line

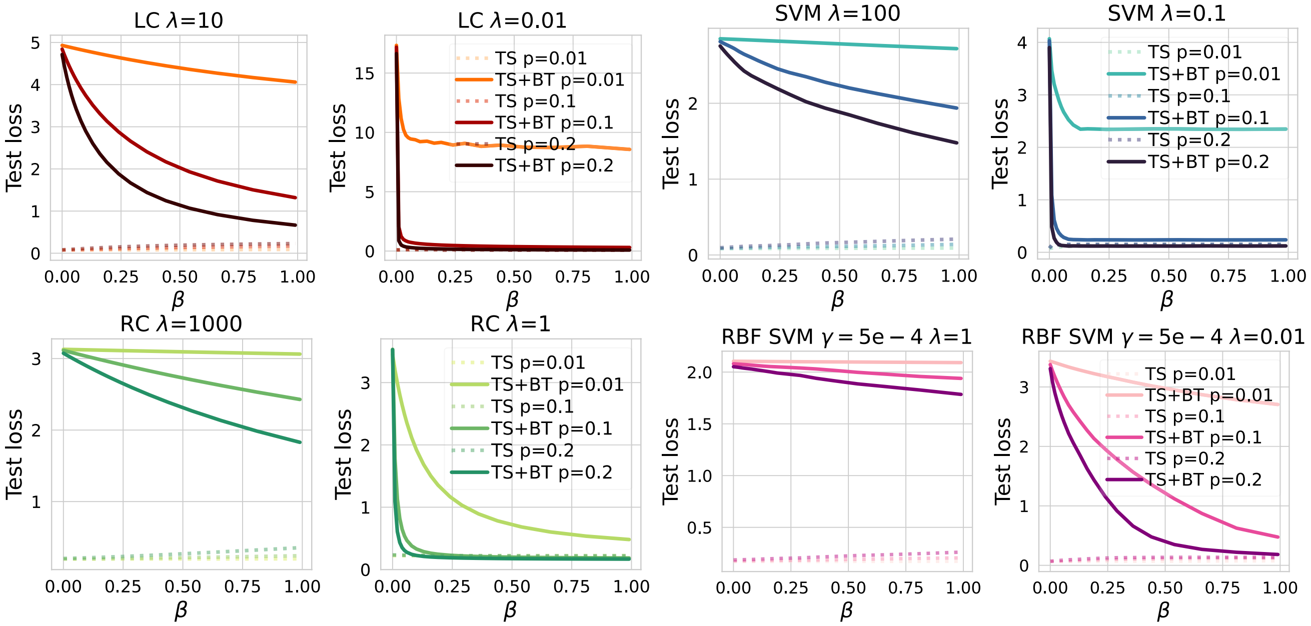

Fig. 16 Backdoor learning curves for different classifiers trained on MNIST 5-2. See the caption of Fig. 15 for further details

Empirical parameter deviation plots. In Figs. 24, 25 and 26, shows how the classifiers' parameters change when the classifiers learn the backdoors. This analysis is carried out with $p = 0.1$. The results do not vary significantly across different classifiers and class pairs. The only exception is MNIST 5 vs 2. The untainted classifier is already quite complex; therefore, it does not increase its complexity when it learns the backdoor.

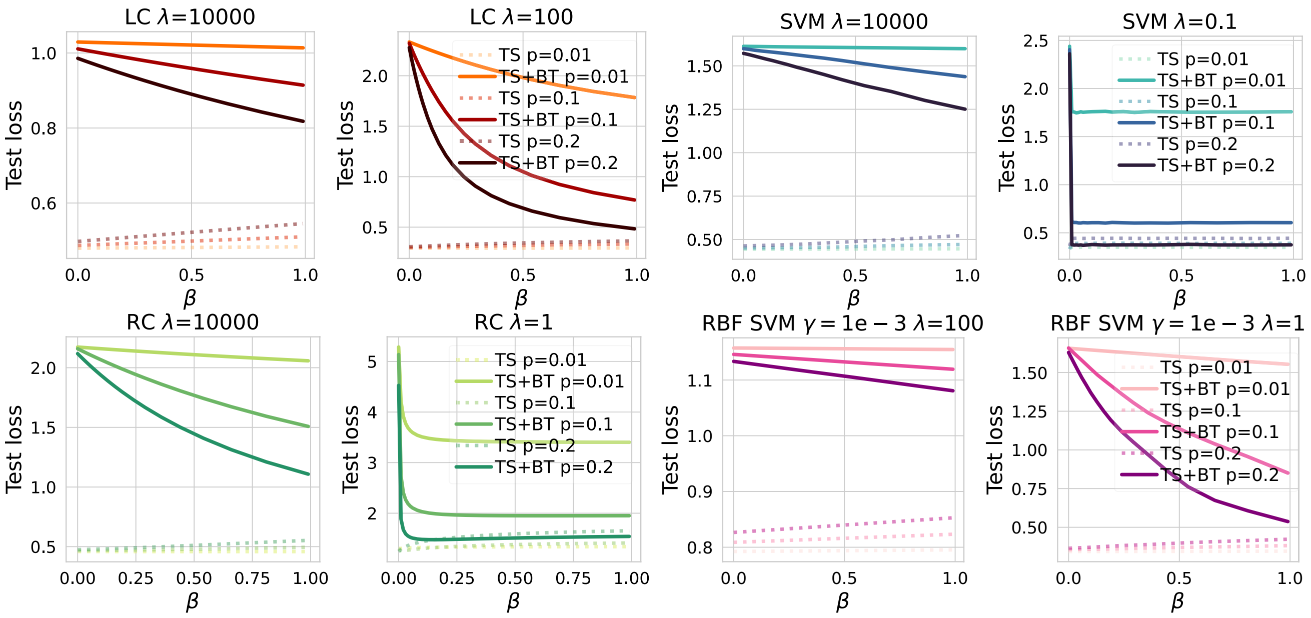

Fig. 17 Backdoor learning curves for different classifiers trained on CIFAR10 *bird vs dog*. See the caption of Fig. 15 for further details

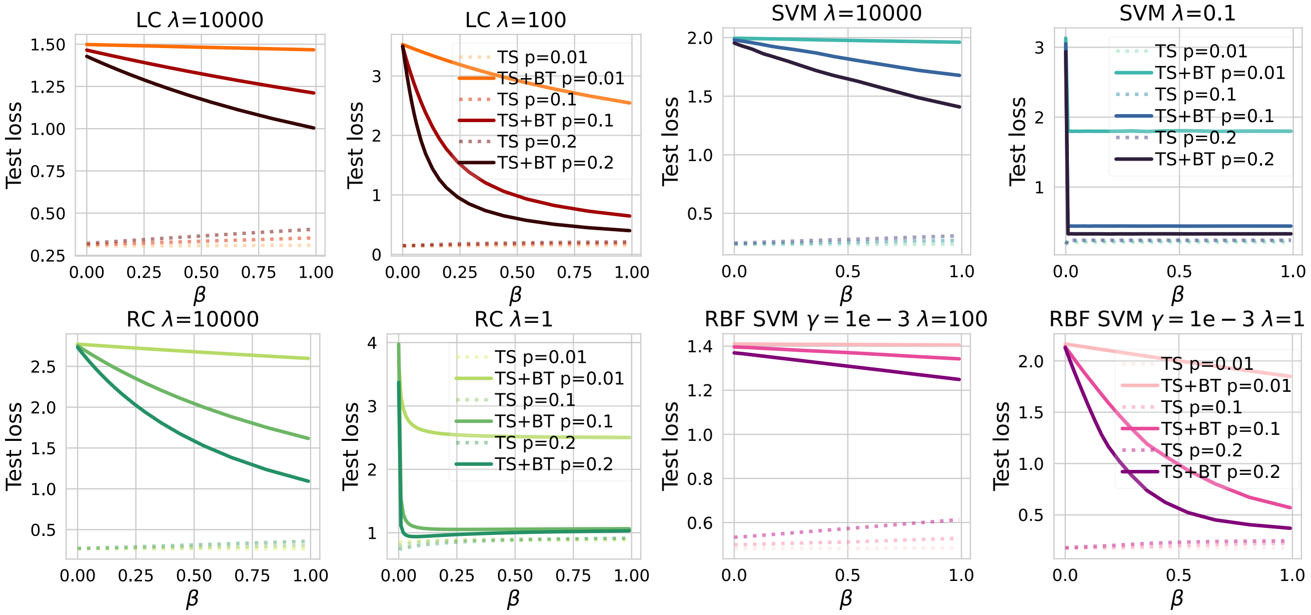

Fig. 18 Backdoor learning curves for different classifiers trained on CIFAR10 *airplane vs truck*. See the caption of Fig. 15 for further details

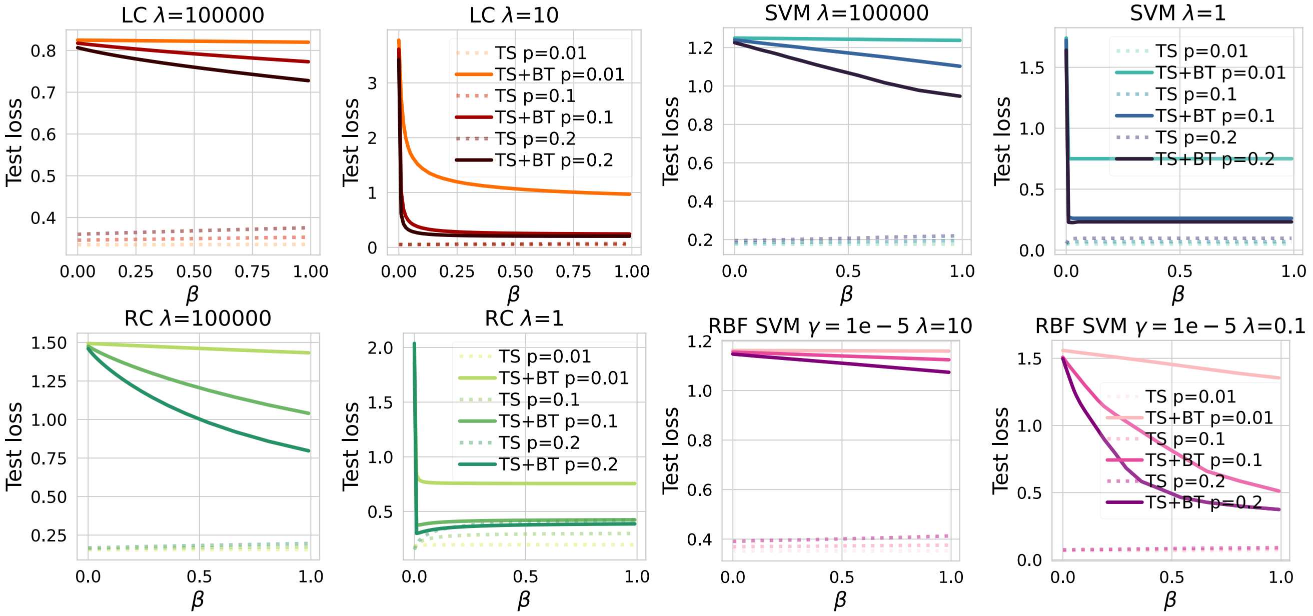

Fig. 19 Backdoor learning curves for different classifiers trained on Imagenette *cassette player vs church*. See the caption of Fig. 15 for further details

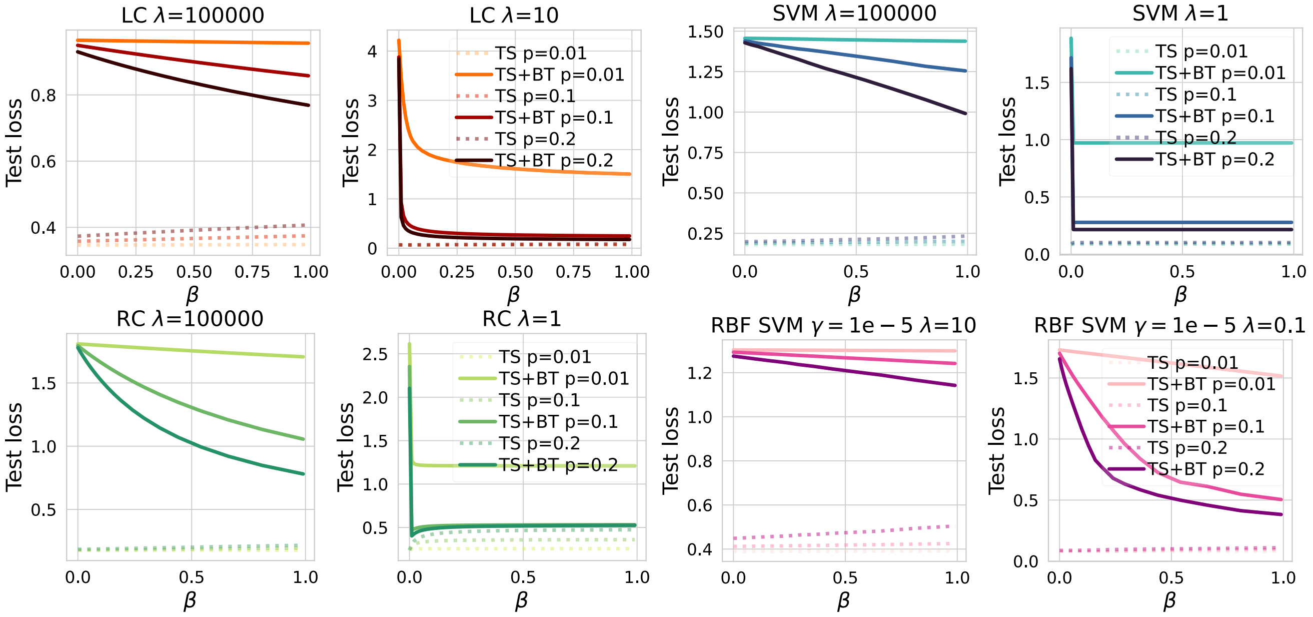

Fig. 20 Backdoor learning curves for different classifiers trained on Imagenette *tench vs parachute*. See the caption of Fig. 15 for further details