Supplementary Material for the ICMLA Paper: Finite Difference Neural Networks: Fast Prediction of Partial Differential Equations

Zheng Shi*†¶, Nur Sila Gulgec‡¶, Albert S. Berahas§¶, Shamim N. Pakzad*‖ and Martin Takáč*¶
*Department of Industrial & Systems Engineering, Lehigh University, Bethlehem, USA.
†IBM, Armonk, USA. ‡Thornton Tomasetti, San Francisco, USA.
§Department of Industrial & Operations Engineering, University of Michigan, Ann Arbor, USA.
Email: ¶{shi.zheng.tfls, sgulgec, albertberahas, takac.mt}@gmail.com, ‖pakzad@lehigh.edu.

I. Detailed Description of Experiments

A. Data Generation

The 200 ICs were generated randomly with 200 distinct random seeds. We generated the data set of the solution for the stable case and used it as our base data for the noisy and unstable cases. Specifically, the noisy data set was formed by adding the multiplicative noises to the base data; the unstable data set was formed by extracting the data of $t \in \{0, 200, 400, 600, 800, 1000\}$ from the base data. For the forcing case, we used a randomly generated forcing function to create the data set of solution.

B. Supplementary Testing Procedures

In addition to the 1000-step prediction studied in the main paper, we adopted two supplementary testing procedures, i.e., the one-step and multi-step predictions, to evaluate the networks' performance of making short-term predictions.

Given a Δt, let $\tau' \in \mathbb{N}^+$ s.t. $\tau' \Delta t \leq T$ and consider a generalized testing data set

$$A_{\text{test}_{\tau'}} = \{(u_s(x,t), u_s(x,t+\tau'\Delta t)) \mid s \in S_{\text{test}},\ x \in \{0, \Delta x, ..., L\},\ t \in \{0, \Delta t, ..., T - \tau'\Delta t\}\},$$

where S_{test} is the index set of ICs for the testing purposes, $(u_s(x,t), u_s(x,t+\tau'\Delta t))$ is a testing sample, $u_s(x,t)$ is the input to the network, and $u_s(x,t+\tau'\Delta t)$ is the target.

The one-step prediction procedure is consistent with the training procedure, and we used $\tau' = 1$ for all cases. To define the multi-step prediction, we let $\tau' = 10$ for the stable, noisy and forcing cases and $\tau' = 3$ for the unstable case. (Note that when $\tau' = 5$ for the unstable case or $\tau' = 1000$ for one of the others, we have the data set for the 1000-step prediction.)

Accordingly, we defined the testing error as MSE in a generalized form

$$\text{MSE}_{\text{test}_{\tau'}} = \frac{1}{|A_{\tau'}|} \sum_{s,x,t} \left(u_s(x,t+\tau'\Delta t) - \tilde{u}_s(x,t+\tau'\Delta t)\right)^2, \tag{1}$$

where $\tilde{u}_s(x,t+\tau'\Delta t)$ denotes a prediction made by the network.

C. Implementation and Design of Experiments

We implemented the networks in Python with PyTorch and trained the networks on an *NVIDIA K80* GPU. For the optimization methods, we implemented the Trust-Region (TR) Newton CG method in Python and used the ADAM optimizer of PyTorch.

The primary goal of the experiments conducted in this paper is to evaluate the training and testing performance of the networks. To this end, we chose different configurations (see Table I) and adopted three testing procedures for each case. And, for each configuration, we trained the network with the TR method and ADAM with learning rates 10^{-3} and 10^{-4}, and used 10 distinct random seeds to initialize the network parameters and to select the stochastic mini-batches. We constrained the training budget of TR to be 100 iterations for the stable, noisy and forcing cases and 300 iterations for the unstable case. And, we allowed the ADAM algorithms to run for 12000 iterations regardless of a case.

TABLE I: Network Configurations.

Hyper-Parameter	Unstable Case	Others†
# FD-Blocks	$1, 2, 3, 4, 6, 8, 10$	$1, 2, 3, 4$
# FD-Filters	$2, 4, 8, 16$	$2, 4, 8, 16$

† includes the stable, noisy and forcing cases.

To study the effects of the hyper-parameters, i.e., the number of `FD-Blocks` and `FD-Filters`, and to validate our design of the architecture, we conducted further experiments with different numbers of `FD-Blocks` against the unstable case and with different numbers of `FD-Filters` against the stable case.

In the following sections, we will present the full experimental results in the order of the stable case, the unstable case, the forcing case and the noisy case.

II. Stable Case

In this section, we present the experimental results of the stable case. Fig. 1 shows the evolution of the training errors of different configurations. Fig. 2 shows the sequential predictions and the squared errors, and Fig. 3 shows the minimum testing errors (over the training process) of the 1000-step predictions by configuration. In addition, Fig. 4 shows the relationship between the training and testing errors (note: the lower and to the left is better). For the two supplementary testing procedures described in Section I-B, given the data of solution at t, the one-step prediction is made at $t+\Delta t$, i.e., $t+1$, and the multi-step prediction is made at $t+10\Delta t$, i.e., $t+10$. Fig. 5 & 6 show the evolutions of the testing errors (1) and Fig. 7 & 8 show the minimum testing errors of the one- and multi-step predictions. To summarize the testing performance, we put the minimum testing errors aggregated over all configurations in Fig. 9. Besides, the results of the sensitivity analysis on different numbers of `FD-Filters` (and 1 `FD-Block`) are shown in Fig. 10, which shows the evolution and the minimum (over the training process) of the training and testing errors.

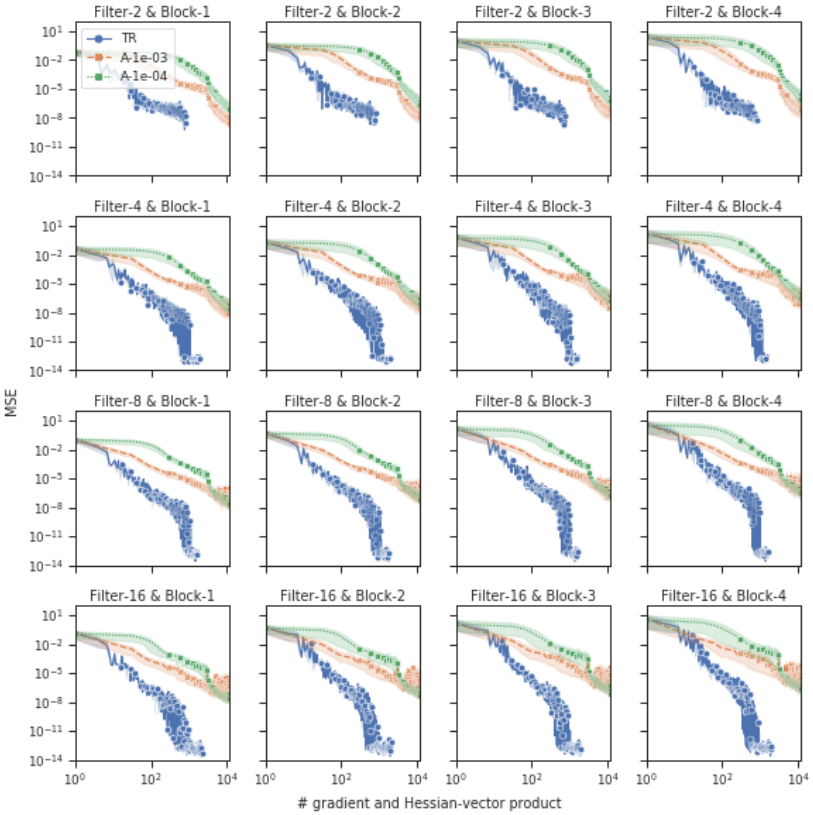


Fig. 1: **Stable Case:** Evolution of the MSE loss of stochastic mini-batch by configuration.

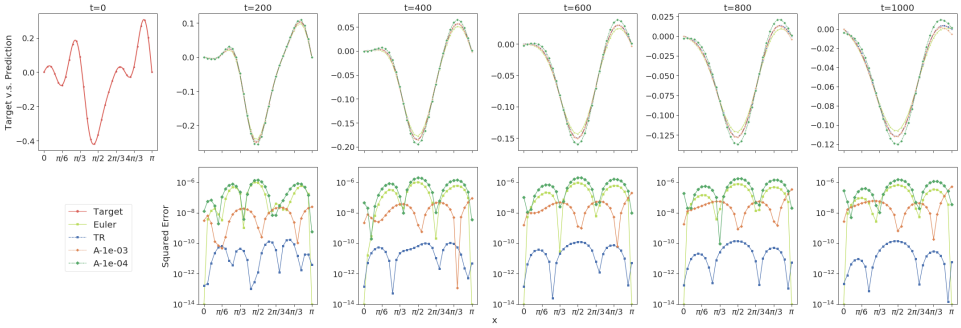


Fig. 2: **Stable Case:** Sequence of predictions.

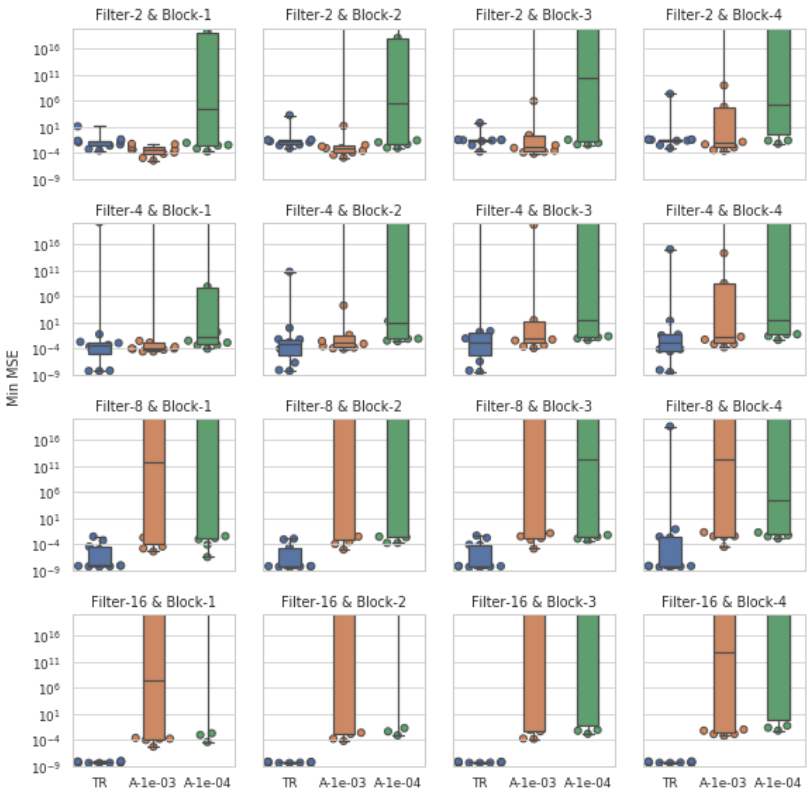


Fig. 3: **Stable Case:** Minimum testing errors by configuration - 1000-step prediction.

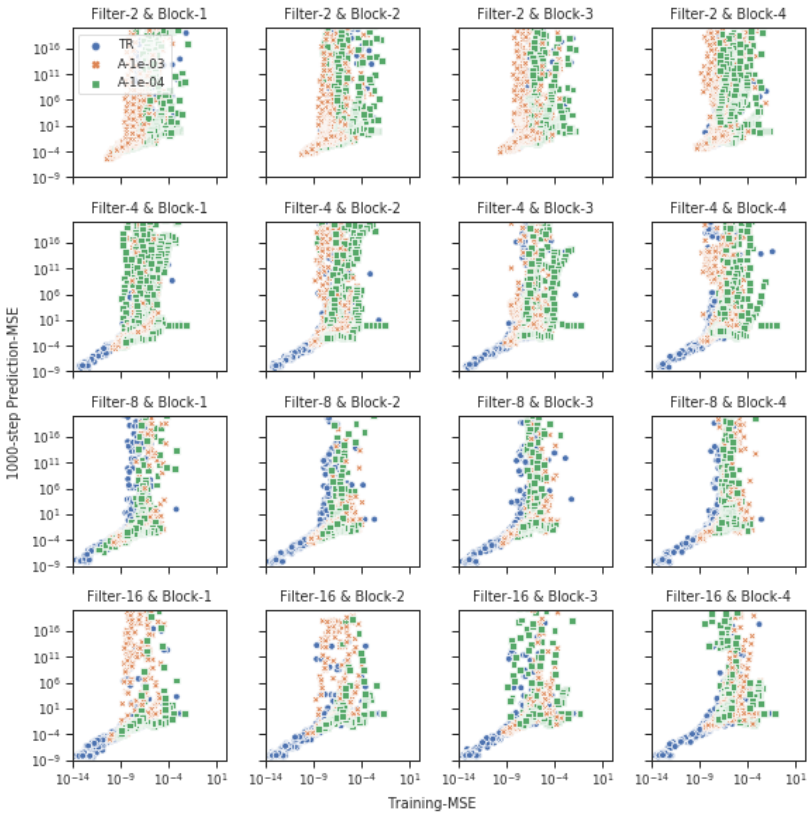


Fig. 4: **Stable Case:** Training vs. 1000-step prediction errors (over the training process) by configuration.

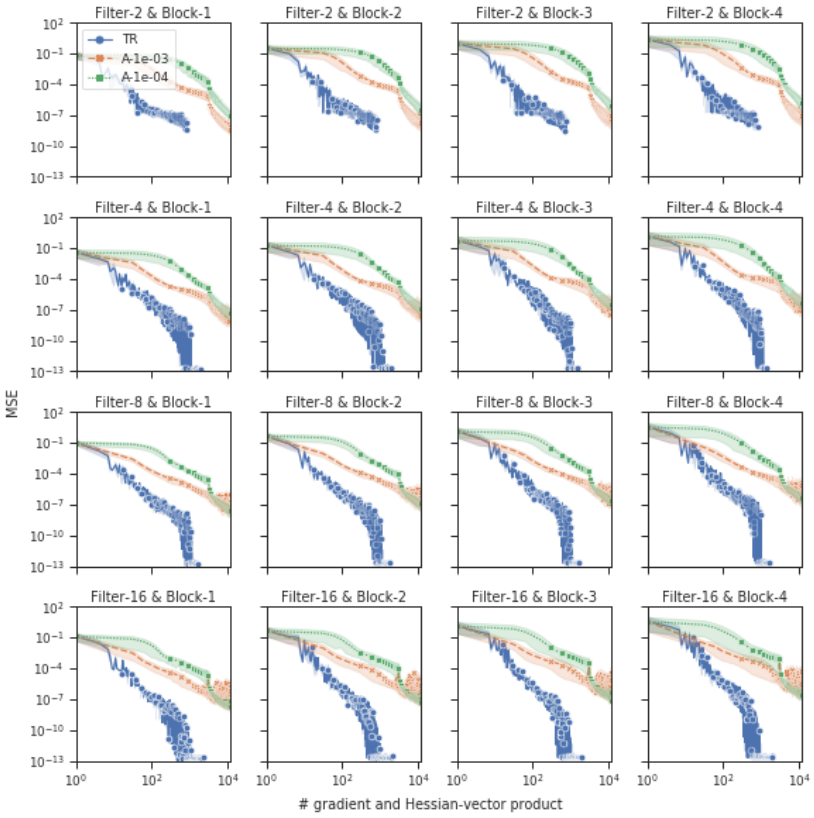


Fig. 5: **Stable Case:** Evolution of the testing errors by configuration - one-step prediction.

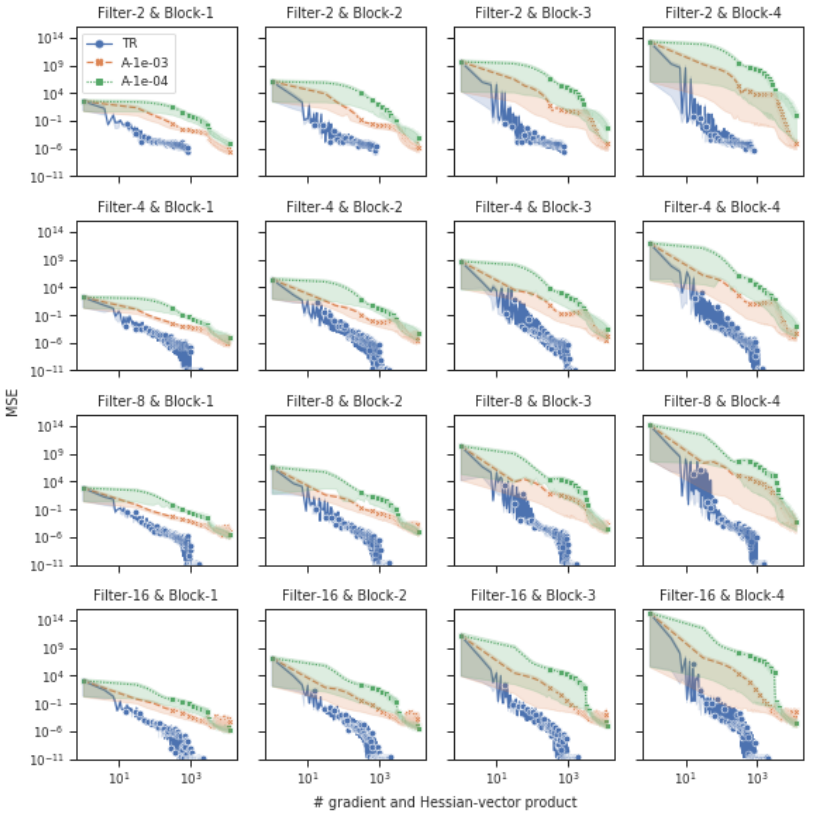


Fig. 6: **Stable Case:** Evolution of the testing errors by configuration - multi-step prediction.

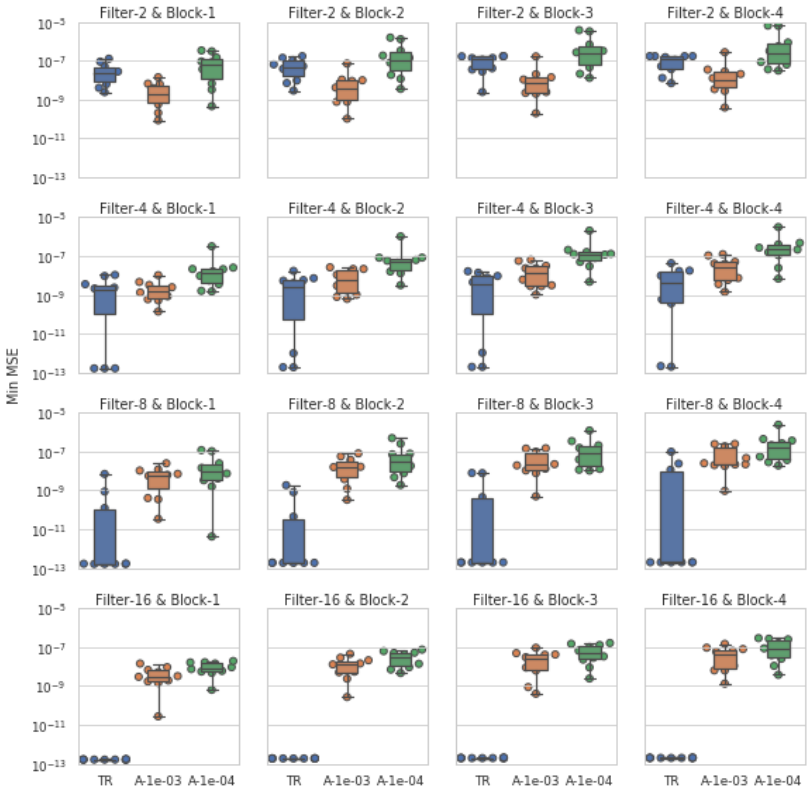


Fig. 7: **Stable Case:** Minimum testing errors by configuration - one-step prediction.

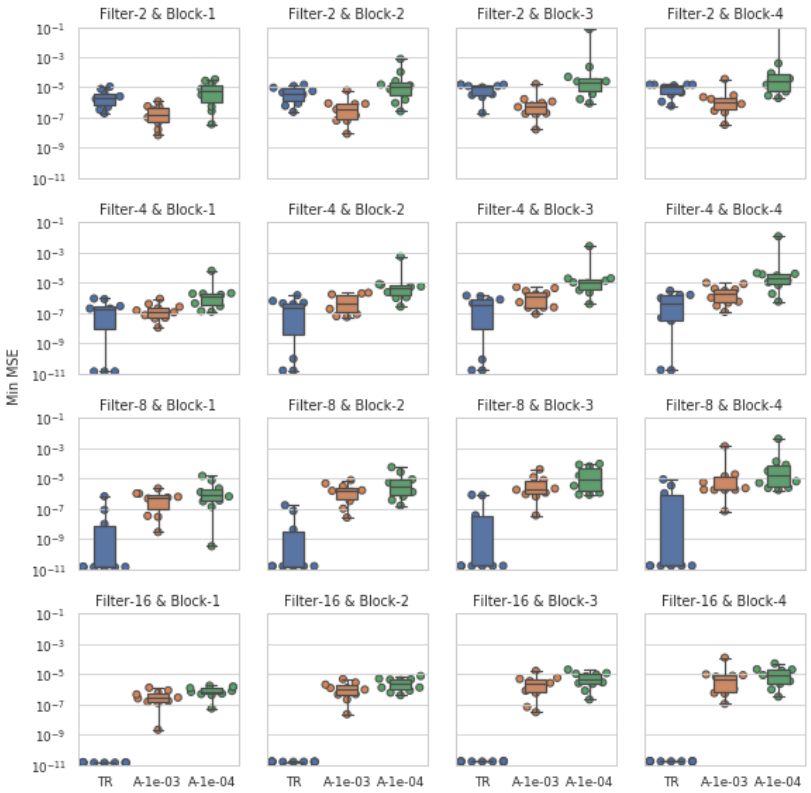


Fig. 8: **Stable Case:** Minimum testing errors by configuration - multi-step prediction.

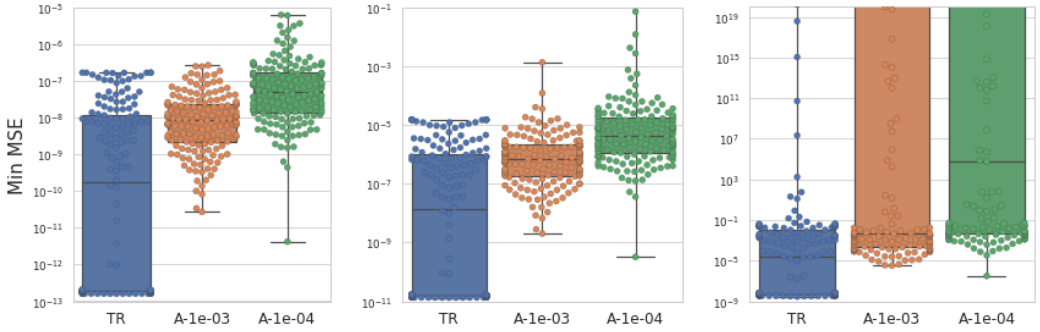


Fig. 9: **Stable Case:** Minimum testing errors of the one-step (left), multi-step (middle) and 1000-step (right) predictions (aggregated over all configurations).

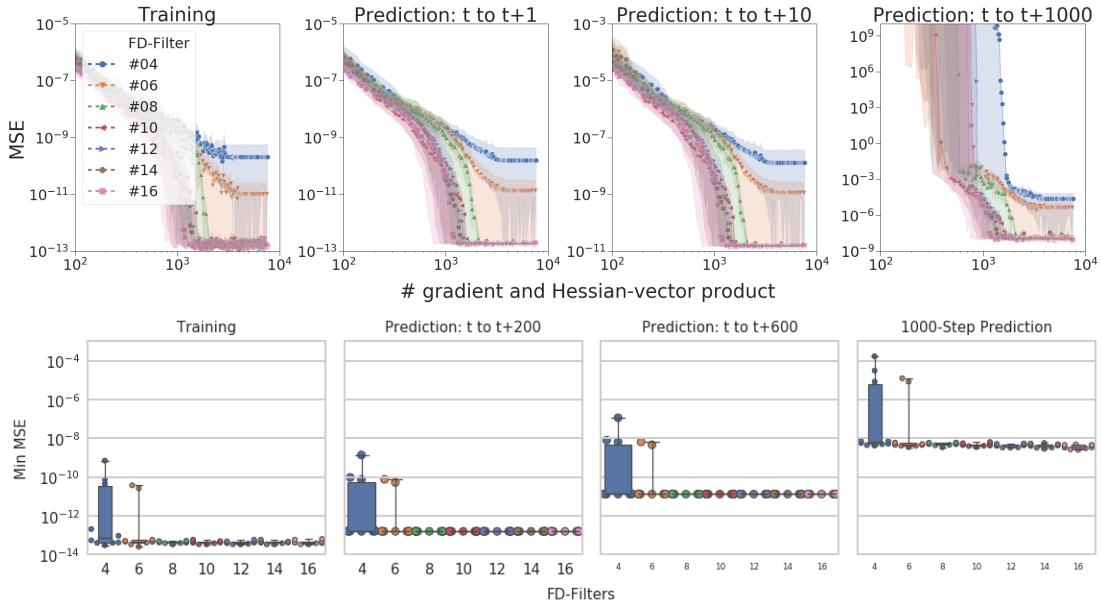


Fig. 10: **Stable Case:** Evolution (first row) and minimum (second row) of the training errors and testing errors of the one-, multi- and 1000-step predictions by configuration of `FD-Filters`.

III. Unstable Case

In this section, we present the experimental results of the unstable case. Fig. 11 shows the evolution of the training errors of different configurations. Fig. 12 shows the sequential predictions and the squared errors, and Fig. 13 shows the minimum testing errors (over the training process) of the 1000-step predictions by configuration. In addition, Fig. 14 shows the relationship between the training and testing errors (note: the lower and to the left is better). For the two supplementary testing procedures described in Section I-B, given the data of solution at t, the one-step prediction is made at $t + \Delta t$, i.e., $t + 200$, and the multi-step prediction is made at $t + 3\Delta t$, i.e., $t + 600$. Fig. 15 & 16 show the evolutions of the testing errors (1) and Fig. 17 & 18 show the minimum testing errors of the one- and multi-step predictions. To summarize the testing performance, we put the minimum testing errors aggregated over all configurations in Fig. 19. Besides, the results of the sensitivity analysis on different numbers of `FD-Blocks` (and 16 `FD-Filters`) are shown in Fig. 20, which shows the evolution and the minimum (over the training process) of the training and testing errors.

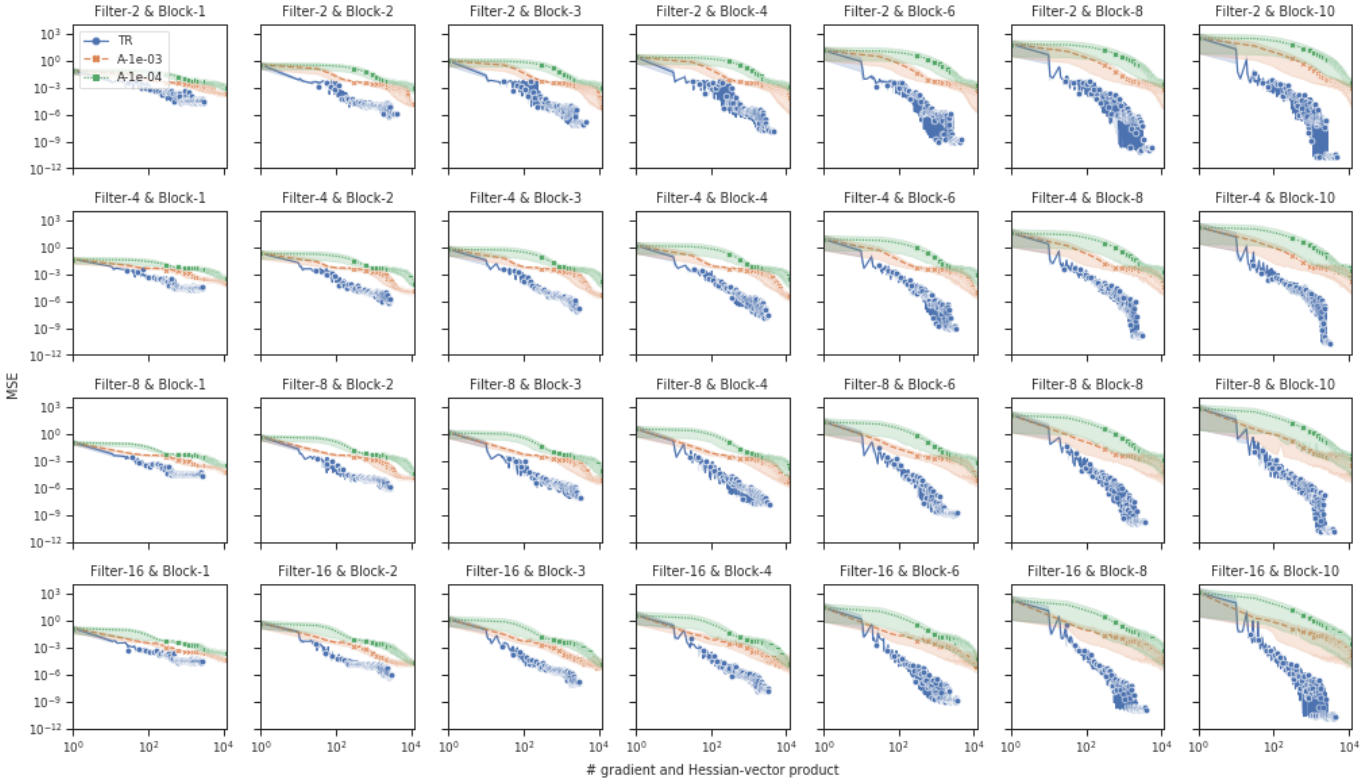


Fig. 11: **Unstable Case:** Evolution of the MSE loss of stochastic mini-batch by configuration.

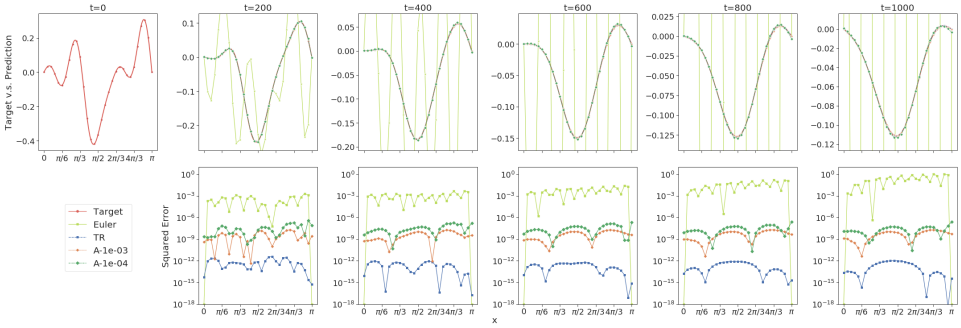


Fig. 12: **Unstable Case:** Sequence of predictions.

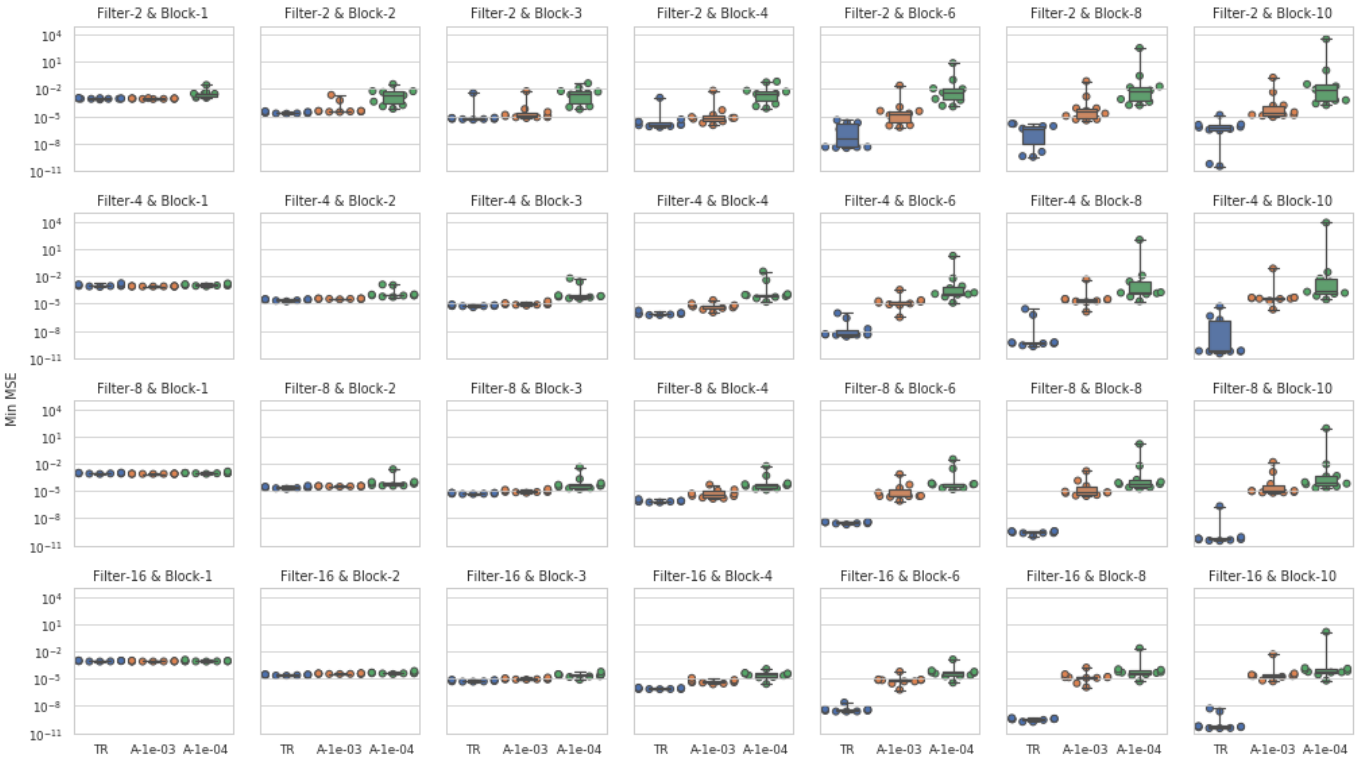


Fig. 13: **Unstable Case:** Minimum testing errors by configuration - 1000-step prediction.

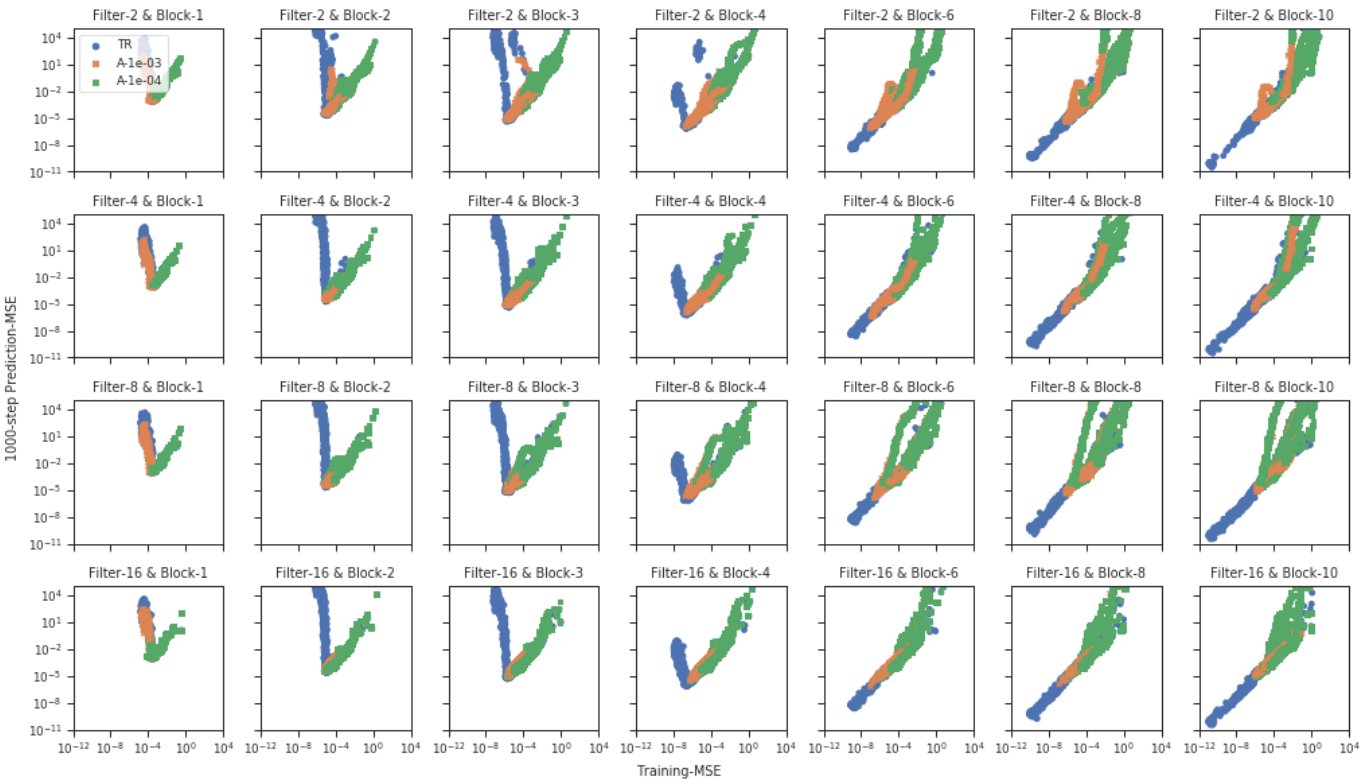


Fig. 14: **Unstable Case:** Training vs. 1000-step prediction errors (over the training process) by configuration.

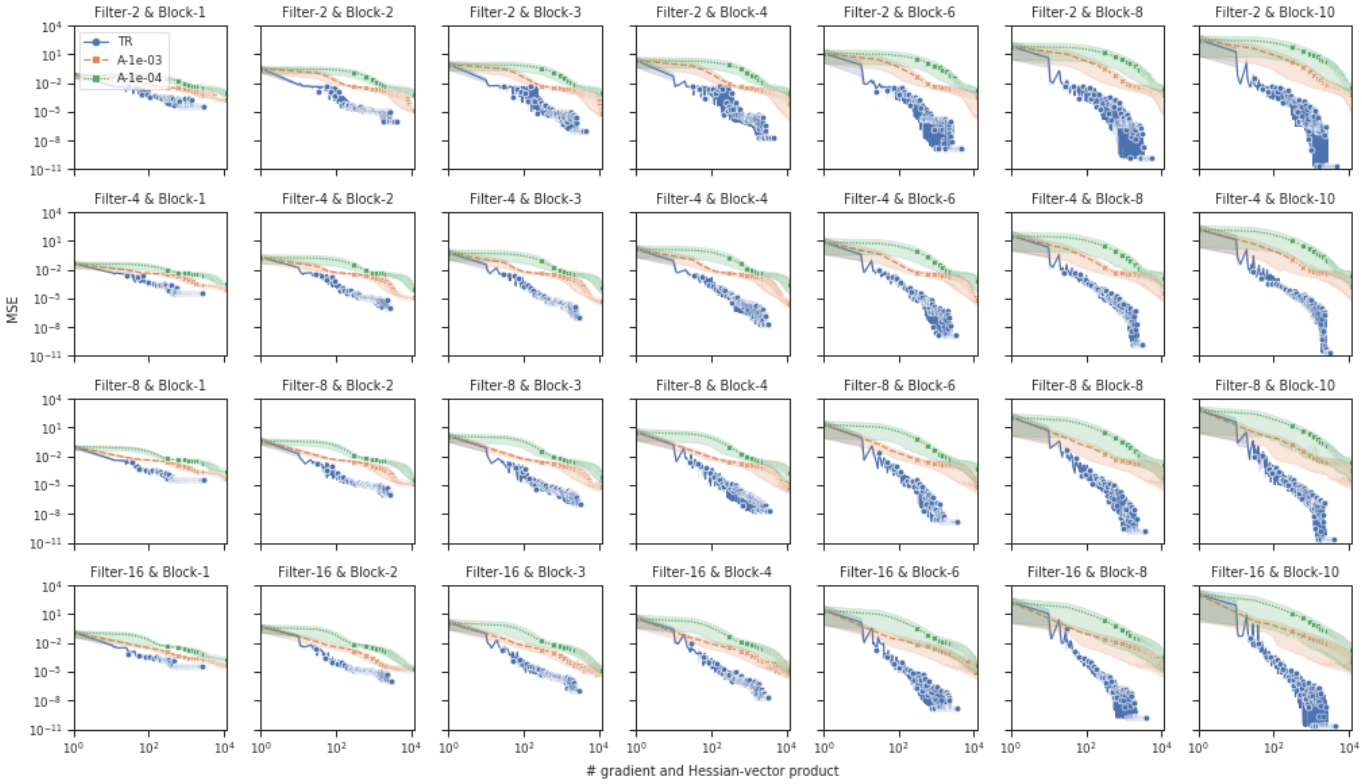


Fig. 15: **Unstable Case:** Evolution of the testing errors by configuration - one-step prediction.

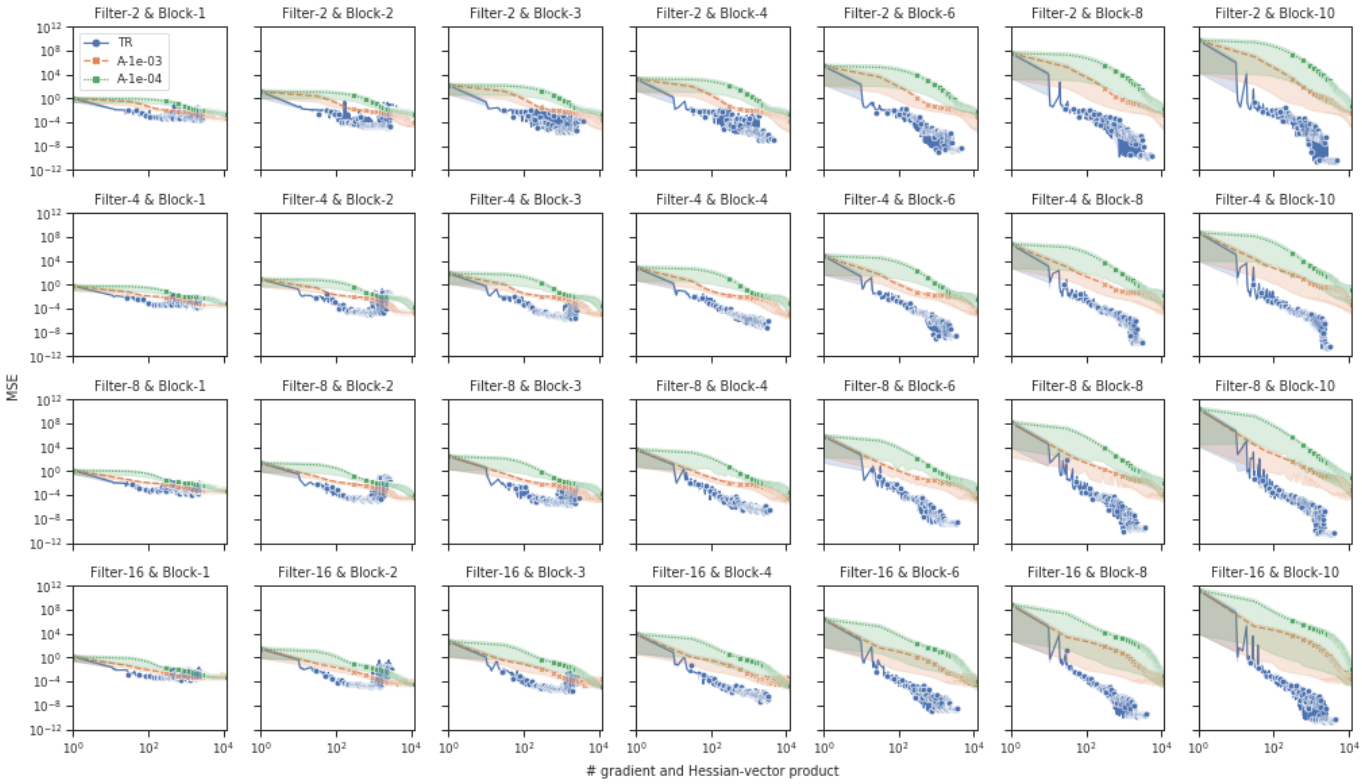


Fig. 16: **Unstable Case:** Evolution of the testing errors by configuration - multi-step prediction.

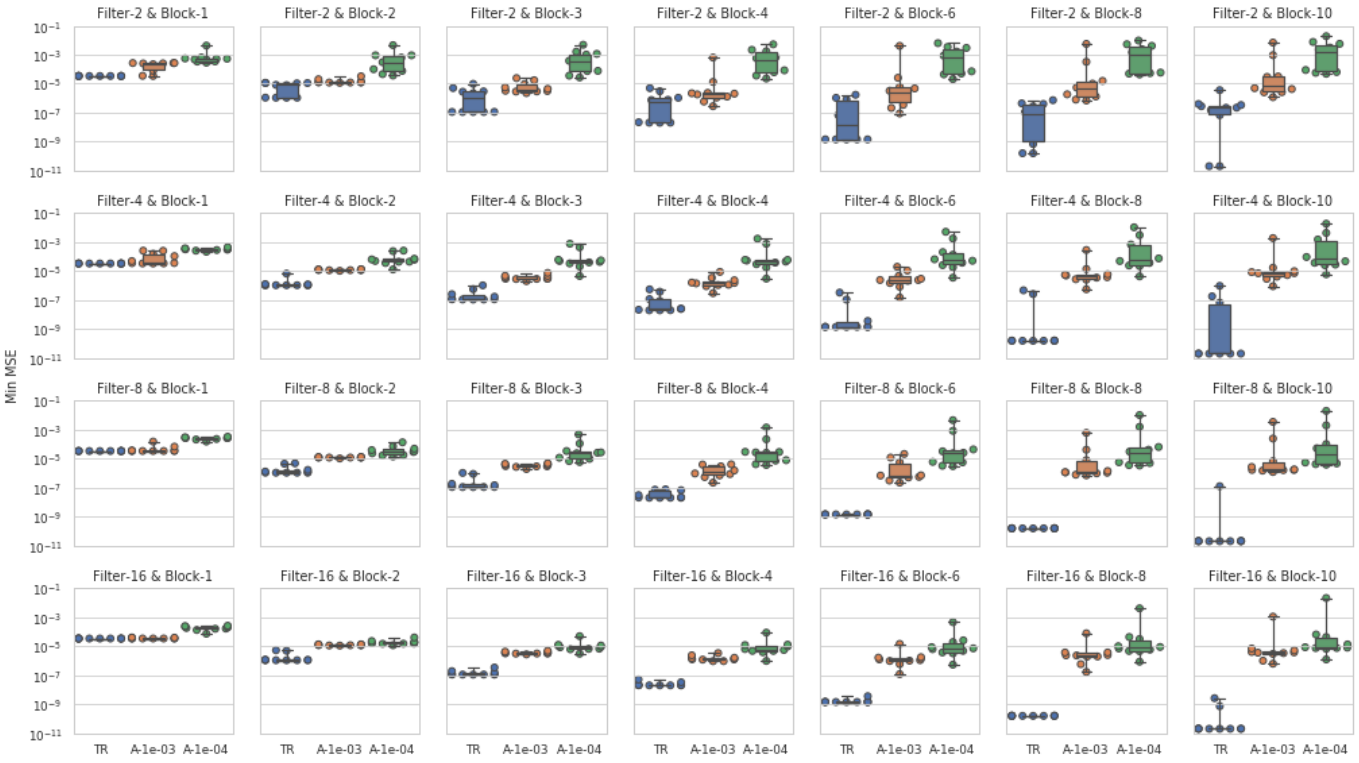


Fig. 17: **Unstable Case:** Minimum testing errors by configuration - one-step prediction.

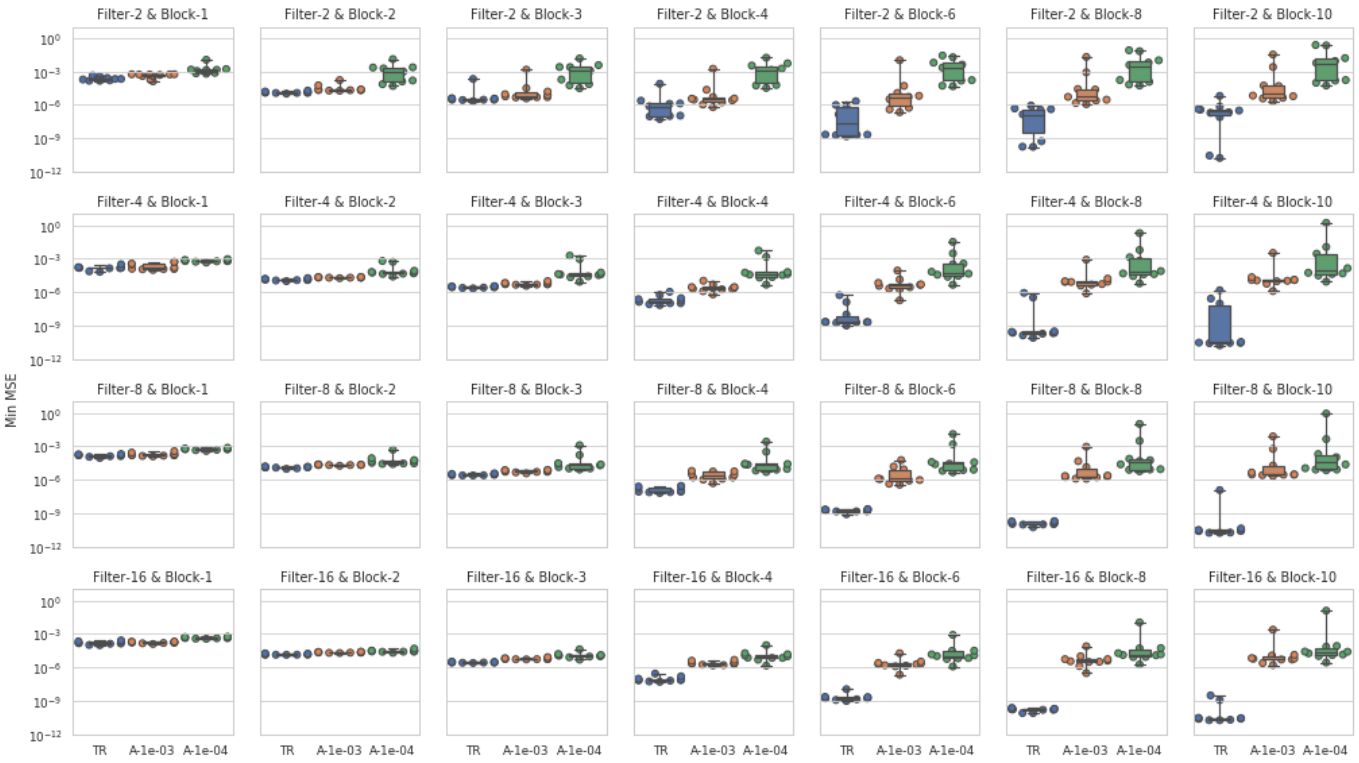


Fig. 18: **Unstable Case:** Minimum testing errors by configuration - multi-step prediction.

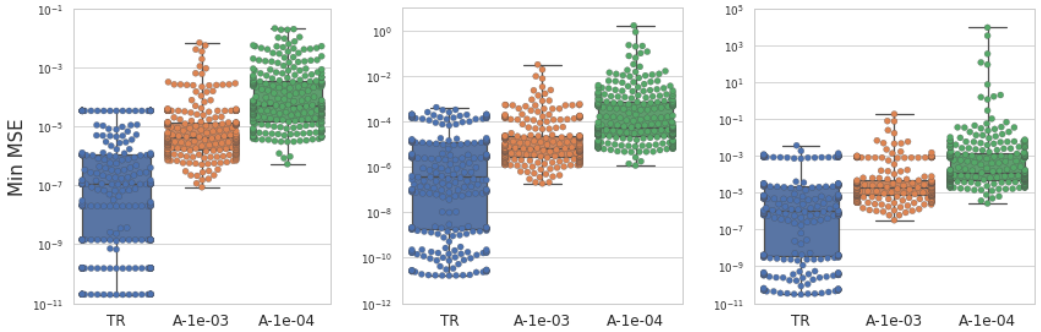


Fig. 19: **Unstable Case:** Minimum testing errors of the one-step (left), multi-step (middle) and 1000-step (right) predictions (aggregated over all configurations).

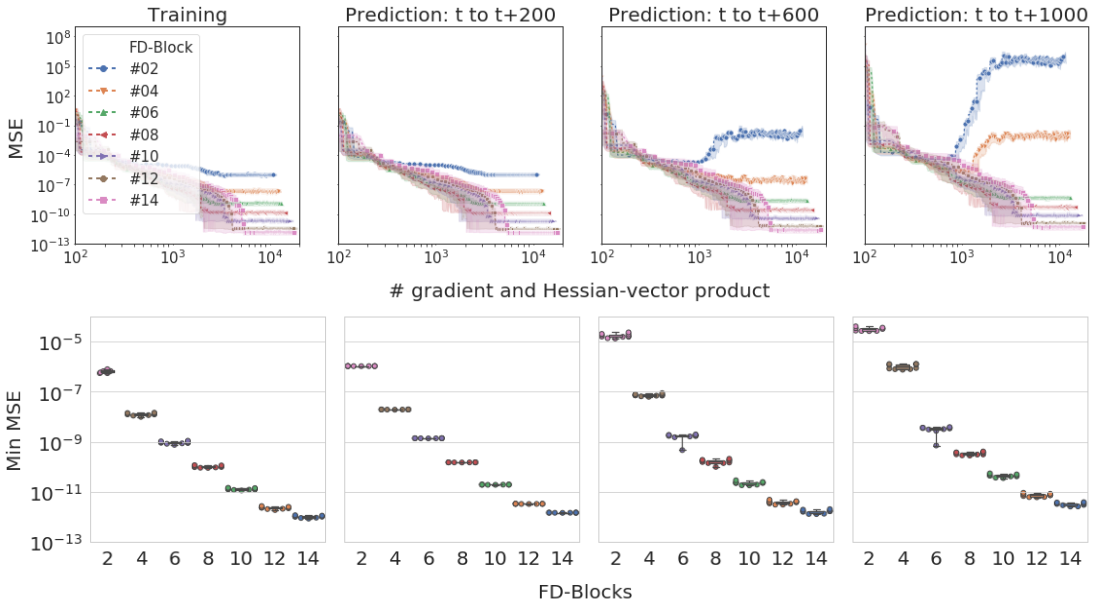


Fig. 20: **Unstable Case:** Evolution (first row) and the minimum (second row) of the training errors and testing errors of the one-, multi- and 1000-step prediction by configuration of `FD-Blocks`.

IV. Forcing Case

In this section, we present the experimental results of the forcing case. Fig. 21 shows the evolution of the training errors of different configurations. Fig. 22 shows the sequential predictions and the squared errors, and Fig. 23 shows the minimum testing errors (over the training process) of the 1000-step predictions by configuration. In addition, Fig. 24 shows the relationship between the training and testing errors (note: the lower and to the left is better). For the two supplementary testing procedures described in Section I-B, given the data of solution at t, the one-step prediction is made at $t+\Delta t$, i.e., $t+1$, and the multi-step prediction is made at $t+10\Delta t$, i.e., $t+10$. Fig. 25 & 26 show the evolutions of the testing errors (1) and Fig. 27 & 28 show the minimum testing errors of the one- and multi-step predictions. To summarize the testing performance, we put the minimum testing errors aggregated over all configurations in Fig. 29.

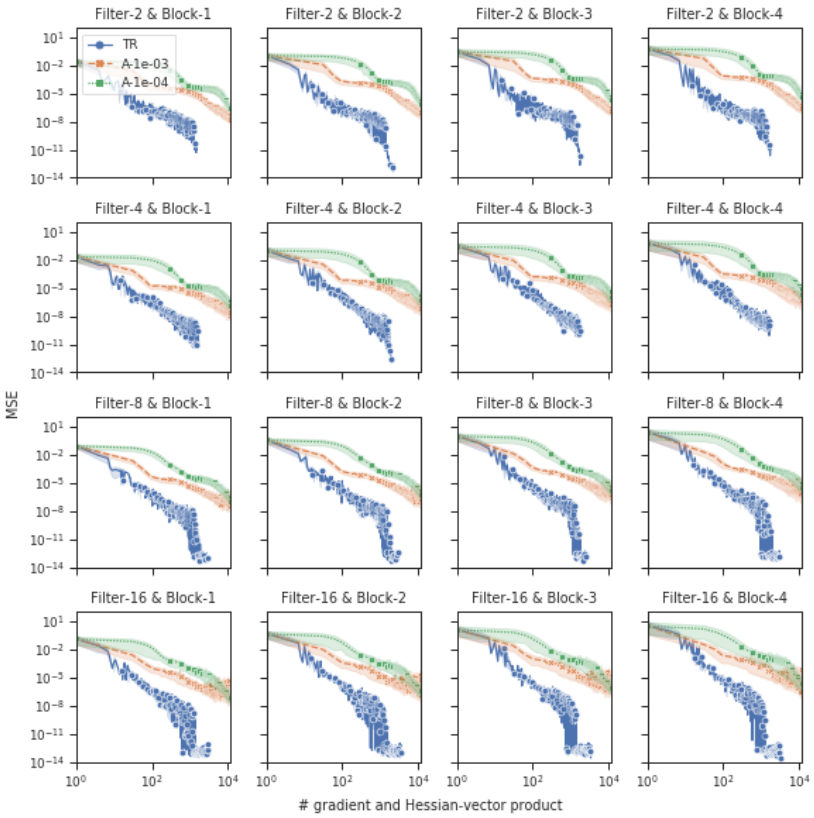


Fig. 21: **Forcing Case:** Evolution of the MSE loss of stochastic mini-batch by configuration.

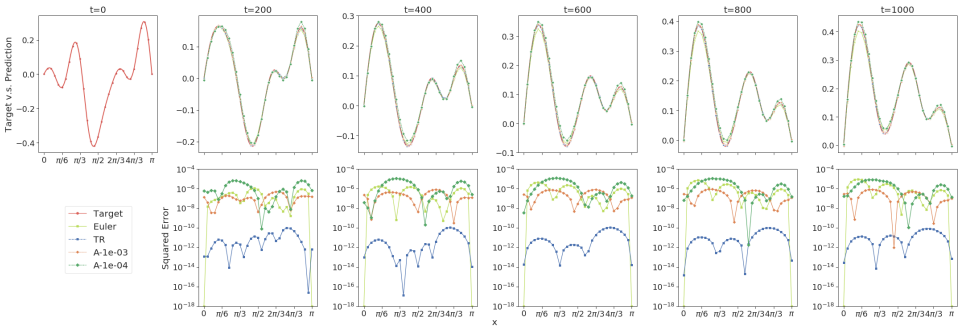


Fig. 22: **Forcing Case:** Sequence of predictions.

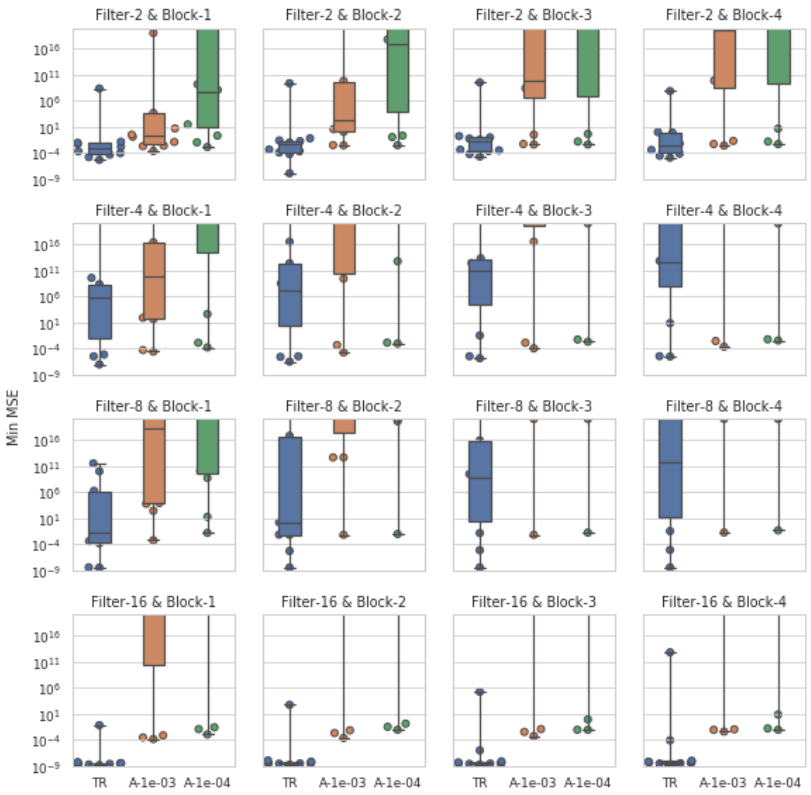


Fig. 23: **Forcing Case:** Minimum testing errors by configuration - 1000-step prediction.

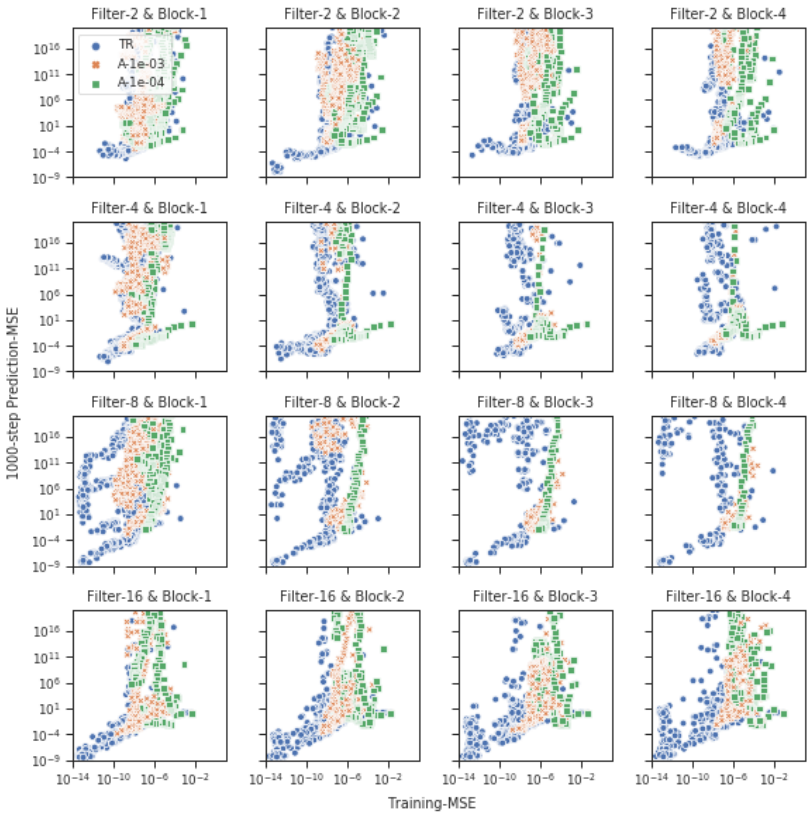


Fig. 24: **Forcing Case:** Training vs. 1000-step prediction errors (over the training process) by configuration.

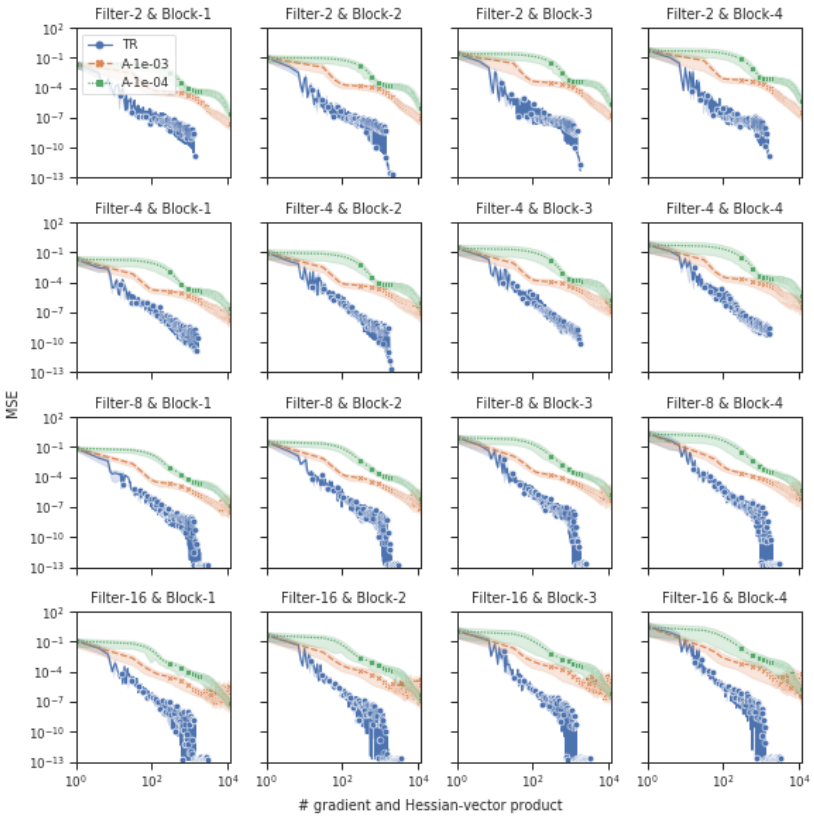


Fig. 25: **Forcing Case:** Evolution of the testing errors by configuration - one-step prediction.

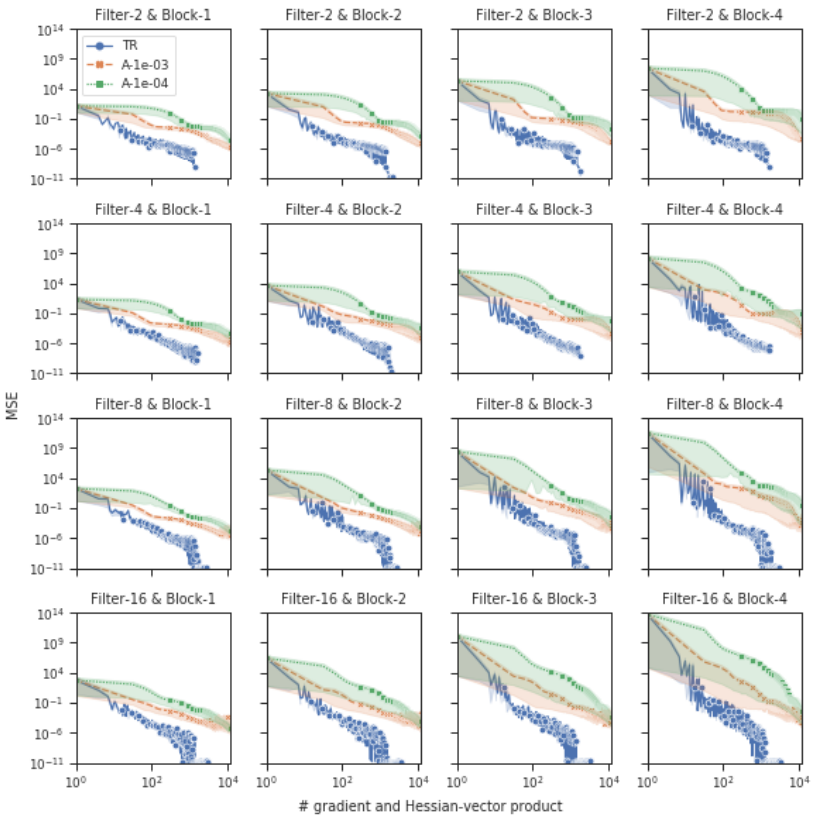


Fig. 26: **Forcing Case:** Evolution of the testing errors by configuration - multi-step prediction.

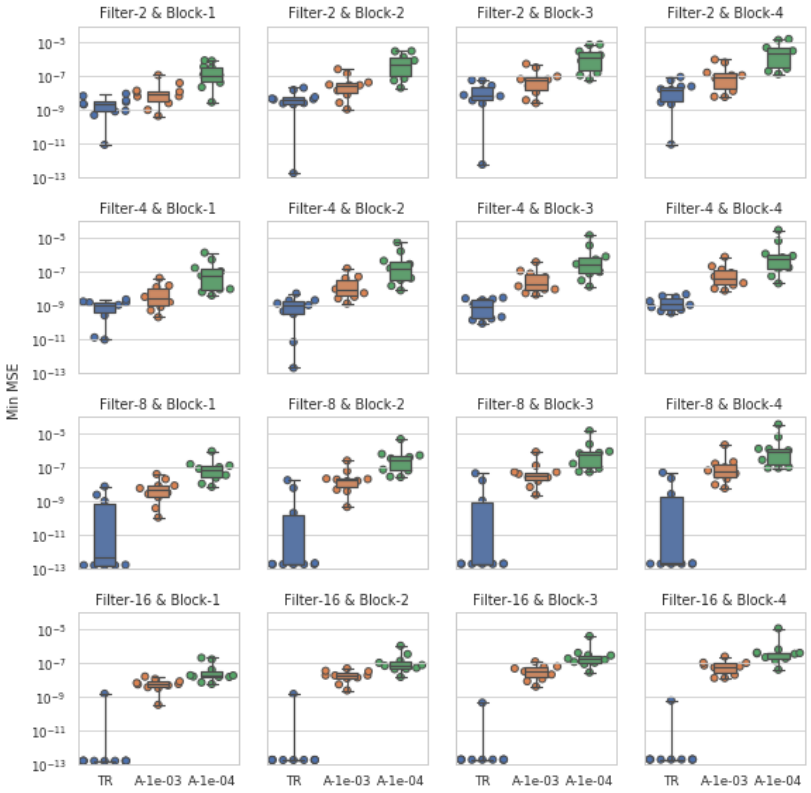


Fig. 27: **Forcing Case:** Minimum testing errors by configuration - one-step prediction.

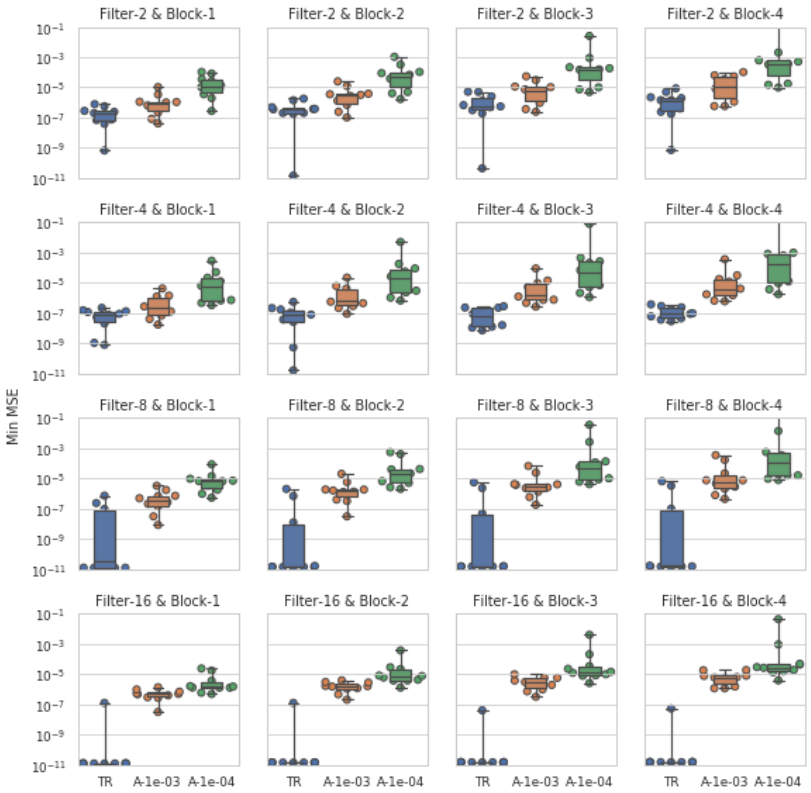


Fig. 28: **Forcing Case:** Minimum testing errors by configuration - multi-step prediction.

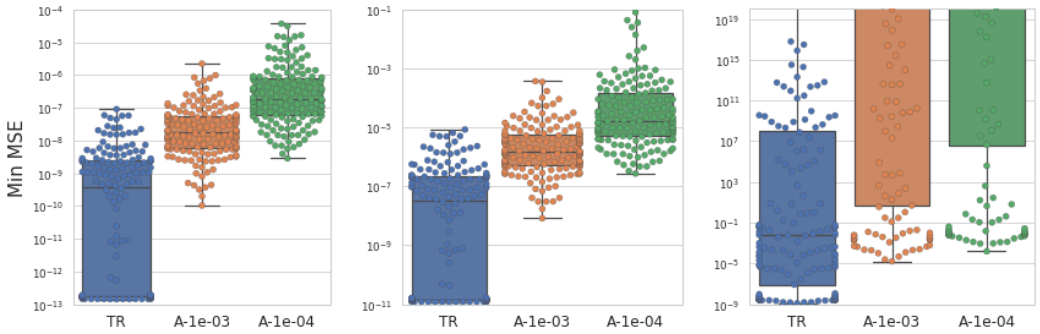
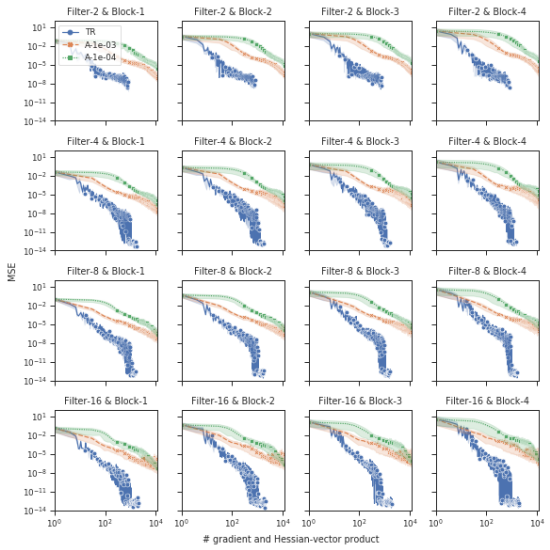


Fig. 29: **Forcing Case:** Minimum testing errors of the one-step (left), multi-step (middle) and 1000-step (right) predictions (aggregated over all configurations).

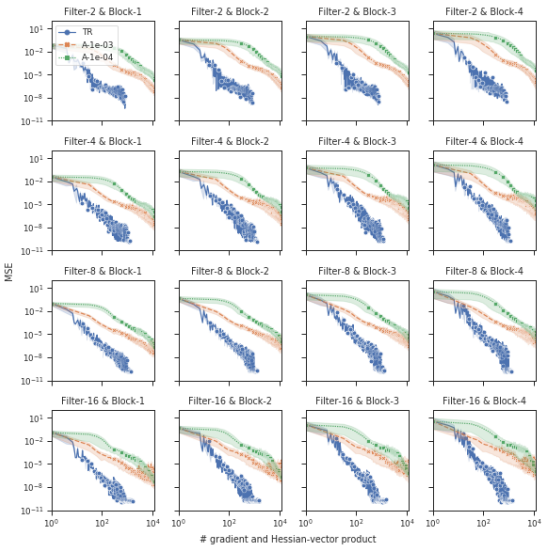
V. Noisy Case

In this section, we present the experimental results of the noisy case. Fig. 30 shows the evolution of the training errors of different configurations. Fig. 31 shows the sequential predictions and the squared errors, and Fig. 32 shows the minimum testing errors (over the training process) of the 1000-step predictions by configuration. In addition, Fig. 33 shows the relationship between the training and testing errors (note: the lower and to the left is better). For the two supplementary testing procedures described in Section I-B, given the data of solution at t, the one-step prediction is made at $t+\Delta t$, i.e., $t+1$, and the multi-step prediction is made at $t+10\Delta t$, i.e., $t+10$. Fig. 34 & 35 show the evolutions of the testing errors (1) and Fig. 36 & 37 show the minimum testing errors of the one- and multi-step predictions. To summarize the testing performance, we put the minimum testing errors aggregated over all configurations in Fig. 38.

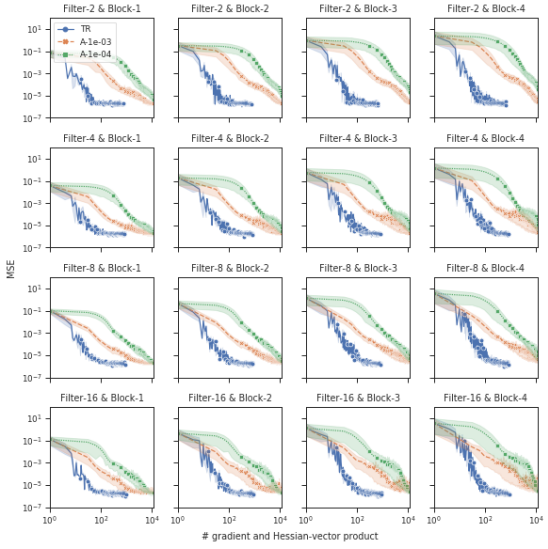
We further conducted experiments on the sensitivity of `FD-Net` to 10 different levels of the multiplicative noise and used 1 `FD-Block` and 16 `FD-Filters` to configure the networks. Fig. 39 shows the evolution of the training errors, Fig. 40 show the minimum testing errors over the training process, and Fig. 41 shows the relationship between the training and testing errors in the training process.



(a) Low

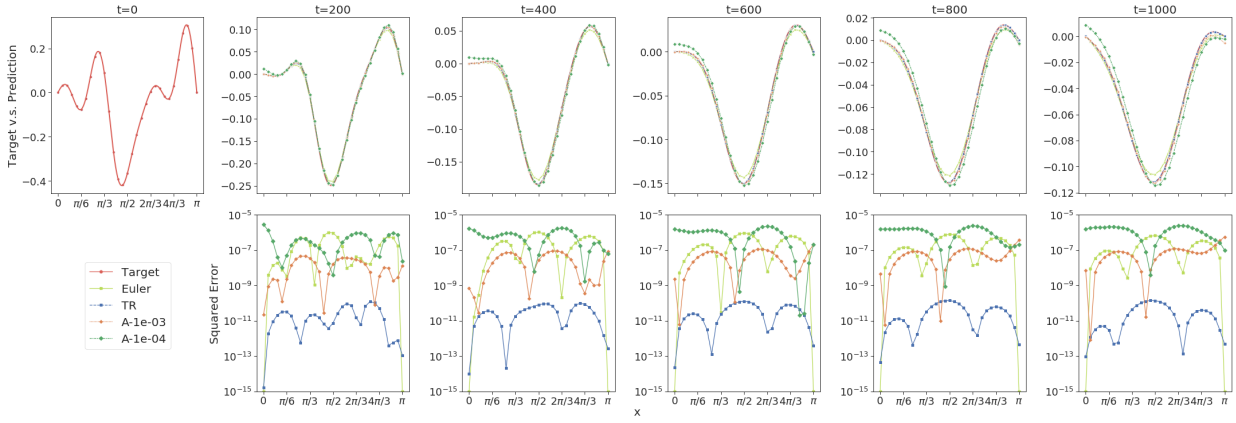


(b) Medium

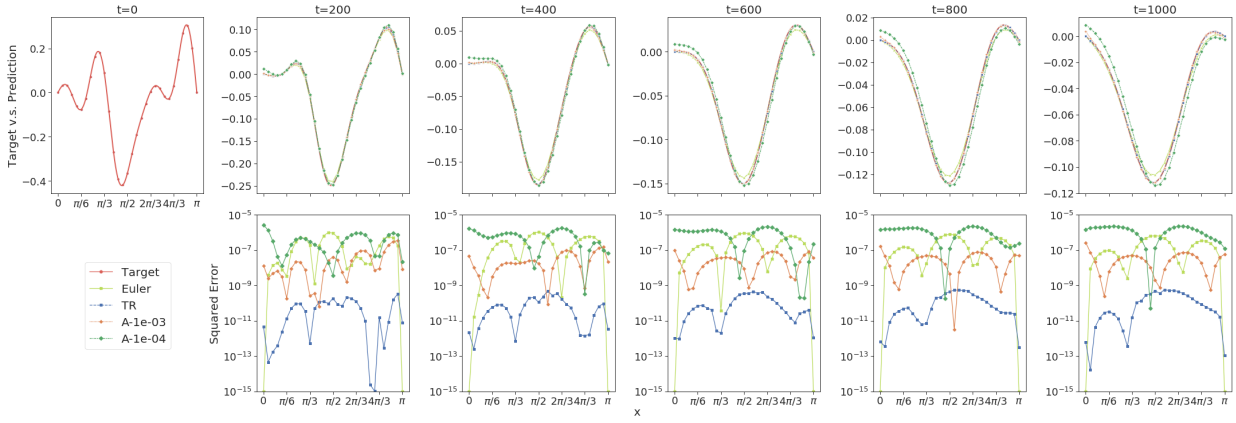


(c) High

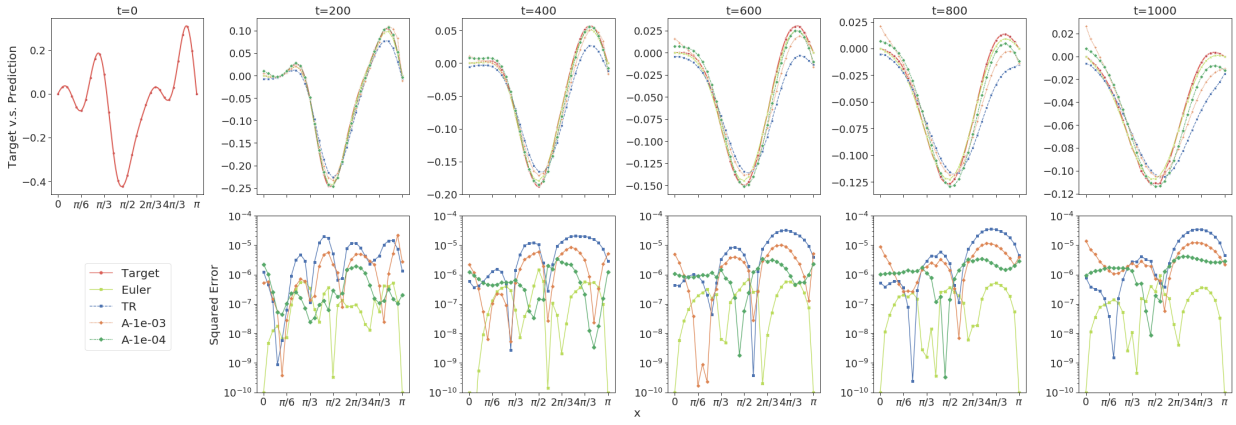
Fig. 30: **Noisy Case:** Evolution of the MSE loss of stochastic mini-batch by level of noise and configuration.



(a) Low

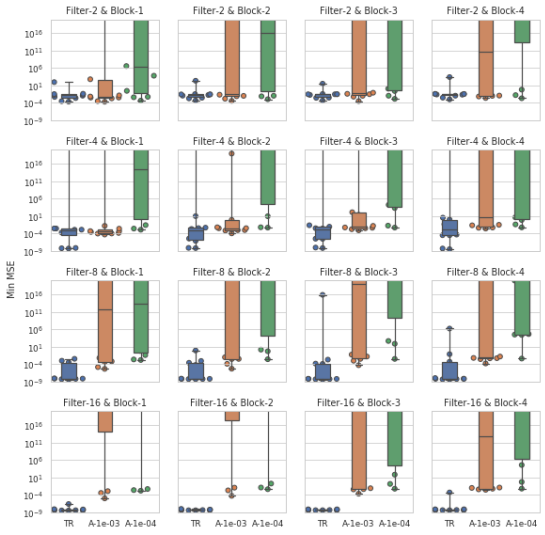


(b) Medium

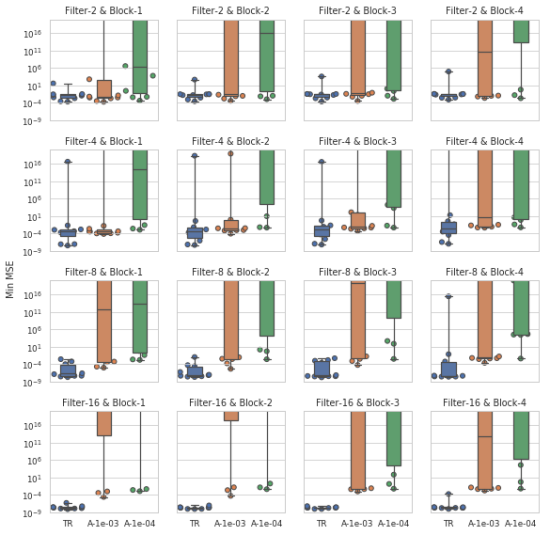


(c) High

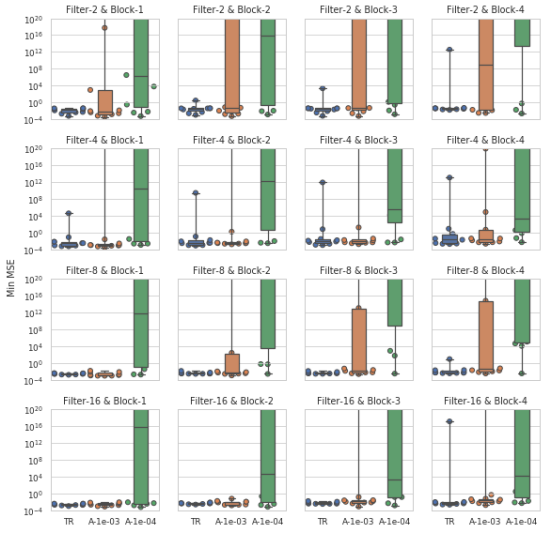
Fig. 31: **Noisy Case:** Sequence of predictions by level of noise.



(a) Low

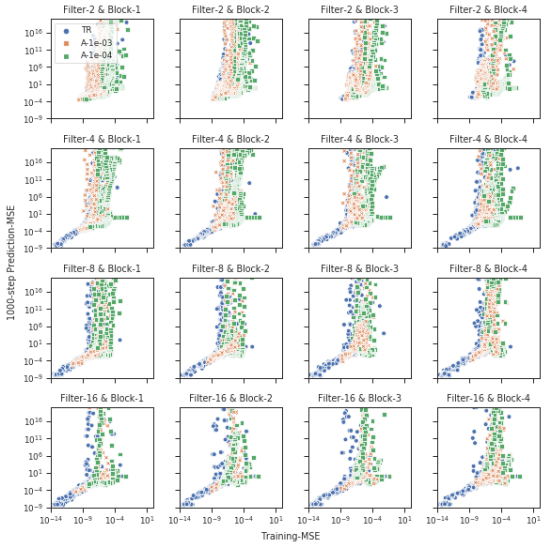


(b) Medium

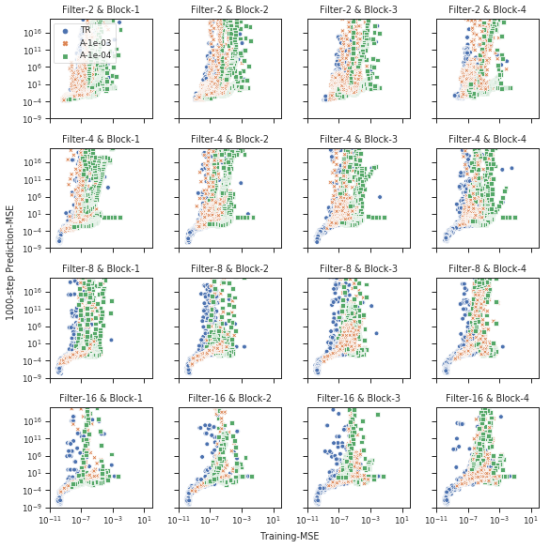


(c) High

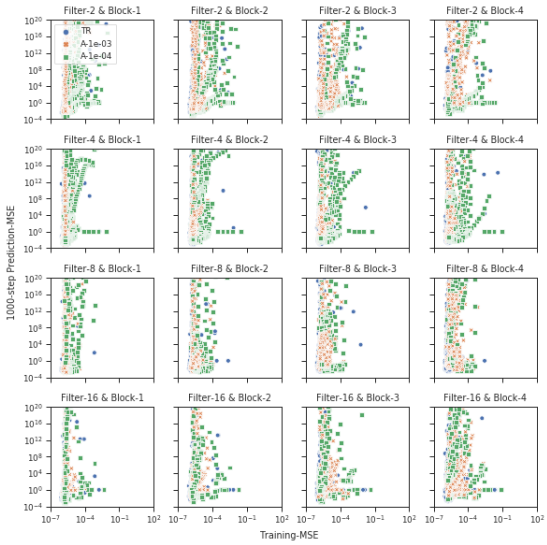
Fig. 32: **Noisy Case:** Minimum testing errors by level of noise and configuration - 1000-step prediction.



(a) Low

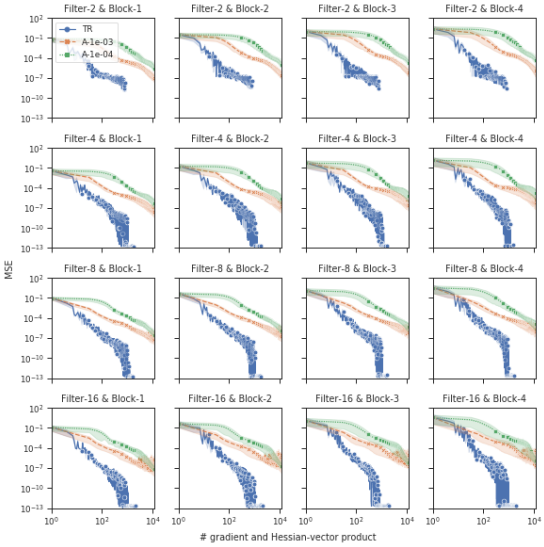


(b) Medium

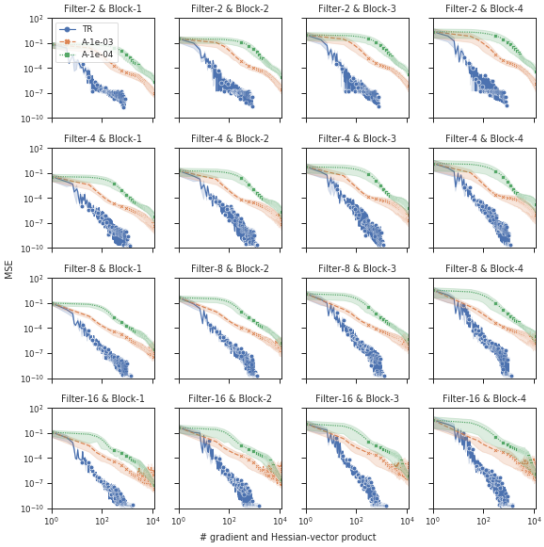


(c) High

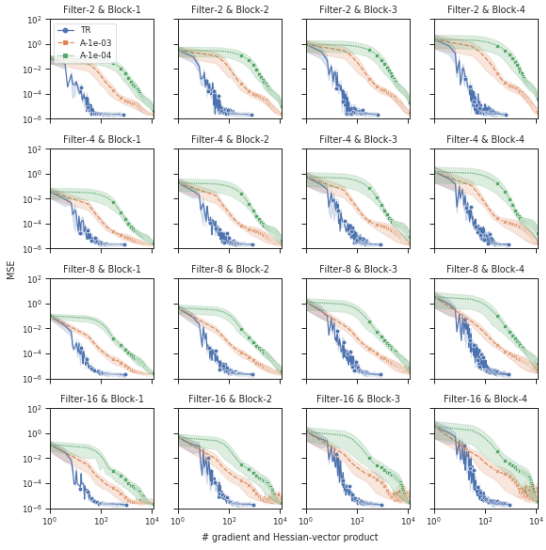
Fig. 33: **Noisy Case:** Training vs. 1000-step prediction errors (over the training process) by level of noise and configuration.



(a) Low

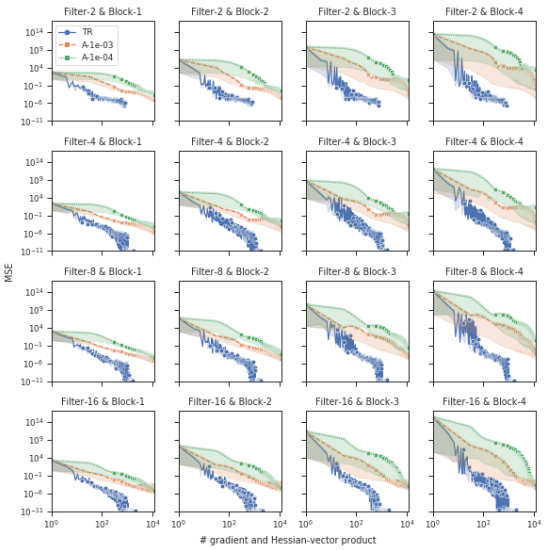


(b) Medium

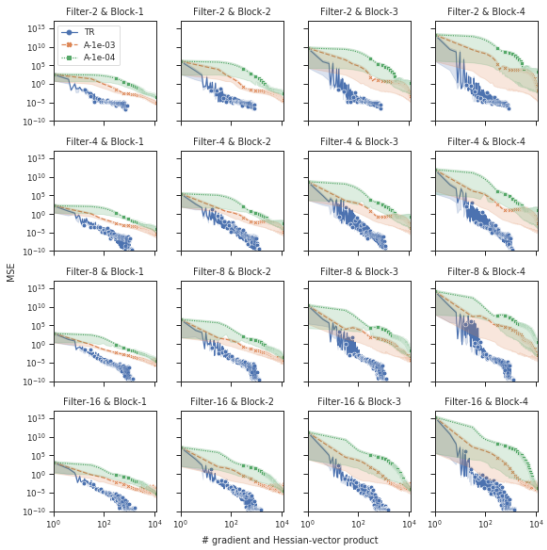


(c) High

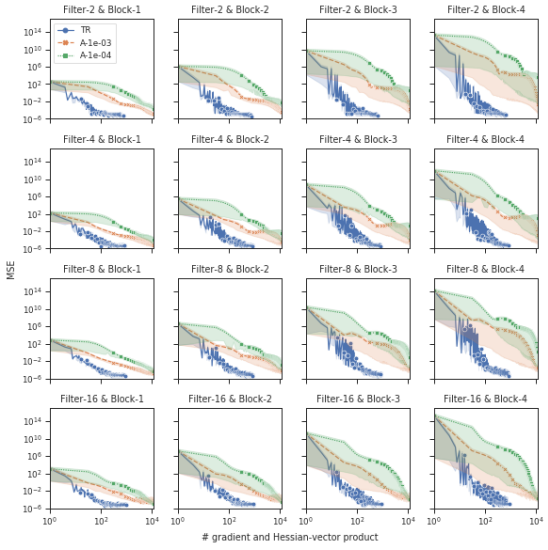
Fig. 34: **Noisy Case:** Evolution of the testing errors by level of noise and configuration - one-step prediction.



(a) Low

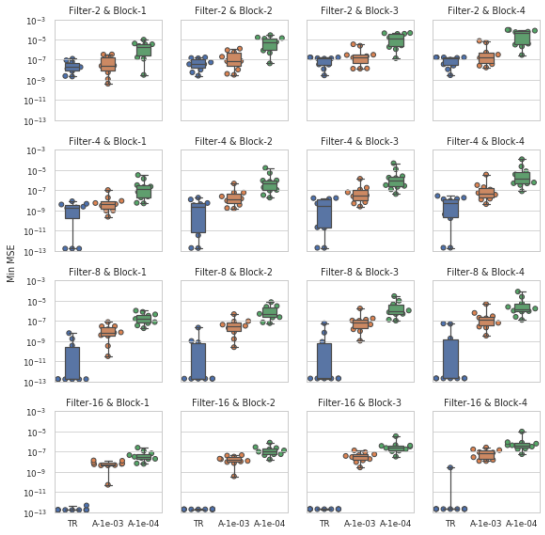


(b) Medium

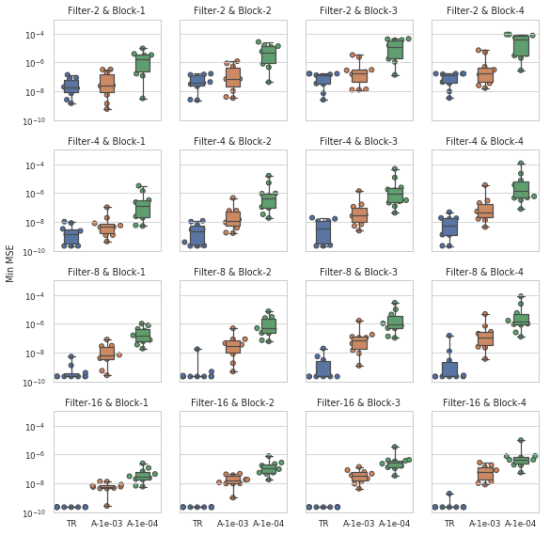


(c) High

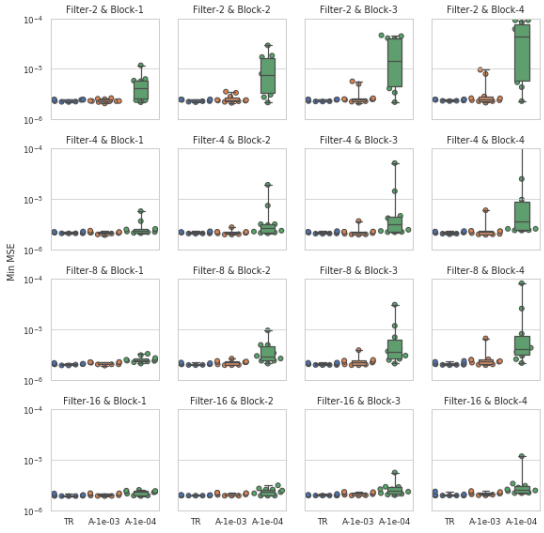
Fig. 35: **Noisy Case:** Evolution of the testing errors by level of noise and configuration - multi-step prediction.



(a) Low

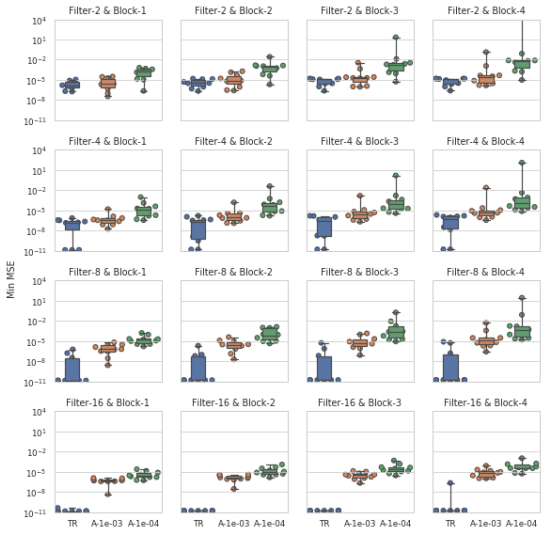


(b) Medium

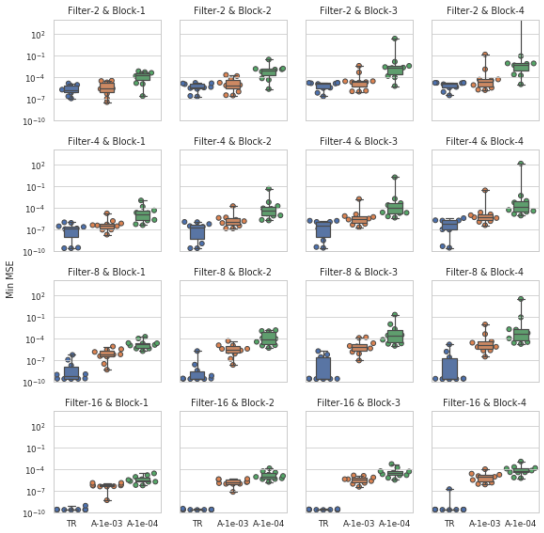


(c) High

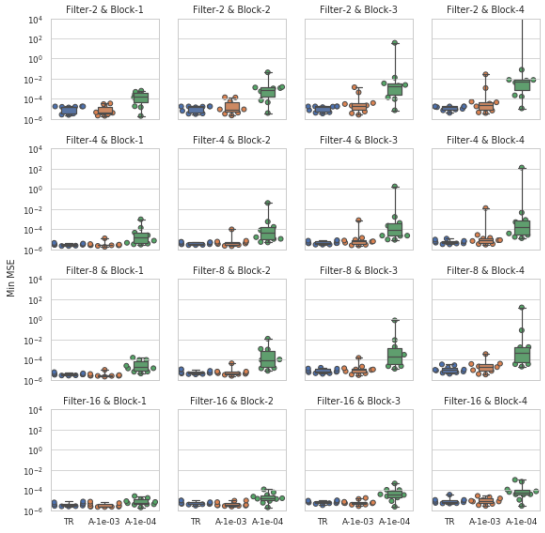
Fig. 36: **Noisy Case:** Minimum testing errors by level of noise and configuration - one-step prediction.



(a) Low

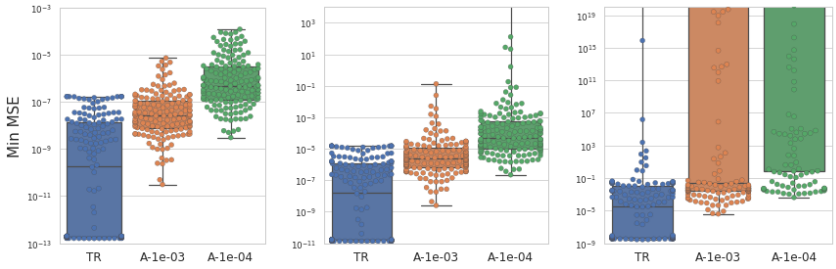


(b) Medium

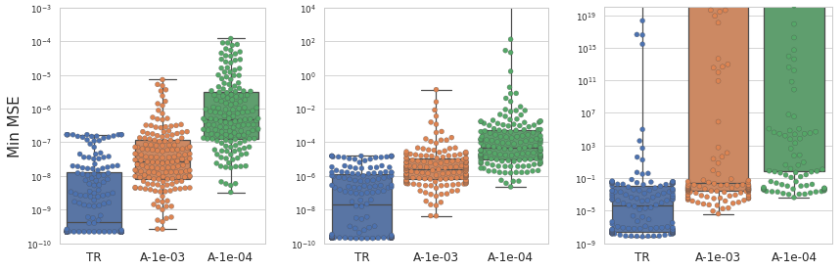


(c) High

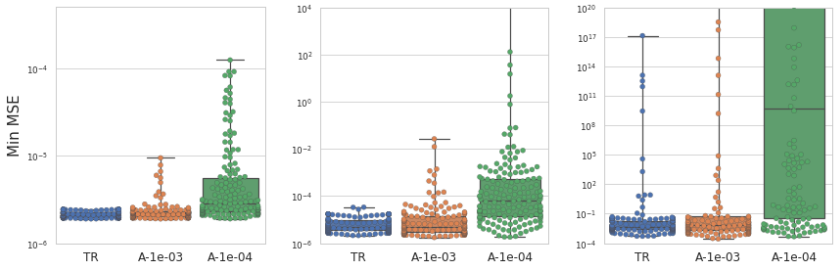
Fig. 37: **Noisy Case:** Minimum testing errors by level of noise and configuration - multi-step prediction.



(a) Low



(b) Medium



(c) High

Fig. 38: **Noisy Case:** Minimum testing errors of the one-step (left), multi-step (middle) and 1000-step (right) predictions (aggregated over all configurations) by level of noise.

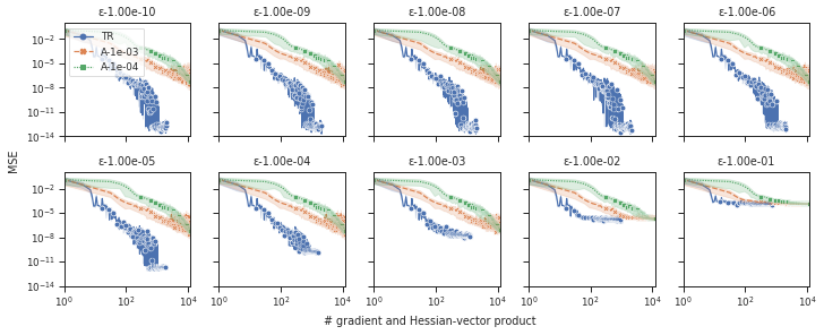
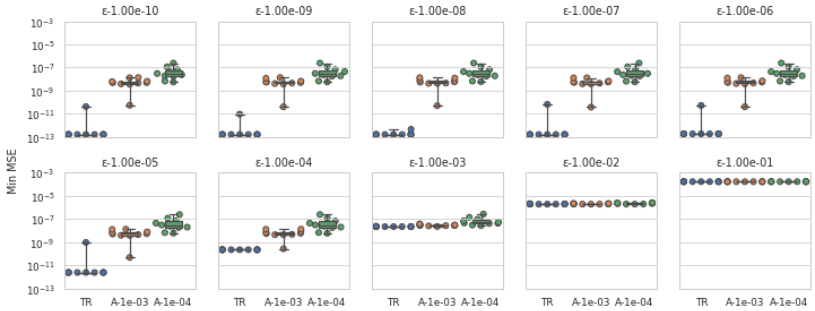
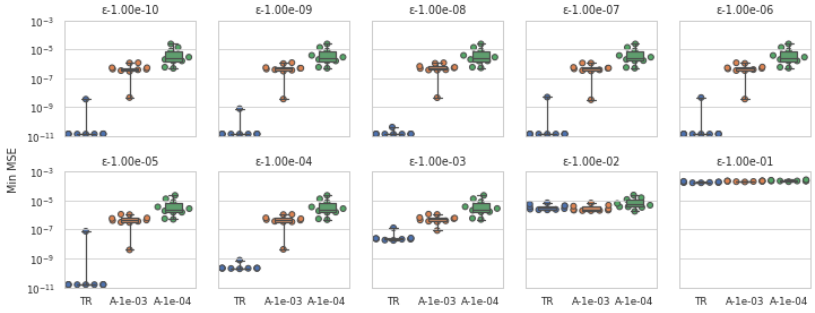


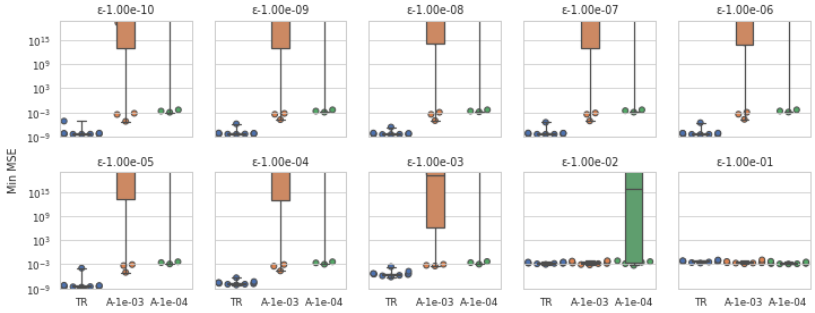
Fig. 39: **Noisy Case:** Sensitivity analysis - evolution of the MSE loss of stochastic mini-batch by level of noise.



(a) Low



(b) Medium



(c) High

Fig. 40: **Noisy Case:** Sensitivity analysis - minimum testing errors by level of noise.

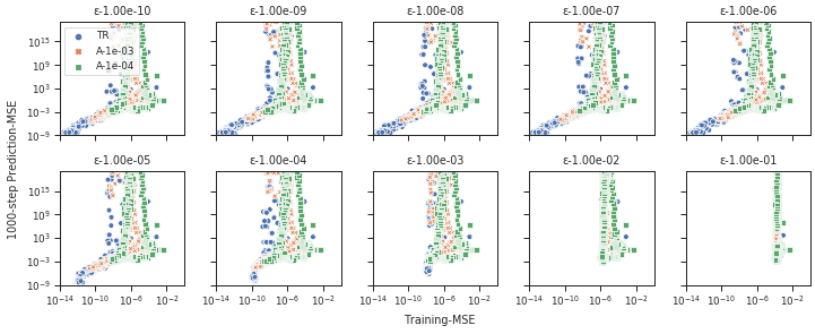


Fig. 41: **Noisy Case:** Sensitivity analysis - training vs. 1000-step prediction (over the training process) by level of noise.