APPENDIX

The following complements the main text and provides further experimental details to enable others to replicate our results.

8.1 DATA TRANSFORMATIONS

To complement the content of the paper, we provide further visualizations and result figures.

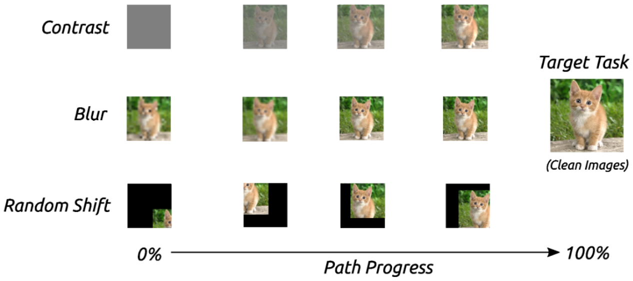

Figure 7: Vertically, different types of task distribution variations are displayed such as contrast variations, blurring and random shifting. Horizontally, the extent of these variations is adjusted for all data points giving us control over the task change discretization.

The task changes created by changing the data distribution along *blurring*, *contrast*, and *random shift* are visualized in the following. We provide Fig. 7 for a more intuitive grasp of the applied transformations and to improve reproducibility of our results.

Blurring: To blur an image, we resize the image to a smaller shape and then back to its original shape. In our case, when we referred to blurred images (0%) refers to a resizing to a quarter of the side-length for each dimension. The original image size we denote as 100% in Fig. 7.

Contrast: 0% contrast refers to a completely grey image. To retain meaningful gradients, a minimum contrast of 10% is used. The original image contrast we denote as 100%. In our experiments in Fig. 4 and Fig. 5 we begin source tasks involving contrast at 10% minimum contrast.

Random shift: Random shift uniformly samples an x and y-axis shift between $[-a, a]$ where a is $x\%$ of the original shape side length. Again a random shift of 0% does not shift the image while place 100%. Tasks in Fig. 4 and Fig. 5 involving random shift start with parameter $a = 100\%$side length for each dimension. All task changes gradually decrease this percentage to 0% of the side length.

For intermediate tasks and associated percentages $\%$ all transformations are scaled linearly.

8.2 STANDARD PYTORCH LAYER INITIALIZATION

Since our bias resetting experiments in Sec. 6 highlight the importance of initialization, we document the standard initialization (without bias resetting) used in all experiments.

Linear and convolutional layer: The weight matrices of the final linear layer and all convolutional filters of the Resnet are initialized using the "Kaiming uniform intialization" proposed by He et al. (2015). By default pytorch uses a factor $a = \sqrt{5}$ which is multiplied to an activation function specific factor to determine the "gain" of the intialization. Please refer to Pytorch and the referenced paper for more information.

Batch normalization layer and biases: All biases are initialized to 0 by default. Note that, to our knowledge, only the batchnormalization layers have a bias within the Resnet. All batch normalization gains are initialized to 1.

8.3 Complementary information on bias resetting experiments

To better understand the *bias resetting* results in Sec. 6, we provide a brief analysis of quantities of interest. We denote as *bias resetting* the process of perturbing biases by adding a constant offset to them. In some cases, most prominently for the CIFAR-10 dataset, this perturbation leads to a decrease of the negative pretrianing effect. Obtained generalization performance of pretrained networks is now closer to the performance of networks only trained on the standard data.

We summarize our analysis results in Tab. 2. In this table we record the summed L2-norm of the biases and summed L2-norm of the weight matrices and filters separately. They are place to the right of the associated mean value of *bias mean* and *weight mean*, respectively. Furthermore, we provide the *average activation percentage*, i.e. the frequency of how often a ReLU-activation function is active in the final trained network on standard unblurred images.

Table 2: Table detailing complementary statistics to the *bias resetting experiments* in Sec. 6. Rows are divided by the training scenario of when to reset biases and the dataset. We further separate the first three rows since no bias resetting is applied in these settings. Bias resetting appears to change the weight norms recorded at the very end of training on the standard uncorrupted data. Activity levels of this final model, which refers to the average percentage of time a ReLU in the model is active appears unchanged.

Dataset	**Scenario**	**Bias mean**	**L2-norm**	**Weights mean**	**L2-norm**	**Avg. activ.**
CIFAR-10	[No, No]	-0.16	6.95	1.56	69.74	69%
FMNIST-10	[No, No]	-0.05	1.26	12.92	303.41	73%
SVHN-10	[No, No]	-0.22	8.83	1.81	78.86	69%
CIFAR-10	[No, Yes]	-0.16	7.78	1.24	65.58	69%
FMNIST-10	[No, Yes]	0.17	2.57	17.45	405.23	74%
SVHN-10	[No, Yes]	-0.20	8.86	1.45	71.80	69%
CIFAR-10	[Yes, No]	-0.15	7.65	1.24	64.21	69%
FMNIST-10	[Yes, No]	0.30	4.48	16.30	379.43	74%
SVHN-10	[Yes, No]	-0.19	8.08	2.06	82.51	70%
CIFAR-10	[Yes, Yes]	-0.13	8.06	1.21	65.74	69%
FMNIST-10	[Yes, Yes]	0.41	5.99	15.26	355.85	74%
SVHN-10	[Yes, Yes]	-0.16	8.18	2.01	81.79	70%

8.4 Accuracy results associated to test loss figures

While the *tasks* we define optimize the cross-entropy error, we provide the associated classification accuracy results for completeness. Accuracy in itself is not part of our learning task but is instead a *proxy-task* that is commonly associated to the cross-entropy error. As visible in Fig. 8 and 11, the results mostly mirror the trends of the test loss visualization in Fig. 3 and 6. In some cases however, the trend of accuracies seem unrelated to the development of the test loss found in the main text.

8.5 Test loss on blurred images before training on standard data

In the following section we show the test loss achieved of the model trained *only* on blurred images evaluated on blurred images. This follows the natural question whether models that are already better on the first pretraining task are also better on the final standard dataset. Since it is more tangible and informative for learning paths consisting of only two tasks, we provide the plots for the bias resetting and learning rate experiments in Fig. 13 respectively Fig. 12. We first note with respect to the baseline that it is generally harder to distinguish classes in blurred images than classes in standard unblurred images.

More importantly, one can observe that a model that is better at distinguishing blurred images in the first task does not necessarily lead to a model which is better at distinguishing unblurred standard images. This aims to give provide a more comprehensive picture of the effects investigated in the presented experiments involving sequential learning effects.

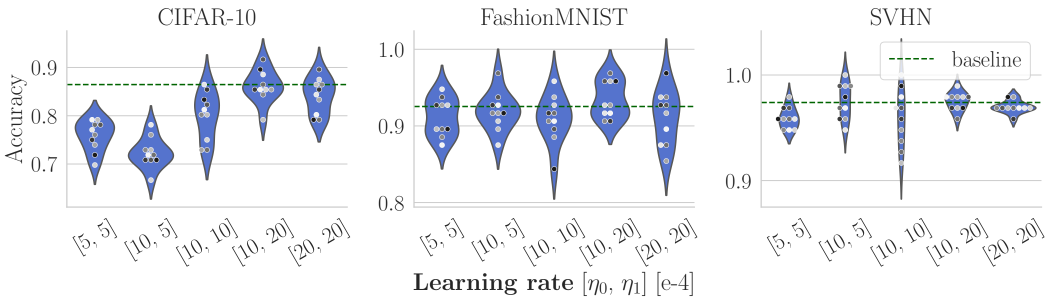

Figure 8: Accuracy on unblurred images. The x-axis displays the learning rates on first task with blurred images and the second task with unblurred images. The y-axis shows the accuracy on the unblurred images with dots colored to distinguish different random seeds. The dotted line indicates the mean baseline performance across 3 seeds. A clear trend that negative pretraining can be overcome when increasing the learning rate is visible for CIFAR-10.

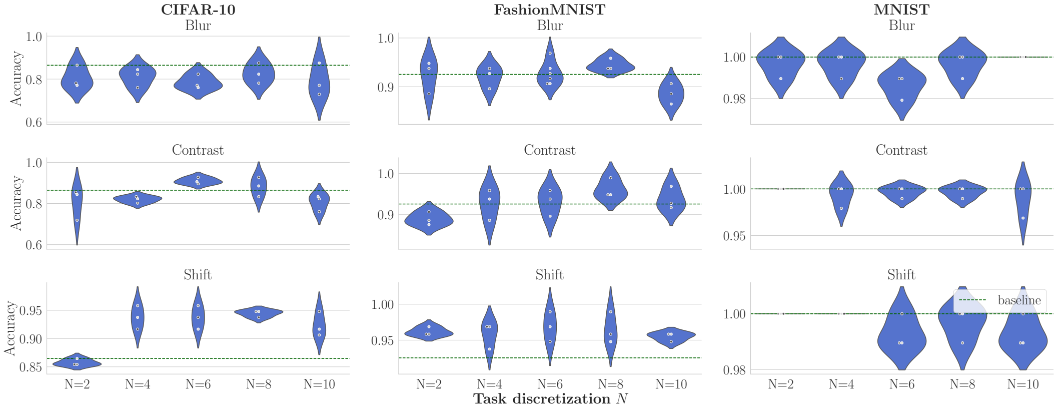

Figure 9: Final test accuracy on unblurred images as swarm and violin plots. The x-axis records the number of discretization steps, where $N = 2$ refers to training first, e.g. on blurred then on sharp images. Higher discretization provide a more gradual task shift from initial to final transformation. The y-axis shows the test accuracy on unblurred images with dots colored to distinguish different random seeds. The dotted line shows the mean baseline performance across 3 seeds. A clear decrease of the negative pretraining effect is visible for CIFAR-10. On FashionMNIST and SVHN, some seeds obtain better performance however on average results do not change noticeably.

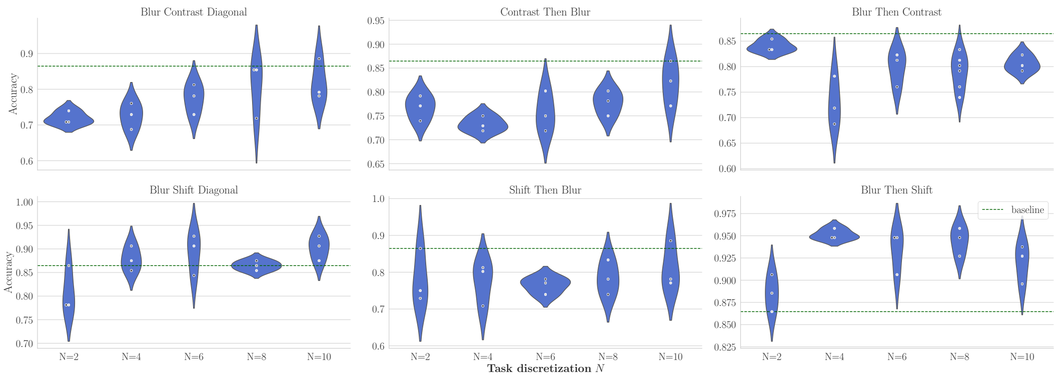

Figure 10: Visualization of 2D-task path on CIFAR-10 results as final test accuracy on standard unblurred images as swarm and violin plots. The x-axis records the number of discretization steps. The transformations are decreased in the order listed in the title. For diagonal paths, both transformations are decreased equally fast to zero. Higher discretization provide a more gradual task shift from initial to final transformation. The y-axis shows the test accuracy on standard images with dots colored to distinguish different random seeds. The dotted line shows the mean baseline performance across 3 seeds. A clear decrease of the negative pretraining effect is visible for many of the displayed 2D CIFAR-10. Notable is the stark differences between the order of task changes such as *Blur then Shift* and *Shift then Blur*.

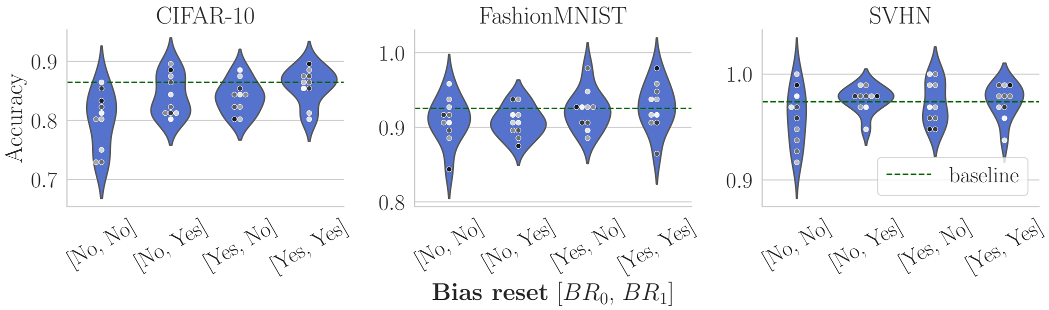

Figure 11: Final test accuracy on unblurred images as swarm and violin plot. The x-axis denotes whether bias resetting was applied before the first task with blurred images and/or the second task with unblurred images. The y-axis shows the test accuracy on unblurred images with dots colored to distinguish different random seeds. The dotted line shows the mean baseline performance across 3 seeds. A clear decrease of the negative pretraining effect is visible for CIFAR-10. On FashionMNIST and MNIST, some seeds obtain better performance however on average results do not change compared to the $[\text{No}, \text{No}]$ negatively pretrained network. MNIST results are very close to 100% accuracy without a noticeable negative pretraining effect.

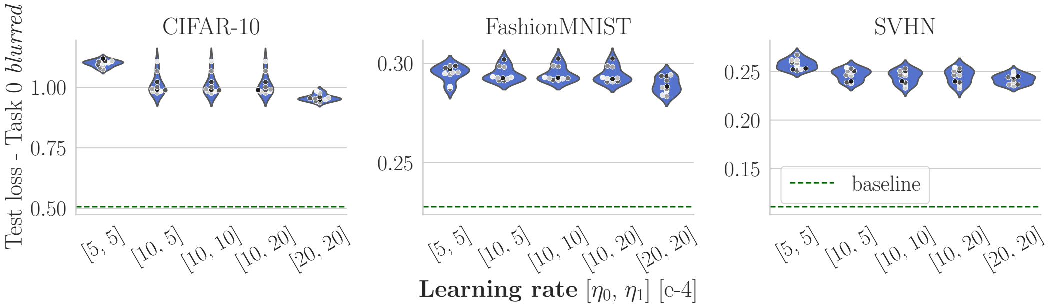

Figure 12: Test loss of the model trained *only* on task 0, i.e. blurred images, evaluated on the blurred task 0. Note that to be consistent with our definition of task, not only training but also validation and testset are transformed and blurred. The x-axis displays the learning rates on the first task with blurred images and the second task (unused in this experiment) with standard unblurred images. The y-axis shows the test loss on blurred images with dots colored to distinguish different random seeds. The dotted line shows the mean baseline performance across 3 seeds. Note, that the baseline performance still refers to the performance on networks trained from scratch on standard unblurred image and tested on standard images.

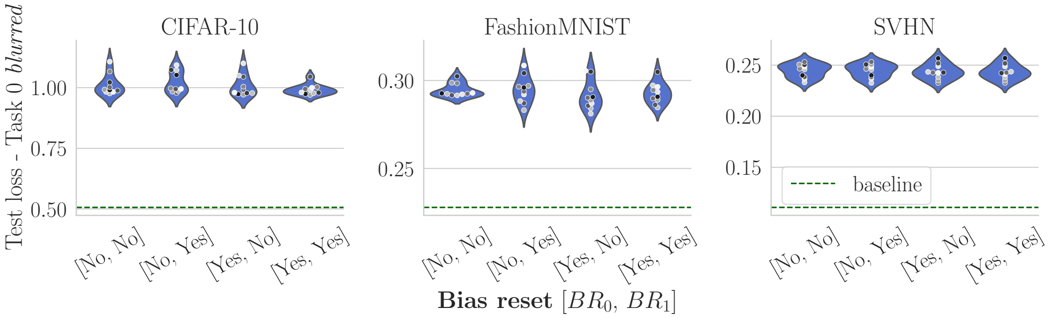

Figure 13: Test loss of the model trained *only* on task 0, i.e. blurred images, evaluated on the blurred task 0. Note that to be consistent with our definition of task, not only training but also validation and testset are transformed and blurred. The x-axis denotes whether bias resetting was applied before the first task with blurred images and/or the second task (unused in this experiment) with unblurred images. The y-axis shows the test loss on blurred images with dots colored to distinguish different random seeds. The dotted line shows the mean baseline performance across 3 seeds. Note, that the baseline performance still refers to the performance on networks trained from scratch on standard unblurred image and tested on standard images.