

Improving Robustness of Convolutional Networks Through Sleep-Like Replay

Jean Erik Delanois

Dept. of Computer Science & Engineering
University of California San Diego
La Jolla, USA
jdelanois@ucsd.edu

Aditya Ahuja

Dept. of Computer Science & Engineering
University of California San Diego
La Jolla, USA
adahuja@ucsd.edu

Giri P Krishnan

Dept. of Medicine
University of California San Diego
La Jolla, USA
gkrishnan@ucsd.edu

Timothy Tadros

Dept. of Medicine
University of California San Diego
La Jolla, USA
ttadros@ucsd.edu

Julian McAuley

Dept. of Computer Science & Engineering
University of California San Diego
La Jolla, USA
jmcauley@ucsd.edu

Maxim Bazhenov

Dept. of Medicine
University of California San Diego
La Jolla, USA
mbazhenov@ucsd.edu

Abstract—Convolutional neural networks (CNNs) are a foundational model architecture utilized to perform a wide variety of visual tasks. On image classification tasks CNNs achieve high performance, however model accuracy degrades quickly when inputs are perturbed by distortions such as additive noise or blurring. This drop in performance partly arises from incorrect detection of local features by convolutional layers. In this work, we develop a neuroscience-inspired unsupervised Sleep Replay Consolidation (SRC) algorithm for improving convolutional filter’s robustness to perturbations. We demonstrate that sleep-based optimization improves the quality of convolutional layers by the selective modification of spatial gradients across filters. We further show that, compared to other approaches such as fine-tuning, a single sleep phase improves robustness across different types of distortions in a data efficient manner.

Index Terms—cnn, convolution, sleep, generalization, robustness

I. INTRODUCTION

Over the past few decades, computer science has made remarkable advancements in the development of models capable of performing intricate visual tasks. Deep learning, in particular, has played a pivotal role in driving this progress, with convolutional neural networks (CNNs) emerging as a significant breakthrough. Inspired by the structural characteristics of the human visual system [8], CNNs owe their success to the introduction of convolutional layers by Lecun et al. [14], [15]. By combining convolutional and feedforward layers, deep networks have achieved state-of-the-art performance for classification and generative tasks [23].

However, despite their proven usefulness, convolutional filters still suffer from significant limitations. While the human visual system excels at accurately performing image-based tasks, even in the presence of substantial perturbations, CNNs trained using backpropagation-based methods are highly sensitive to distortions [4]. The impressive performance of these networks quickly degrades when models operate in real-life applications and dynamic uncontrolled environments modify

inputs with perturbations such as additive noise, blur, or other distortions (e.g., lighting, image quality, background, contrast, and perspective) [3]. This decrease in performance could be attributed to the perturbations degrading the quality of features that the convolutional layers are able to extract. Since the convolutional layers are trained on unperturbed (clean) images, they are unable to extract useful features from distorted ones. Most existing methods for improving the robustness of convolutional filters often involve explicit fine-tuning on predefined sets of perturbations or data augmentations [27], [30]. However, such supervised approaches require prior knowledge of the specific deformations or extensive training. These techniques face challenges when limited data is available for fine-tuning or when unforeseen and untrained

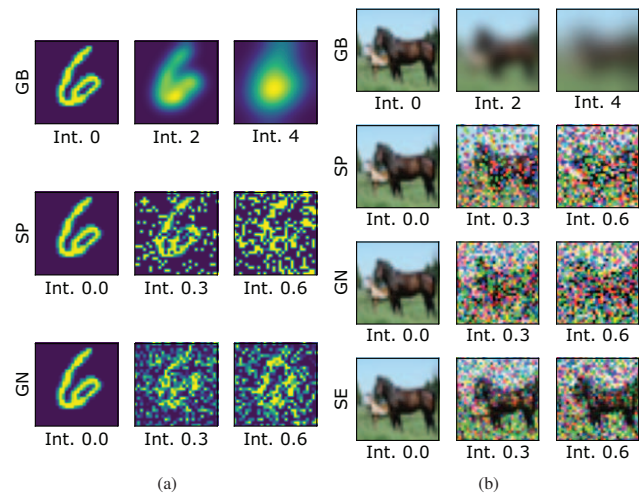


Fig. 1: Example images from MNIST (a) and CIFAR-10 (b) shown over distortion types (Gaussian noise (GN), Gaussian blur (GB), Salt & pepper (SP), and Speckle (SE)) with varying magnitude. Rows determine distortion type while columns display increasing intensity (Int.) magnitude from left to right.

Supported by NSF (grant 2223839) and NIH (grant 1R01MH125557)

distortions are encountered in real-world scenarios, this leads to a lack of generalization to out-of-distribution examples.

In contrast, biological systems have leveraged other mechanisms to improve memory representation and increase generalizability. Sleep has long been known to enhance learning in situations with limited experience, facilitate continuous learning, generalize knowledge acquired during wakefulness, and enable backward and forward transfer of knowledge [2], [11], [12], [16], [18], [28]. This functionality is prevalent and highly stereotyped in a variety of species ranging from insects [5], [17], [31] to mammals [2], [18]. Two crucial components are believed to underlie the role of sleep in memory consolidation: the spontaneous replay of memory traces in the absence of external input and local unsupervised synaptic plasticity that modifies synaptic weights [22], [29]. Previous studies have demonstrated that applying sleep-like processing, Sleep Replay Consolidation (SRC), to fully connected feedforward networks can enhance continual learning during sequential task training [25] and improve model robustness and generalizability [24].

While several other biologically inspired approaches to enhance network generalizability to visual distortions exist, they often suffer from increased computational cost [26], lack dynamism [6], or require gathering expensive neural recordings or other hard to acquire data [7], [19]. To address these limitations, we present a novel approach that implements SRC in exclusively convolutional layers, thereby extending the previous work by making SRC applicable to all segments of the CNN architecture. Importantly, our method provides a dynamic solution that does not increase inference computation costs.

SRC is implemented by converting the CNN to a spiking neural network (SNN) and simulating unsupervised replay in SNN. This involves (a) replacing the ReLU activation function with a Heaviside function to gain a notion of spikes, (b) introducing input noise reflective of the training data to induce network activity, (c) applying local Hebbian-type plasticity rules to convolutional layers to modify synapses based on spiking patterns. We evaluate our method using two well-known image classification data sets, MNIST and CIFAR-10, and incorporate standard distortions commonly encountered in both machine learning and real-world environments. These distortions include Gaussian blur, Additive Gaussian noise, Salt & pepper, and Speckle, with varying intensities. Figure 1 illustrates the diverse range of distortions used for evaluation. Our findings demonstrate that sleep-based optimization enhances the structure of convolutional blocks, enabling CNNs to improve their performance on distorted data.

A. Main contributions

- We develop an unsupervised sleep-like optimization algorithm, Sleep Replay Consolidation (SRC), for convolutional networks to improve robustness and generalization to noisy inputs.
- Our biologically inspired approach is computationally efficient, does not increase inference cost, and does not

require prior knowledge of the type of input perturbation while providing improvements across different types of distortions. In contrast, other biologically motivated methods are costly and fine tuning approaches only improve performance on pre-defined augmentations.

- We identify that SRC modifies CNN filters through selective gradient expansion focusing CNN attention to the critical image features that result in improved generalization.

II. METHODS

A. Data and Distortions

We tested SRC on two standard image classification data sets, MNIST [15] and CIFAR10 [13]. MNIST consists of 60,000 28x28 monochromatic hand written digits (0-9) while CIFAR-10 contains 60,000 32x32 color images of 10 classes (cars, birds, ships, etc). We applied a variety of common distortions (as used in [4], [6], [19], [26], [27], [30]) to these data sets and tested model performance across a variety of intensities. Certain distortions, such as brightening / darkening, yielded minuscule degradation in performance causing any potential benefits to be masked; we therefore only selected distortions that caused a significant decline in accuracy for the baseline model. All distorted values were clamped at the minimum and maximum pixel values to keep inputs in a reasonable range. Our final set of distortions is detailed below:

- Gaussian blur (GB): Involves convolving the input image with a Gaussian kernel, varying σ values are used to modify intensity. This type of distortion can be introduced when items present in the image are in motion.
- Additive Gaussian noise (GN): Noise drawn from a Gaussian distribution is added pixel-wise to the input image.
- Salt and pepper (SP): Also known as impulse noise, randomly selects input image pixels and sets it to either the minimum or maximum possible input value, the frequency of pixels selected controls the intensity. This type of input noise can arise in digital images taken by cameras with faulty sensors.
- Speckle (SE): A pixel-wise multiplicative noise where a random value is drawn from a Gaussian distribution and multiplied with the original pixel value to generate the new input values. Speckle noise is commonly a result of wave interference in images that are generated through the emission of specific frequencies of light, such as ultrasound and/or radar.

Visualizations of all distortions are shown in Figure 1.

B. Models

In an effort to generate interpretable results, we used smaller, more simple models with the goal of improving transparency and understandability of the underlying mechanisms. For MNIST we used a four layer CNN consisting of two convolutional and two feedforward layers. Both convolutional layers leveraged 3x3 filters with a stride of one, no padding, and a ReLU activation, each filter bank had 1/10 input

channels and 10/20 output channels respectively. After each convolution there was a maxpool with a window size and stride of two. The feedforward layers received an input that matched the output size of the convolutional layers (500) followed by a hidden layer of size 64 with an output size of 10. The hidden layer leveraged a ReLU activation function and dropout during training with a rate of 0.5. The CIFAR model was of a similar structure with the only differences being the number of channels in the convolutional layers which was increased to 3/50 and 50/50 and the size of the feedforward portion of the network receiving a 1800 dimensional vector as an input with a 1200 dimensional hidden layer, the output was kept to 10 units. All layers present, both feedforward and convolutional, omitted bias terms to allow for a standard conversion to a spiking neural network [1], this did not notably impact the overall performance of these networks. Model parameters are summarized in Table I.

	MNIST	CIFAR-10
Conv Channels	1, 10, 20	3, 50, 50
Filter Size / Stride	3x3 / 1	3x3 / 1
Maxpool Size / Stride	2 / 2	2 / 2
FF Layer Dims	500, 64, 10	1800, 1200, 10
Dropout	0.5	0.3

TABLE I: Network parameters

C. Sleep Replay Consolidation (SRC)

In short, SRC is applied by first converting a CNN to an SNN using a standard transformation [1], followed by simulated replay, during which unsupervised synaptic modifications occur. The altered SNN is then converted back into a CNN where the updated weights can be used in the conventional CNN forward pass.

In the SNN conversion, original network structure is preserved. A membrane potential (voltage) is simulated for each node in the network. Voltage is comprised of a running sum of inputs determined by presynaptic activity combined with the input weights and is subject to decay, effectively simulating dynamics of a leaky integrate and fire neuron. The ReLU activation is swapped for a Heaviside function to develop a notion of spikes. Once a neuron's membrane potential surpasses the given threshold, the neuron emits a spike and the voltage is reset to 0. To ensure that activity propagated across layers, layer wise scale factors to synaptic weights are generated in accordance with the Data-Based Normalization technique specified in [1] and multiplied by a hyperparameter coefficient. These modifications are applied to convolutional layer neurons, successfully converting CNN to SNN, while preserving network architecture and synaptic weight structure.

During the sleep phase, the SNN's activity is driven by randomly distributed Poisson spiking input with firing rates determined by the average values of each input pixel activation from the training data set. Hebbian style learning rules are applied to modify the weights: a weight is increased between two nodes when both pre- and post-synaptic nodes are activated and a weight is decreased when the post-synaptic node is activated but the pre-synaptic node is not. After this

Algorithm 1 : Sleep Replay Consolidation

```

1: procedure SLEEP( $n, I, scales, thresholds$ )  $\triangleright I$  is input
2: Initialize  $v$  (voltage) = 0 vectors for all neurons
3: for  $t \leftarrow 1$  to  $Ts$  do  $\triangleright Ts$  - Time step duration of sleep
4:    $S \leftarrow 0s$ 
5:    $S(1) \leftarrow$  Convert input  $I$  to Poisson-distributed spiking activity
6:    $S =$  ForwardPass( $S, v, W, scales, thresholds$ )
7:    $W =$  BackwardPass( $S, W$ )
8: end for
9: end procedure
10: procedure FORWARDPASS( $S, v, W, scales, threshold$ )
11: for  $l \leftarrow 2$  to  $n$  do  $\triangleright n$  - number of layers
12:    $\alpha \leftarrow scales(l-1)$ 
13:    $\beta \leftarrow threshold(l)$ 
14:    $v(l) \leftarrow \lambda v(l) + (\alpha * W(l, l-1) * S(l-1))$   $\triangleright W(l, l-1)$  - weights
15:    $\triangleright \lambda$  - decay rate
16:    $S(l)_i \leftarrow 1 \forall i$  where  $v(l)_i > \beta$   $\triangleright$  Propagate spikes
17:    $v(l)_i \leftarrow 0 \forall i$  where  $v(l)_i > \beta$   $\triangleright$  Reset spiking voltages
18: end for
19: return  $S$ 
20: end procedure
21: procedure BACKWARD PASS( $S, W$ )
22: for  $l \leftarrow 2$  to  $n$  do  $\triangleright n$  - number of layers
23:   if isConvolutionalLayer( $l$ ) then
24:      $F \leftarrow$  getConvolutionalFilters( $l$ )  $\triangleright$  All filters in layer  $l$ 
25:     for  $f$  in  $F$  do  $\triangleright$  Loop over all filters
26:        $L_f \leftarrow$  getFilterActivations( $f$ )  $\triangleright$  Pre/post activations for  $f$ 
27:       for  $(l_{f-}, l_{f+})$  in  $L_f$  do  $\triangleright$  For all input/output filters
28:          $S(l_{f-}) \leftarrow$  getSpikes( $f-$ ).  $\triangleright$  Presynaptic activity
29:          $S(l_{f+}) \leftarrow$  getSpikes( $f+$ ).  $\triangleright$  Postsynaptic activity
30:          $W^{(f)}_{i,j} \leftarrow$ 

$$\begin{cases} W^{(f)}_{i,j} + inc & \forall i, j \text{ where } S(l_{f+})_j = 1 \ \& \ S(l_{f-})_i = 1 \\ W^{(f)}_{i,j} - dec & \forall i, j \text{ where } S(l_{f+})_j = 1 \ \& \ S(l_{f-})_i = 0 \\ W^{(f)}_{i,j} & \text{Otherwise} \end{cases}$$

 $\triangleright$  Conv STDP
31:       end for
32:     end for
33:   else
34:      $W^{(l, l-1)}_{i,j} \leftarrow$ 

$$\begin{cases} W^{(l, l-1)}_{i,j} + inc & \forall i, j \text{ where } S(l)_j = 1 \ \& \ S(l-1)_i = 1 \\ W^{(l, l-1)}_{i,j} + dec & \forall i, j \text{ where } S(l)_j = 1 \ \& \ S(l-1)_i = 0 \\ W^{(l, l-1)}_{i,j} & \text{Otherwise} \end{cases}$$

35:      $\triangleright$  Linear STDP
36:   end if
37: end for
38: return  $W$ 
39: end procedure

```

	MNIST	CIFAR
No. of Time Steps (Ts)	222	10
Weight Multiplier ($scales$ coefficient)	2.78	46.81
Voltage Thresholds ($thresholds$)	[4.15, 9.47]	[7.00, 23.96]
Decay Rate (λ)	0.99	0.94
Synaptic Increase (inc)	$3.87 * 10^{-4}$	$6.52 * 10^{-4}$
Synaptic Decrease (dec)	$-3.13 * 10^{-4}$	$-1.98 * 10^{-4}$
Dt	0.001	0.001
Max Firing Rate	328.89	64.62

TABLE II: Hyperparameters used for SRC. Corresponding variable names as used in Algorithm 1 are within parentheses. Dt and the Max Firing Rate are used to generate input for the sleep stage.

unsupervised sleep period has been executed, the CNN model is restored by eliminating the simulated voltage, removing scale factors, and restoring the original activation functions. A pseudo code description of SRC is shown in Algorithm 1.

This approach can be directly applied to a fully connected network (as in [25]) since it produces one-to-one mapping from any pair of pre and post activations to the corresponding weights. However, implementing this to convolutional layers

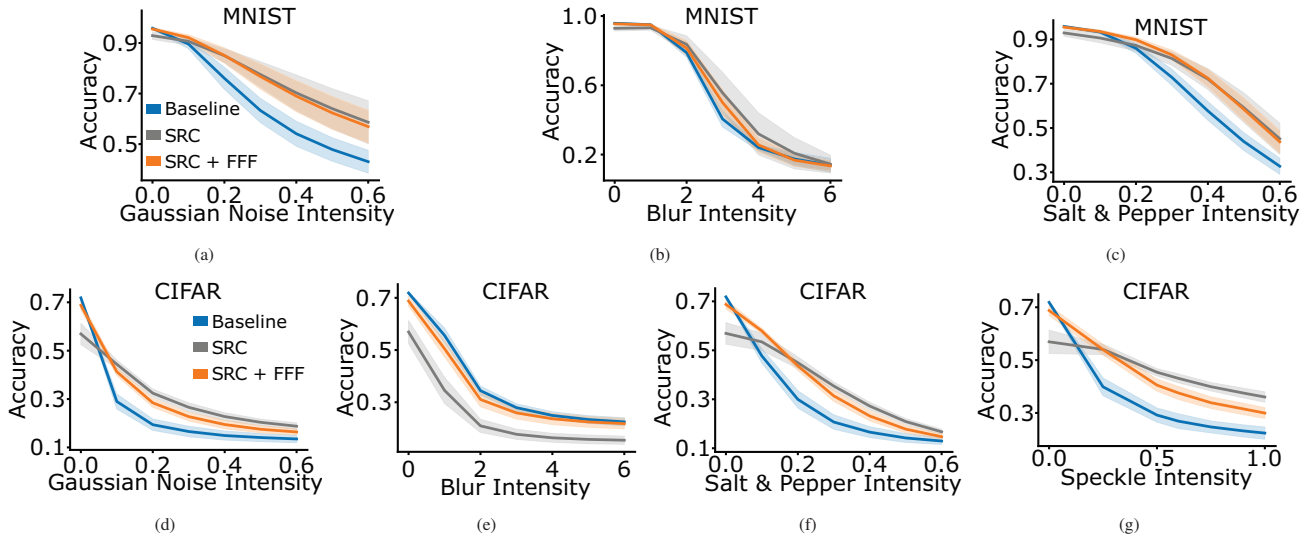


Fig. 2: MNIST (a-c) and CIFAR-10 (d-g) accuracy vs distortion intensity for Gaussian Noise, Blur, Salt & Pepper, and Speckle. Lines / shaded regions correspond to mean / standard deviation across trials. Note that the application of SRC notably improves performance on distorted inputs over baseline model.

is more complicated. Because of parameter sharing, a single weight may take part in multiple synaptic events. Thus, based on the network activity, we have an option of updating the same set of weights multiple times during a single iteration of SRC. Our implementation therefore accumulates synaptic updates over all activations that are associated to a given convolutional weight for every iteration.

The SRC hyperparameters were selected through the use of a standard python Genetic Algorithm implementation tasked to optimize mean validation performance over the Blur and Salt & Pepper distortions for a single trial. The optimal hyperparameters were used across trials to ensure no overfitting occurred, all the parameters are presented in Table II.

D. Experimental Design

All models underwent a standard training protocol. The naive MNIST / CIFAR model was trained for 50 epochs with a learning rate of 0.01 / 0.3 on the undistorted data set until a steady performance was reached. A binary cross entropy loss function along with a standard stochastic gradient descent optimizer was used to alter model parameters. Following baseline training the model underwent periods of SRC and subsequent Feedforward Fitting (Described in Section III-B). Each experiment below was repeated for 10 trials, each of 10 trials received a unique random seed causing differences in model weight initialization, training sample order, and SRC input noise generation.

III. RESULTS

A. SRC improves model performance on distorted data

Our initial set of experiments sought to explore whether SRC was capable of improving CNN generalizability over a variety of distortions for the MNIST and CIFAR-10 data sets. Ten trials (see Methods) were run using the baseline CNN model comprised of two convolutional and two feedforward

layers. This model was trained on clean unperturbed images until a plateaued mean performance of roughly 95% (MNIST) and 70% (CIFAR-10) accuracy on the undistorted data set.

The baseline model was tested across a variety of distortions, specifically additive Gaussian noise, Gaussian blur, Salt & Pepper, and Speckle (Speckle noise was excluded from MNIST as maximum intensity minimally degraded baseline performance) with results displayed in Figure 2. There was a direct and clear correlation between distortion intensity and baseline model performance (Figure 2a-g blue line). Increasing distortion intensity led to a significant drop in accuracy, sometimes to chance (see Figure 2b,d,f for intensities (6, 0.6, 0.6) respectively), as the substantial image distortions destroy convolutional feature representations which in turn causes the decision making layers to predict incorrectly.

After establishing the baseline, SRC was applied exclusively to the convolutional layers, as described above, and performance was tested again. We found clear improvement in overall model performance across a wide array of perturbation intensities (see Figure 2; note that the gray line is above the blue line in all cases except for (e)). Particularly for larger distortion values, SRC was capable of improving performance up to roughly 15% for MNIST (Figure 2a, difference between gray and blue) and 10% for CIFAR 10 (Figure 2g, difference between gray and blue). Since SRC weight modifications were only present in convolutional layers, the performance improvements suggest that filter robustness was increased as a result of SRC.

Overall SRC was able to improve performance across most distortion types. However, we found reduced generalizability to the blur distortion, especially for CIFAR 10 (Figure 2e). Although undesirable, it is in line with a variety of biologically inspired works where the given method is not always applicable to all perturbations [19], [6]. While other distortions are

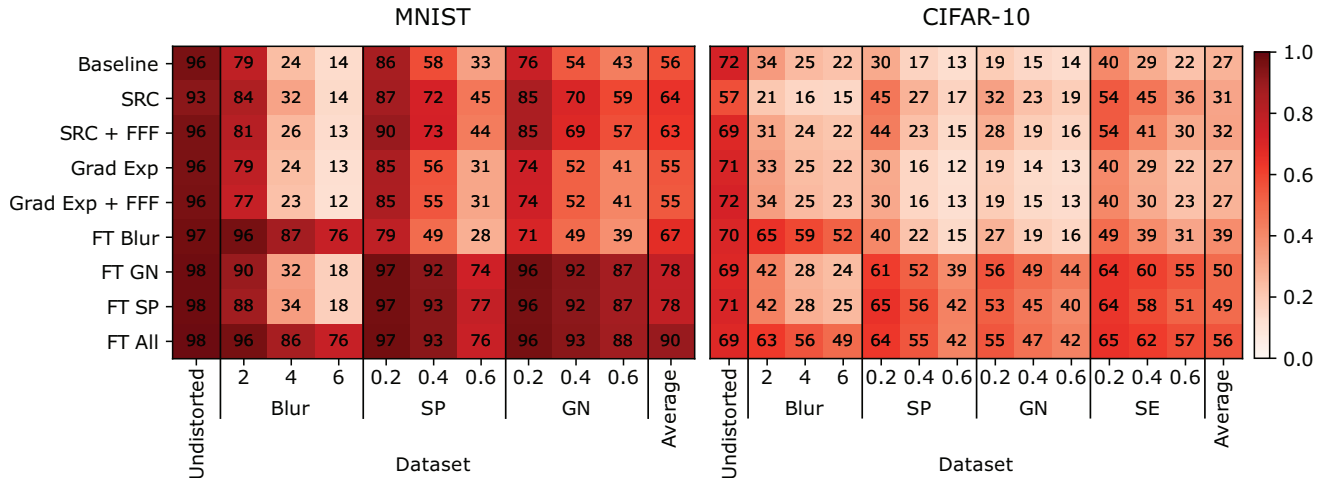


Fig. 3: Model performance on MNIST and CIFAR-10 with varying types and degrees of distortion. The unsupervised SRC phase significantly improved model performance on distorted inputs compared to the baseline while other naive unsupervised approaches (Gradient Expansion) fell short. Although fine-tuning on distortions can enhance performance, it requires extra data and can lack broad generalization.

comprised of pixel-wise perturbations, blurring by definition works on a greater spatial distance which makes it unique. SRC was able to slightly improve MNIST accuracy across greater blur intensities, suggesting that parameter modification may improve performance for blur distortion. It is important to note, the increase in robustness was achieved through a completely unsupervised learning technique which had no information about what specific types of distortions may be used for future testing.

B. Feedforward Fitting (FFF) recovers undistorted performance

While we found a clear improvement in performance on heavily distorted inputs following SRC, we also observed a drop in accuracy for minimally distorted and clean inputs on the order of 1.5% and 10% for MNIST and CIFAR-10 respectively (Figure 2a-g gray below blue for small distortion intensities). Although there may be circumstances where a general model that performs across a wide array of distortions is preferable to a model that performs well narrowly on clean inputs, clearly conserving undistorted performance is desirable. We hypothesised the drop in clean performance may result from a “miss-match” between convolutional and feedforward layers since only convolutional layers were modified by SRC. To test this, an additional training stage was implemented referred to below as Feedforward Fitting (FFF). Here the feedforward head of the network undergoes minimal training on the undistorted training data set; labels along with features extracted by the frozen convolutional weights are used to perform backpropagation on the feedforward layers only. This process thereby adjusts the decision making head of the network to the newly developed feature extractors formed after SRC.

FFF was applied until training set performance was saturated which took 1 / 5 epochs with a learning rate of 0.01 / 0.1 for MNIST / CIFAR. This regained lost performance on the minimally distorted data sets (Figure 2a-g, note orange line

near blue line for low distortion values) while significantly maintaining the performance gained for higher distortions (Figure 2a-g orange line near gray line for higher distortion values).

C. Fine-tuning Comparisons

The classic machine learning approach to gain model performance on new data distributions is fine-tuning (FT). Although this is an effective paradigm, it requires foresight of specific potential data perturbations and additional time to train the model. Nevertheless, it represents an ideal accuracy and is used as a benchmark. To compare our unsupervised SRC to this standard supervised method, we developed fine-tuned models each specializing in a specific distortion with one model specializing on all distortions. These fine-tuned models were first initialized using weights from the model trained on undistorted data. They then underwent 10 additional epochs of training (with learning rates of 0.05 / 0.15 for MNIST / CIFAR-10) using the specialized data set comprised of the undistorted data combined with varying levels of distortion from their expertise. The average accuracy across 10 trials for the fine-tuned models along with baseline, SRC, and SRC + FFF models is presented in Figure 3.

As anticipated, each fine-tuned network demonstrated outstanding performance on their respective perturbation, establishing a theoretical performance ceiling for these models on the corresponding distortions (Figure 3). We intuitively predicted fine-tuning on a specific distortion would lead to improved performance on that corresponding perturbation while showing no significant increase, or even a decline, in performance on other distortions. This pattern was evident for the MNIST model fine-tuned on blur which achieved optimal blur performance ranging from 96% - 76% across corresponding blur intensities 2 to 6, while performance on different distortions was below the baseline (Figure 3 left). Interestingly, when the MNIST model was fine-tuned on GN or SP, we observed a remarkable degree of transfer learning

to other distortions; all fine-tuned models for CIFAR-10 also demonstrated this high degree of transfer (Figure 3 right). The reason behind substantial transfer learning in these experiments was not immediately clear as other studies have suggested that this should not typically be the case [10]. While a certain degree of transfer learning between similar distortions might be expected, such as GN and SP (refer to Figure 1 for visualizations), the transfer between dissimilar distortions could be attributed to the simplicity of our data sets or the small size of our models which may act as a form of regularization.

Overall we found the fine-tuned models to be top performers in their respective domains, with the model fine-tuned on all distortions achieving the highest overall average accuracy. We also saw transfer learning proportional to degree of similarity between the trained and tested distortion types. SRC was able to outperform fine-tuned models on untrained distortions where little transfer learning was observed. When a high degree of transfer learning was present, the fine-tuned models outperformed SRC (e.g., fine-tuning on SP, GN and SE led to higher performance compare to SRC or SRC + FFF across distortions). However, it is important to note that the fine tuned models required a significantly higher degree of training. Specialized models were trained for an additional 10 epochs on a fine-tuning data set that contained seven times the number of training examples as in the original training set (one partition undistorted and 6 partitions of varying degrees of distortions). In contrast, SRC was able to increase generalizability with no additional data, highlighting the fact that SRC may also be a more efficient approach to increase model robustness when specifics of anticipated distortions are unknown.

D. Gradient Expansion

To gain insight as to why SRC is capable of improving model performance, we performed a weight analysis on the convolutional filters. Examining the spatial gradient of convolutional filters is often used as a metric for filter quality [9], [20], by inspecting the quality of filters across all convolutional blocks in the network we can determine the quality of the CNN. We developed a measure that is computed by simply taking the pixel-wise spatial gradient (for all filters in a given layer) and fitting a Gaussian probability distribution to their values, thereby obtaining a probabilistic representation for the filter gradients in each convolutional layer. We can examine the properties of this distribution, for instance the variance, to understand the estimated quality of convolutional blocks. A

	Baseline	Baseline + SRC	Baseline + GradExp
MNIST (C1)	$7.21 * 10^{-2}$	$1.47 * 10^{-1}$	$2.72 * 10^{-1}$
MNIST (C2)	$1.06 * 10^{-2}$	$4.36 * 10^{-2}$	$4.18 * 10^{-2}$
CIRAR (C1)	$1.48 * 10^{-1}$	$1.71 * 10^{-1}$	$1.83 * 10^{-1}$
CIFAR (C2)	$9.75 * 10^{-3}$	$1.04 * 10^{-2}$	$1.03 * 10^{-2}$

TABLE III: The mean standard deviation of spatial gradient variance across models. C1 and C2 refer to the results for the first and second convolution layer, respectively. We observe that both the SRC and GradExp models increase variance of the spatial gradient, however these changes are accompanied by a performance increase only in the SRC model.

narrow distribution would imply many repeated filters while a wider distribution would suggest a large variety of filters - this variability could enable rich feature extraction and therefore be beneficial for classification.

We noted that sleep increases the variance of the convolutional filter’s spatial gradient distribution across layers (compare first two columns in Table III). This can be interpreted as SRC producing more diverse and robust feature extractors through local activation patterns within the network and offers a possible explanation as to why sleep-like replay is capable of improving model performance across distortions.

	Baseline / SRC	GradExp / SRC
MNIST (C1)	$1.4984 * 10^{-1}$	$3.1373 * 10^{-2}$
MNIST (C2)	$3.0821 * 10^{-1}$	$7.6575 * 10^{-3}$
CIRAR (C1)	$6.4440 * 10^{-3}$	$2.8431 * 10^{-4}$
CIFAR (C2)	$8.7511 * 10^{-4}$	$2.5147 * 10^{-5}$

TABLE IV: KL divergence values between the baseline & SRC models (left column), and the Gradient Expansion (GradExp) & SRC models (right column). C1 and C2 refer to results for the first and second convolution layer respectively. Note that distributions on the right are much more similar than distributions on the left, displaying that the spatial gradient distributions of SRC and GradExp are similar - while both being different from the baseline.

To test if simply increasing the variance of filter spatial gradient magnitudes would increase performance, we artificially expanded the spatial gradients of the convolutional filters from the baseline model to approximate distribution of those in the SRC model (compare columns 1 and 3 in Table III). Thus, we choose a set of hyperparameters $\{\alpha_1, \dots, \alpha_L\}$ (see Table V for selected values), and increase the absolute value of all filter elements by that amount (Eq. 1). To account for layer specific weight statistics, we choose different α_l values for each layer to approximate changes observed following SRC:

$$W(l) = \begin{cases} W(l) + \alpha_l, & \text{if } W(l) \geq 0 \\ W(l) - \alpha_l, & \text{otherwise} \end{cases} \quad (1)$$

To ensure that these generated Gradient Expansion (GradExp) models have different spatial gradient distributions from our baseline model yet are similar to SRC models, we measured the KL divergence of the convolutional filter’s spatial gradient distributions for baseline vs. SRC and SRC vs. GradExp models (Table IV). We found a relatively high KL divergence between baseline and SRC (left column), signifying SRC is meaningfully modifying filters, and a relatively low divergence between SRC and GradExp models (right column) thereby verifying that our artificially generated spatial gradients are statistically similar to those achieved through SRC.

Two versions of the gradient expanded model were tested across distortion intensities for both MNIST and CIFAR-10. The first expanded convolutional filter gradients exclusively, the second applied Feedforward Fitting (FFF) to the network head (utilizing the same hyperparameters described in Section III-B) following filter gradient expansion to allow the decision layers to acclimate to the new feature extractors. Average MNIST and CIFAR-10 accuracy of these models across 10 trials is shown in Figure 3. Both variants of this model show no improvement over baseline (less than 1%) on any

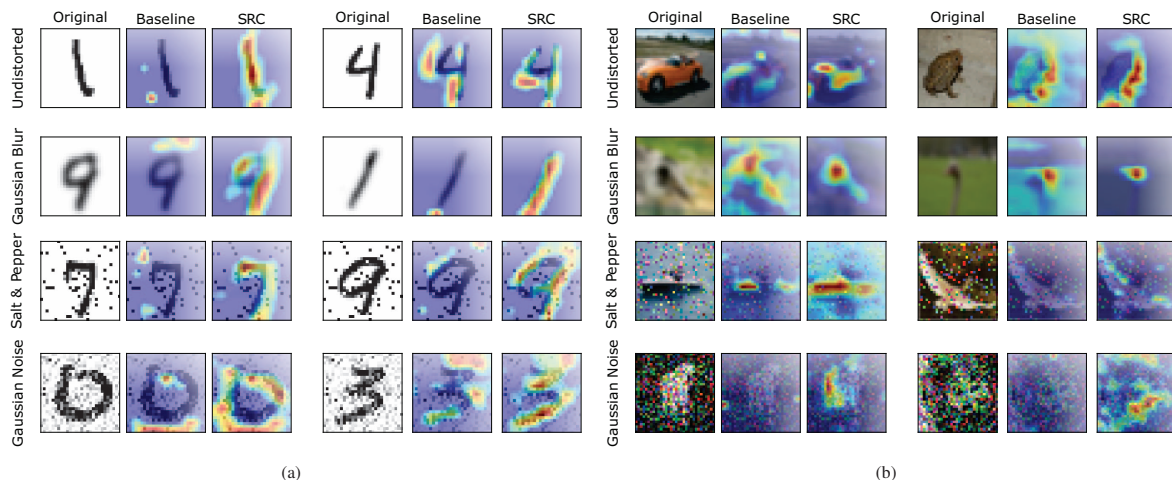


Fig. 4: Grad-CAM visualizations for MNIST (a) and CIFAR-10 (b) that display SRC improves attention quality over baseline model.

MNIST, Trial 1 - 10	
(C1)	[0.70 0.21 0.32 0.11 0.43 0.31 0.34 0.34 0.19 0.11]
(C2)	[0.11 0.09 0.09 0.10 0.08 0.14 0.08 0.07 0.12 0.11]
CIFAR, Trial 1 - 10	
(C1)	[0.035 0.050 0.050 0.080 0.070 0.060 0.075 0.060 0.065 0.040]
(C2)	[0.005 0.005 0.004 0.003 0.003 0.005 0.005 0.004 0.004 0.004]

TABLE V: Hyperparameters (α_l) for Gradient Expansion as described in Section III-D. We list values used for each of the 10 random trials. C1 and C2 refer to results for the first and second convolution layer respectively.

distortion intensity for either data set. This demonstrates that a general increase of the filter gradients is not sufficient to create robust filters resistant to input perturbations. This further suggests that SRC enables selective increases in the magnitude of convolutional spatial gradients. Additionally, the fact that applying FFF following gradient expansion does not increase performance shows that further feedforward training on equivalent quality convolutional filters is futile. Only if the feedforward head is allowed to train on higher quality convolutional blocks, like the ones developed in SRC, is there an improvement in distorted and undistorted performance.

E. Model attention and Grad-CAM analysis

To gain a deeper qualitative and quantitative understanding of how SRC impacts the network, analysis was developed using Gradient-weighted Class Activation Mapping (Grad-CAM) [21]. Grad-CAM is a visualization technique that creates an attention map for a given input to identify what the network focuses on. It operates by supplying an image as input and performing a forward pass followed by the calculation of gradients with respect to a given output label. Gradient values are then used to weight final convolutional activations (which maintain their spatial relevance), the intuition being more important features will have higher gradient values. This approach develops a notion of what input regions the network is attending to.

Generally speaking, we were able to observe improvements in attention as a result of SRC, some of the best examples

from both MNIST and CIFAR-10 are displayed in Figure 4 panels a and b, respectively. The results were particularly dramatic for MNIST. Given the original input image (Figure 4a, 1st and 4th column), the baseline model often attends to seemingly random pixels even on clear images (Figure 4a, 1st row, 2nd column). However, after SRC, model attention overlapped with the original input image significantly better (Figure 4a 3rd and 6th column). Importantly, SRC significantly enhanced attention on perturbed images. In the presence of noise the baseline model would often attend to noisy pixels or attention would be disrupted away from the original digit. Following SRC, the model was able to cut through the noise and the attention heat map took the shape of the original digit, implying the network is focusing on relevant features as opposed to irrelevant noise. A similar result was obtained for CIFAR-10 (Figure 4b) although the improvement was less consistent, some images displayed no improvement while others displayed clear benefit.

In an attempt to quantify attention improvements, we constructed a rudimentary metric that was compatible with the MNIST data set. The metric consisted of developing a pixel wise mask of the original digit (1's were assigned to input locations with nonzero pixel values and 0's everywhere else) followed by a cosine similarity between the mask and the attention vector output by Grad-CAM. Values close to 1 indicate a large overlap between the clean input image and the network's attention while values near 0 signify a misplaced network focus. This metric was averaged across all trials for every distortion / intensity combination for each model with the results displayed in Table VI. The overlap of attention and the original undistorted input digit is significantly higher for the model that underwent SRC when compared to the baseline or GradExp models. This implies the nontrivial selective filter gradient enhancement provided by SRC was able to improve convolutional filter quality and focus, even in the presence of meaningful perturbation; thereby increasing model performance.

Model	Baseline	SRC	SRC + FFF	Grad Exp	Grad Exp + FFF
Attention Overlap	0.145	0.229	0.193	0.146	0.150

TABLE VI: Grad-CAM Attention Overlap Metric. It can be seen that the SRC increases attention overlap with the ground truth image over baseline. Gradient Expansion models also increase accurate attention but without the performance benefit seen with SRC.

IV. CONCLUSION

In this work we developed a biologically inspired sleep-like optimization stage, termed the Sleep Replay Consolidation (SRC) algorithm, and showed it is compatible with CNNs and capable of improving convolutional filter quality thereby increasing model performance on distorted data sets. We examined SRC on standard image classification data sets, MNIST and CIFAR-10, and found that it substantially improves performance for moderate to high levels of distortion intensity. We further identified mechanisms of improvement as related to non-linear selective expansion of the convolutional filter's spatial gradient distribution across layers. Our study, combined with previous work [24], [25], suggests that sleep-like unsupervised replay may provide multiple benefits to different classes of ANNs, including improving continual learning, generalization and adversarial robustness.

REFERENCES

- [1] DIEHL, P. U., NEIL, D., BINAS, J., COOK, M., LIU, S.-C., AND PFEIFFER, M. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International Joint Conference on Neural Networks (IJCNN)* (2015), IEEE, pp. 1–8.
- [2] DIEKELMANN, S., AND BORN, J. The memory function of sleep. *Nature Reviews Neuroscience* 11, 2 (Jan. 2010), 114–126.
- [3] DODGE, S., AND KARAM, L. Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)* (2016), IEEE, pp. 1–6.
- [4] DODGE, S., AND KARAM, L. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)* (2017), IEEE, pp. 1–7.
- [5] DONLEA, J. M., THIMGAN, M. S., SUZUKI, Y., GOTTSCHALK, L., AND SHAW, P. J. Inducing sleep by remote control facilitates memory consolidation in *idrosophila*. *Science* 332, 6037 (June 2011), 1571–1576.
- [6] EVANS, B. D., MALHOTRA, G., AND BOWERS, J. S. Biological convolutions improve dnn robustness to noise and generalisation. *Neural Networks* 148 (2022), 96–110.
- [7] FEL, T., RODRIGUEZ RODRIGUEZ, I. F., LINSLEY, D., AND SERRE, T. Harmonizing the object recognition strategies of deep neural networks with humans. *Advances in Neural Information Processing Systems* 35 (2022), 9432–9446.
- [8] FUKUSHIMA, K., AND MIYAKE, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.
- [9] GAVRIKOV, P., AND KEUPER, J. Cnn filter db: An empirical investigation of trained convolutional filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 19066–19076.
- [10] GEIRHOS, R., TEMME, C. R. M., RAUBER, J., SCHÜTT, H. H., BETHGE, M., AND WICHMANN, F. A. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems* 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 7538–7550.
- [11] JI, D., AND WILSON, M. A. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience* 10, 1 (2007), 100–107.
- [12] KIRKPATRICK, J., PASCANU, R., RABINOWITZ, N., VENESS, J., DESJARDINS, G., RUSU, A. A., MILAN, K., QUAN, J., RAMALHO, T., GRABSKA-BARWINSKA, A., ET AL. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.
- [13] KRIZHEVSKY, A., AND HINTON, G. Learning multiple layers of features from tiny images, 2009.
- [14] LECUN, Y., BENGIO, Y., ET AL. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.
- [15] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.
- [16] LEWIS, P. A., AND DURRANT, S. J. Overlapping memory replay during sleep builds cognitive schemata. *Trends in cognitive sciences* 15, 8 (2011), 343–351.
- [17] MELNATTUR, K., KIRSZENBLAT, L., MORGAN, E., MILITCHIN, V., SAKRAN, B., ENGLISH, D., PATEL, R., CHAN, D., VAN SWINDEREN, B., AND SHAW, P. J. A conserved role for sleep in supporting spatial learning in *idrosophila*. *Sleep* 44, 3 (Sept. 2020).
- [18] RASCH, B., AND BORN, J. About sleep's role in memory. *Physiological Reviews* 93, 2 (Apr. 2013), 681–766.
- [19] SAFARANI, S., NIX, A., WILLEKE, K., CADENA, S., RESTIVO, K., DENFIELD, G., TOLIAS, A., AND SINZ, F. Towards robust vision by multi-task learning on monkey visual cortex. *Advances in Neural Information Processing Systems* 34 (2021), 739–751.
- [20] SCHUBERT, L., VOSS, C., CAMMARATA, N., GOH, G., AND OLAH, C. High-low frequency detectors. *Distill* 6, 1 (2021), e00024–005.
- [21] SELVARAJU, R. R., COGSWELL, M., DAS, A., VEDANTAM, R., PARIKH, D., AND BATRA, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 618–626.
- [22] STICKGOLD, R. Sleep-dependent memory consolidation. *Nature* 437, 7063 (2005), 1272–1278.
- [23] SZEGEDY, C., LIU, W., JIA, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1–9.
- [24] TADROS, T., KRISHNAN, G., RAMYAA, R., AND BAZHENOV, M. Biologically inspired sleep algorithm for increased generalization and adversarial robustness in deep neural networks. In *International Conference on Learning Representations* (2019).
- [25] TADROS, T., KRISHNAN, G. P., RAMYAA, R., AND BAZHENOV, M. Sleep-like unsupervised replay reduces catastrophic forgetting in artificial neural networks. *Nature Communications* 13, 1 (2022), 7742.
- [26] TETI, M., KENYON, G., MIGLIORI, B., AND MOORE, J. Lcanets: Lateral competition improves robustness against corruption and attack. In *International Conference on Machine Learning* (2022), PMLR, pp. 21232–21252.
- [27] VASILJEVIC, I., CHAKRABARTI, A., AND SHAKHNAROVICH, G. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760* (2016).
- [28] WALKER, M. P., AND STICKGOLD, R. Sleep-dependent learning and memory consolidation. *Neuron* 44, 1 (2004), 121–133.
- [29] WEI, Y., KRISHNAN, G. P., AND BAZHENOV, M. Synaptic mechanisms of memory consolidation during sleep slow oscillations. *Journal of Neuroscience* 36, 15 (2016), 4231–4247.
- [30] ZHOU, Y., SONG, S., AND CHEUNG, N.-M. On classification of distorted images with deep convolutional neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), IEEE, pp. 1213–1217.
- [31] ZWAKA, H., BARTELS, R., GORA, J., FRANCK, V., CULO, A., GÖTSCH, M., AND MENZEL, R. Context odor presentation during sleep enhances memory in honeybees. *Current biology : CB* 25, 21 (November 2015), 2869–2874.