On the Effect of Unsupervised Regularization for Image Classification

Nikhilesh Belulkar, Mitali Juneja, Tristan Saidi, Sagarika Sharma
Department of Computer Science
Columbia University
New York, NY 10025
nb2953@columbia.edu
mj2944@columbia.edu
tls2160@columbia.edu
ss5946@columbia.edu

Abstract

In this project we explore the effect of unsupervised regularization in traditional classification problems, specifically image recognition. We explore the effect of adding a reconstructive decoding head to a standard convolutional neural network to create a supervised autoencoder (SAE). This allows gradients from the decoder to backpropagate through the network and influence the weights of the early stages of the convolutional network. We tested the results of this SAE under a variety of circumstances while varying aspects of the model greatly. We tested the classification performance of the model in the face of varying levels of noise, and we experimented with different weightings between reconstruction and classification losses. We also studied the SAE's performance when given largely unlabeled data, looking to see if training the reconstructive head of the network on large amounts of unlabeled data improved classification performance when given only a small amount of labeled data.

1 Introduction

Convolutional neural networks (CNNs) are standard models for image classification. However, they suffer from a few shortcomings, two of which we list below:

1. Without proper regularization techniques, they can be prone to overfitting.
2. As is typical of most neural techniques, in the absence of a pre-trained initialization for network parameters, a large number of labeled examples is needed to learn a proper mapping between the inputs and their desired outputs.

Here, we propose a modification of traditional image classification CNNs: the addition of a decoding head, that reconstructs the input image using an intermediate latent representation of the model (Fig. 1), thereby creating a supervised autoencoder (SAE).

Our motivation is twofold. First, we believe that the resulting autoencoder portion of the network encourages learning latent representations that are more generalizable, since they must be suited for not only image classification but also image reconstruction. In other words, the addition of an image reconstruction task explicitly forces the latent representations of the network to capture salient features of the input image, effectively serving as a form of regularization. Second, the autoencoder portion of the network can be trained in an unsupervised manner, using unlabeled examples. As a result, the weights in the encoder portion of the network can be optimized, to some extent, without the

I Can't Believe It's Not Better Workshop III at NeurIPS 2022.

use of any labeled examples. Ideally, one would expect this to further increase image classification accuracy for the SAE over a traditional CNN, which cannot make use of the information contained in unlabeled examples.

In this project, we experiment with the SAE architecture on a standard image classification dataset (CIFAR-10), in order to gain a better understanding of the merits of this approach. Specifically, we study the responses of the SAE model in a variety of experimental settings, including (1) adding noise to our input images (forcing the autoencoder to learn a latent space useful for reconstructing a denoised version of the input, which can help provide robustness in image classification as well), (2) the relative number of unlabeled and labeled examples used during training, and (3) the relative weights given to the losses of the two tasks that the SAE performs (reconstruction and classification).

In section 3, we provide details on the architecture of our SAE, as well as our experimental baseline. We also describe the optimization procedure we use for the SAE, taking into account the losses from the two individual tasks. We also provide more details about the experiments we conduct. In section 4, we present and analyze the results of our experiments.

The contributions of our project are as follows:

1. We demonstrate that the SAE architecture can provide more robustness in image classification. Specifically, in the face of a shift in the type and levels of noise seen during training and testing, the SAE outperforms its corresponding standard CNN.
2. We find that contrary to intuition, there is no clear relationship between classification accuracy and the relative weight given during training to the reconstruction loss vs. the classification loss.
3. Our experiments generally demonstrated that, with the exception of a few levels and types of input noise, the SAE did not provide much gain in image classification accuracy over a standard CNN. We take this to be an indication that the information needed to classify an image is quite unlike the information needed to reconstruct it, and, consequently, the addition of an image reconstruction task was not able to improve image classification performance.

2 Related Works

Our paper develops upon ideas from other papers including supervised auto-encoders, unsupervised pre-training and convolutional neural networks.

In our project we utilize a SAE to create denoised latent embeddings that can then be used for classification of noisy inputs. For images, this consists of a series of convolutional layers to down-sample the data to a latent representation. This is followed by a series of convolutional layers that up-sample that representation, bringing it back to the size of the original input image, so that a reconstruction loss can be computed. We utilize the base SAE architecture proposed in [LPW18] in our paper.

To construct our baseline, which we use to benchmark our SAE, we develop a standard convolutional neural network architecture for image classification. The CNN architecture we utilized was pioneered by Yann LeCun in the construction of LeNet for handwritten digit recognition [LBBH98]. In the construction of our network we use similar techniques to classify images, including a series of convolutional layers and subsampling layers that utilize maxpooling, followed by a series of fully connected layers. This CNN architecture is used to benchmark the classification performance of the SAE in noisy settings.

A few research groups have previously studied the idea of using autoencoders to improve classification of noisy image data. A Canadian research group worked on stacking several autoencoders to obtain better classification performance [VLL$^+$10]. This paper tested noisy classification with salt & pepper noise as well as blackout noise. In our project, we investigated using a simpler autoencoder structure, to see if there were any interesting results to be learned in a minimalist setting (without using stacked autoencoders). The stacked approach has also been further experimented with by [RHAM18], who compared a stacked supervised autoencoder (using multiple autoencoders) to an architecture in which only one autoencoder was used. However, this paper utilized different types of noise (Gaussian and binomial) than the types we explore. Our project differs, in that we explore the usefulness of the SAE architecture (comprised of a single autoencoder) with zero-out noise.

Further, other groups have explored how to perform noisy image classification using traditional, non-neural techniques, such as SVMs [dCCN+16]. Interestingly, these groups found that feature extraction methods for handling noisy image data including LBP (local binary pattern) and HOG (histogram of oriented gradients) produced better image classification results than SVM baselines when both training and testing data was corrupted with Gaussian, Poisson and salt & pepper noise. This is in contrast to our results, which found no difference in SAE performance and CNN baseline performance when both training and testing data was corrupted with similar noise, and probably speaks to how CNN-based methods for image classification are inherently denoising.

Researchers at UMontreal [EBC+10] reference that architectures using unsupervised pre-training have better generalization. They argue that unsupervised pre-training guides learning towards minima that support better generalization from the training data set. The paper argues that such unsupervised training also improves performance for noisy data. Furthermore, another paper by the same research group [EMB+09] seeks to quantify how effective unsupervised pre-training can be for different depth models. Their results show that unsupervised pre-training is more beneficial and acts as a regularizer for deep models; however it harms generalization performance for shallower models.

This lends evidence that an SAE architecture could be better than a standard CNN for the classification of noisy data in specific settings.

3 Method

3.1 Supervised Autoencoder Model

In order to achieve a SAE, a model that both reconstructs and classifies an input image, we utilized the architecture depicted in Fig. 1. Similar high level architectures have been utilized in the past [LPW18], but ours utilizes different layers and parameters. At a high level, our model consists of three main components. A convolutional encoder stage maps the input image (scaled so that all values are between 0 and 1) $x \in [0,1]^{32\times32\times3}$ to a latent representation $\psi \in \mathbb{R}^{2560}$. This is achieved by a series of convolutional layers, batch normalizations, and rectified linear unit nonlinearities. This latent representation ψ is then passed to both a decoder network (which does autoencoder reconstruction) and an image classification network. The decoder alternates between convolutional layers, batch normalizations, upsampling, and rectified linear unit nonlinearities, mapping ψ to the reconstructed image $\hat{x} \in [0,1]^{32\times32\times3}$. Note that the final output of the decoder is passed through a sigmoid activation function, in order obtain a scaled output image, with all values $\in [0,1]$. The classifier network uses a single linear layer followed by a softmax operation to map ψ to $\hat{y} \in \mathbb{R}^{10}$, a probability distribution over the 10 classes.

$$\begin{aligned} x \in [0,1]^{32\times32\times3}, f_{enc} : x \mapsto \psi \in \mathbb{R}^{2650} &\text{ , where } \psi \text{ is an embedded representation} \\ \psi \in \mathbb{R}^{2650}, f_{dec} : \psi \mapsto \hat{x} \in [0,1]^{32\times32\times3} &\text{ , where } \hat{x} \text{ is the reconstructed representation} \\ \psi \in \mathbb{R}^{2650}, f_{class} : \psi \mapsto \hat{y} \in \mathbb{R}^{10} &\text{ , where } \hat{y} \text{ is a probability distribution over the classes} \end{aligned} \tag{1}$$

In certain cases, we applied zero-out noise to the input image. With this type of noise, we randomly mask out a fraction, ϵ, of the input example's elements with zeros. This gives us a noised input vector $\tilde{x}$ that takes the place of x in eq.1.

For each labeled training example, we compute the following loss function during training, which optimizes for both the image classification and reconstruction tasks.

$$\mathcal{L}_{glob} = \mathcal{L}_{class} + \beta\mathcal{L}_{rec} \tag{2}$$

β controls the relative importance of the reconstruction loss in comparison to the classification loss. Here the classification loss is denoted by $\mathcal{L}_{class} = \sum_{y_i \in y} y_i \cdot log(p_i)$, the cross entropy loss between the predicted distribution over outputs $\hat{y}$ and the ground truth labels y, while $\mathcal{L}_{rec} = ||x - \hat{x}||_2^2$ is the mean squared error between the reconstructed image $\hat{x}$ and the original input x. In the instances where training data was corrupted with noise, the mean squared error was still calculated between the reconstructed image and the original, de-noised image x.

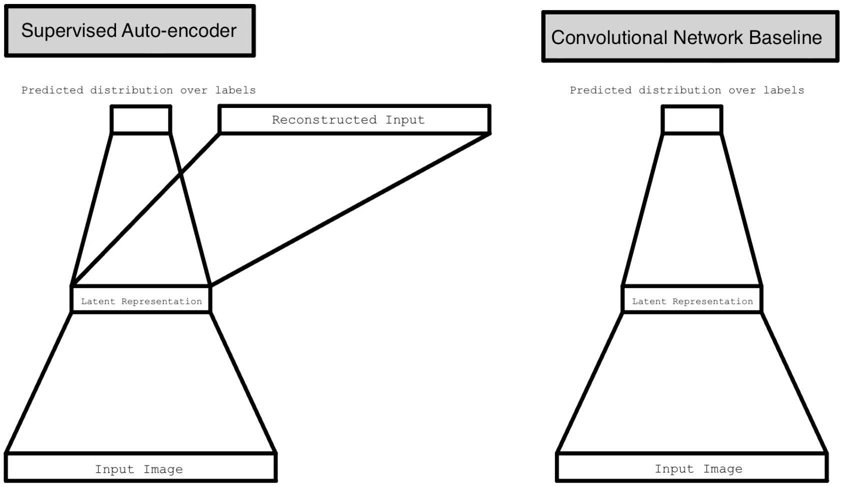

Figure 1: Visualization for Supervised Autoencoder (left) and Convolutional Baseline Network (right)

The SAE model was used in a variety of experimental settings, as outlined in section 4.

3.2 Baseline Network

As a baseline, we used a standard convolutional network derived from the architecture of our SAE. The model consists of an encoder of the same architecture $f'_{enc} : x \mapsto \psi \in \mathbb{R}^{2650}$ and the same classifier layer $f'_{class} : \psi \mapsto \hat{y} \in \mathbb{R}^{10}$. A visualization of the architecture can be seen in Fig. 1.

3.3 Experiments and Testing

Our experiments sought to explore the benefits and limitations of our model in a variety of contexts. The original motivation behind our SAE model was twofold, and this guided our experimentation process.

1. **Denoising Properties:** We believed that the additional reconstruction loss of the SAE would encourage representations that hold onto as much information about the original image as possible. We surmised that this would provide benefit at test time in the face of new or more extreme types of noise. We tested this theory by experimenting with noised input images, as well as shifts in the types and levels of noise seen at train vs. test time.
2. **Unsupervised training:** This network architecture was also partially motivated by the challenge of acquiring large sets of labelled data. In this situation, the SAE can be trained both on the available labelled data as well as any unlabeled data that might be available. We hoped that this would provide a boost to classification performance when small amounts of labeled data were provided.

The details of our experiments can be decomposed into four different categories. Each category either varies parameters of the SAE model, or explores one of the aforementioned assumptions about the nature and benefits of an SAE.

1. **Types of Noise:** We experimented with adding zero-out noise (in which some percentage of the input pixels are blacked out) to the input image. We sample and add the noise each time an example appears during training. The classification label remains the same as for the original image. For reconstruction, we compute the loss against the original image, which forces the autoencoder to denoise its input, potentially encoding the salient features of the image in its latent space.
2. **Loss weights:** We experimented with different weights for the reconstruction and classification losses. We generally observed the reconstruction loss to be multiple orders of

magnitude smaller than the classification loss, indicating that the gradients from classification would likely be much stronger than the gradients from reconstruction. To compensate, we experimented with increasing the weight on the reconstruction loss, by simply applying a scalar multiplier on the computed reconstruction loss.

3. **Number of labeled vs unlabeled examples:** Given that the reconstruction task is unsupervised, it can be trained using unlabeled examples. This motivated us to experiment with using some number of unlabeled examples, seen only by the autoencoder portion of the SAE. Our intuition was that this might help the encoder learn useful representations that capture salient information to the image, and provide a suitable "initialization" for our supervised classification task. This is related to the idea of transfer learning, and should help improve downstream classification accuracy and generalization when there are very few labeled examples available.

4. **Pre-training vs iterative training:** In the experiments where we used unlabeled examples to train the autoencoder portion of the SAE, we tried 2 different training strategies. The first follows traditional transfer learning approaches – we first train the autoencoder using the unlabeled examples for some number of epochs (eg. 40). Then, using this learned initialization, we train using the labeled examples, optimizing for classification. However, this has the downside that over the course of the supervised training phase, the encoder's weights can increasingly deviate from their pre-trained values, potentially eliminating any effect that the denoising task could have on classification performance and generalization. To test this theory, we also experimented with an iterative form of training, where in *each* epoch, we fed one batch of unlabeled examples, followed by a second batch of labeled examples. This effectively iterates between training for reconstruction and training for classification in each epoch. The intuition for this approach is that it prevents the encoder's weights from "forgetting" the denoising task, and reinforces the denoising gradients throughout training. Ultimately, if the denoising task has any impact on classification generalization, we hypothesized that it would be more apparent in the iterative training setting.

4 Results

4.1 Zero-out noise

We experimented with the robustness of the SAE model by looking at its performance in the face of challenging types of noise. In particular, we played with the introduction of zero-out noise, where a certain percentage (set by the parameter ϵ) of the entries in the input tensor of the image are set to 0. As an example, $\epsilon = 0.3$ yields the corruption that can be seen in Fig. 2.

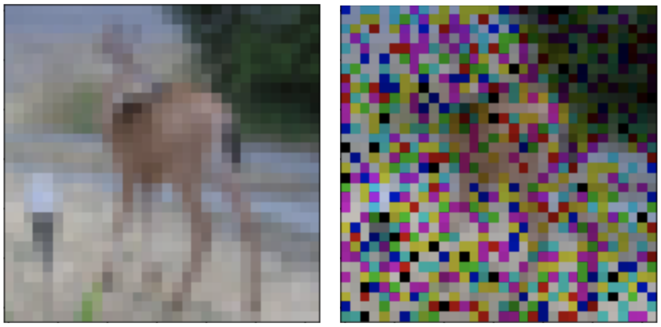

Figure 2: Original (left) and corrupted (right) versions of the same image

Our experiments related to this type of noise took on two different forms. We first trained both our convolutional baseline and our SAE on CIFAR-10 using the original input images (i.e., $\epsilon = 0$). At test time, we experimented with several values of ϵ, as seen in Fig. 3. To compare the performance of our networks, we passed both the baseline and our SAE a test set with varying levels of zero-out noise (i.e., a variety of ϵ values). Using this approach, we found that the SAE model consistently outperforms the baseline by about $3 - 5\%$, especially as the value of ϵ was increased.

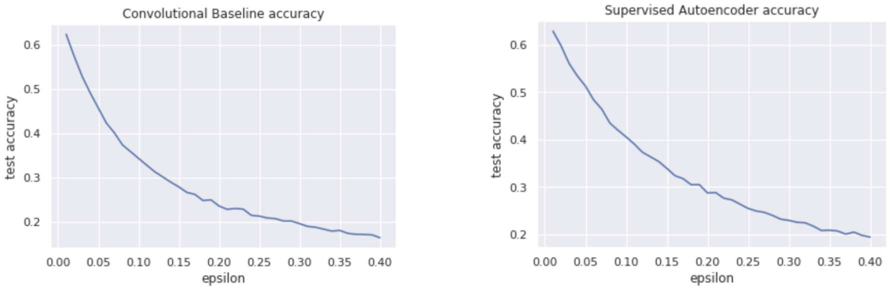

Figure 3: Convolutional baseline network (top) and SAE (bottom) performance with varying levels of test time zero-out noise

We then proceeded to incrementally add zero-out noise to the train set to see how this affected the relative performance of the baseline and the SAE. We varied both training and testing ϵ to explore the relationship between the performances of the two models, as seen in Fig. 4. Clearly the two models perform similarly when the test time ϵ matches training ϵ. However, the SAE pulls ahead when training and test ϵ diverge, with an absolute performance accuracy of around $3 - 5\%$. This provides interesting insight about the nature of our model. Under situations in which test time noise is unlike or far more extreme than the noise seen at training time, the SAE performs better. We presume that this boost in performance is due to the incentive for the SAE model to retain information about the input that is relevant for reconstruction of the original, de-noised image. This gives the network a test-time advantage over a standard convolutional network, when neither model has seen the type or severity of the noise at hand.

4.2 Loss weights

Varying β, the relative weight of reconstruction and classification loss in eq. 2 affects the degree to which the SAE learns to better reconstruct the image, or to better classify it. We initially hypothesized that the gradient signals from the decoder (reconstruction) network would enhance classification performance. But to our surprise, it seemed that the two tasks (reconstruction and image classification) seem to compete and hinder each other in some cases.

As a benchmark for comparison, we ran the convolutional baseline network on the entire CIFAR-10 dataset. The training curves and test time performance (with varying levels of zero-out noise) can be seen in Fig. 5. We tested the SAE in the same manner, additionally varying the weight of the reconstruction loss by modifying β in eq. 2. For the SAE, we will note that all training was done without the presence of zero-out noise (i.e., $\epsilon = 0$).

The results for various β values can be seen in Fig. 6. To our surprise, no real pattern emerged as we varied β. All training curves follow similar trajectories, and no choice of β appears as a standout, superior choice. Furthermore, we observed that robustness to zero-out noise seen at test time was not clearly related to β. $\beta = 10$ provided the largest improvement over the convolutional baseline's robustness to test time zero-out noise, with an absolute advantage of about 3% in the face of extreme corruption of the input image ($\epsilon = 0.4$). But overall, for standard classification with no input noise ($\epsilon = 0$), SAE's use of reconstruction loss does not seem to help classification performance. This reinforces the findings from the previous results section: SAE's main performance benefits come in the face of unseen test-time noise.

4.3 Labeled vs. unlabeled data splits

Motivated by the potential of the architecture to improve generalization of classification with limited labeled data but abundant unlabeled data, we varied the labeled and unlabeled data split used during training of the networks. In this case, the unlabeled data is used to train f_{enc} and f_{dec} only, whereas labeled data is used to train the f_{enc} and f_{class}.

At the beginning of the training procedure for each of these networks, the CIFAR-10 dataset is split into a 10% validation set. We also remove labels from some percentage — as used in experiment, 95%, 90%, 80%, and 50% — of the remaining training dataset. For each of these training data splits,

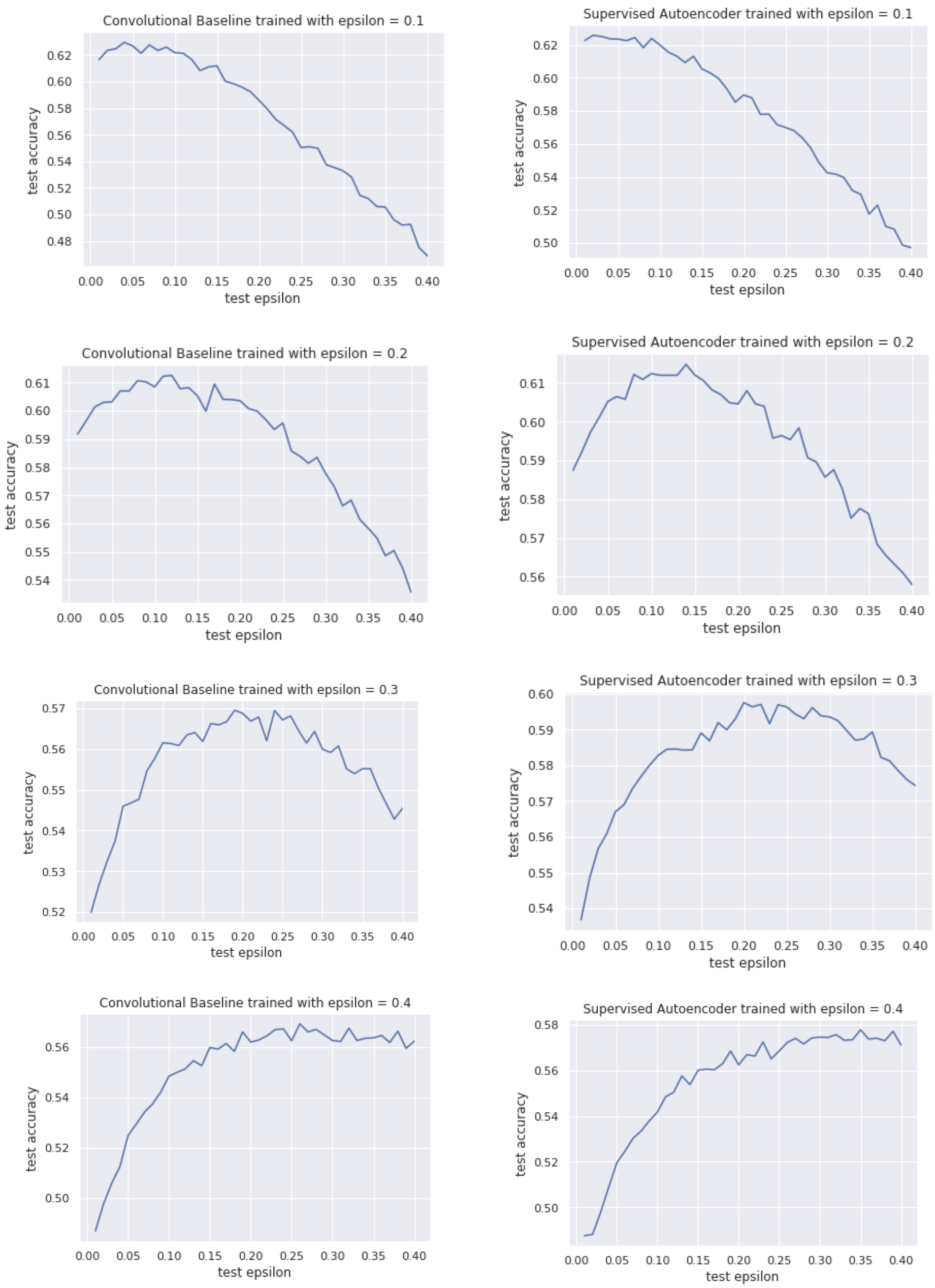

Figure 4: Convolutional baseline network (top) and SAE (bottom) performance with varying levels of test time and train time zero-out noise

the SAE is first trained on the unlabeled data for 40 epochs on the reconstruction task only. No weight updates are made to f_{class} during this stage. Following this, we use the labeled data to train on the image classification task for 40 epochs, with no updates made to f_{dec}. For our convolutional baseline network, we train solely on the labeled portion of the training data for 40 epochs. Note that, unlike previous methods, no noise is added during training time. For both networks, we train using the Adam optimizer, with a learning rate of 1.0×10^{-4}.

Fig. 7 displays our results for these settings. For both networks, it is apparent that as a larger portion of labeled training data is used, the classification accuracy on the validation set increases. However, the SAE does not demonstrate improved generalization performance (in any labeled vs. unlabeled

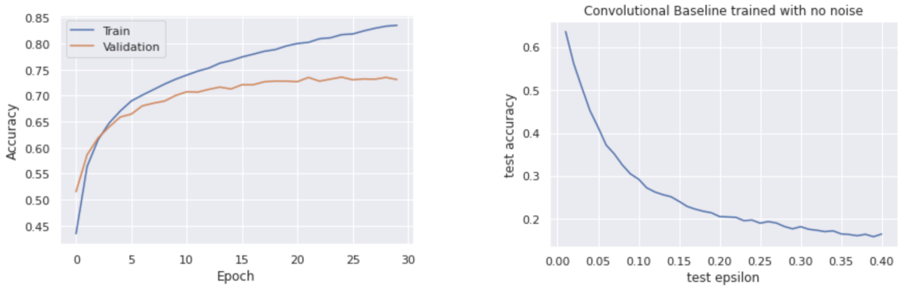

Figure 5: Convolutional Baseline Network trained on CIFAR-10 examples, testing on varying levels of zero-out noise

split) in comparison to the convolutional baseline. Instead, it shows rather similar generalization performance, outperforming slightly in some splits and underperforming slightly in others. This is contrary to expectation, as we hypothesized that the SAE model would learn salient features of classes from the comparably large amount of unlabeled data and transfer this learning, yielding improved performance on the classification task.

In a similar manner to our previous experiments, we conducted an additional analysis of the robustness of each of these models, by measuring classification accuracy on test data corrupted with increasing levels of zero-out noise. The results displayed in Fig. 8, suggest that the SAE, for any split of the training data, is still able to outperform the convolutional baseline network across all noise levels we tested. The accuracy of both models also increases with the portion of labeled data seen during training time, as expected.

4.4 Iterative unsupervised and supervised training

The results of our above experiments suggest that the SAE does not improve generalization on classification tasks. Our previous hypothesis was motivated by a potential positive effect of image reconstruction information on classification. However, we can alternatively view image reconstruction as simply a form of regularization for image classification, that can help improve image classification generalization. We investigated this new viewpoint, by testing whether an iterative training procedure can improve generalization, due to a potential regularization effect.

The training procedure for this experiment is largely similar to the training procedure used in section 4.3. The sole deviation is the order in which the supervised autoencoder sees the unlabeled and labeled data — now, in *each* training epoch, there is first a pass to train for reconstruction (using a batch of unlabeled data) and then a pass to train for classification (using a batch of labeled data).

The results for the training and validation accuracies of each of the models can be seen in Fig. 9. It is evident, once again, that the generalization performance of the SAE network and the convolutional network trained with the same amount of labelled data is comparable. The SAE model with iterative unsupervised and supervised training seems to overfit to the training data, given the incredibly high training accuracy and increasing gap between the training and validation accuracy over the course of training.

Finally, following all of our previous experiments, an additional analysis was performed on the robustness of the SAE and convolutional networks. We train both models on the same amount of labelled data. Our procedure exactly follows that described in section 4.1. Our results are displayed in Fig. 10. The SAE model performs comparably to the convolutional baseline model. It also consistently has worse test accuracy in comparison to the SAE model with separate unsupervised and supervised training periods (akin to pre-training and transfer stages). Once again, the classification accuracy of each model in the robustness tests increases with the amount of labeled data used during training.

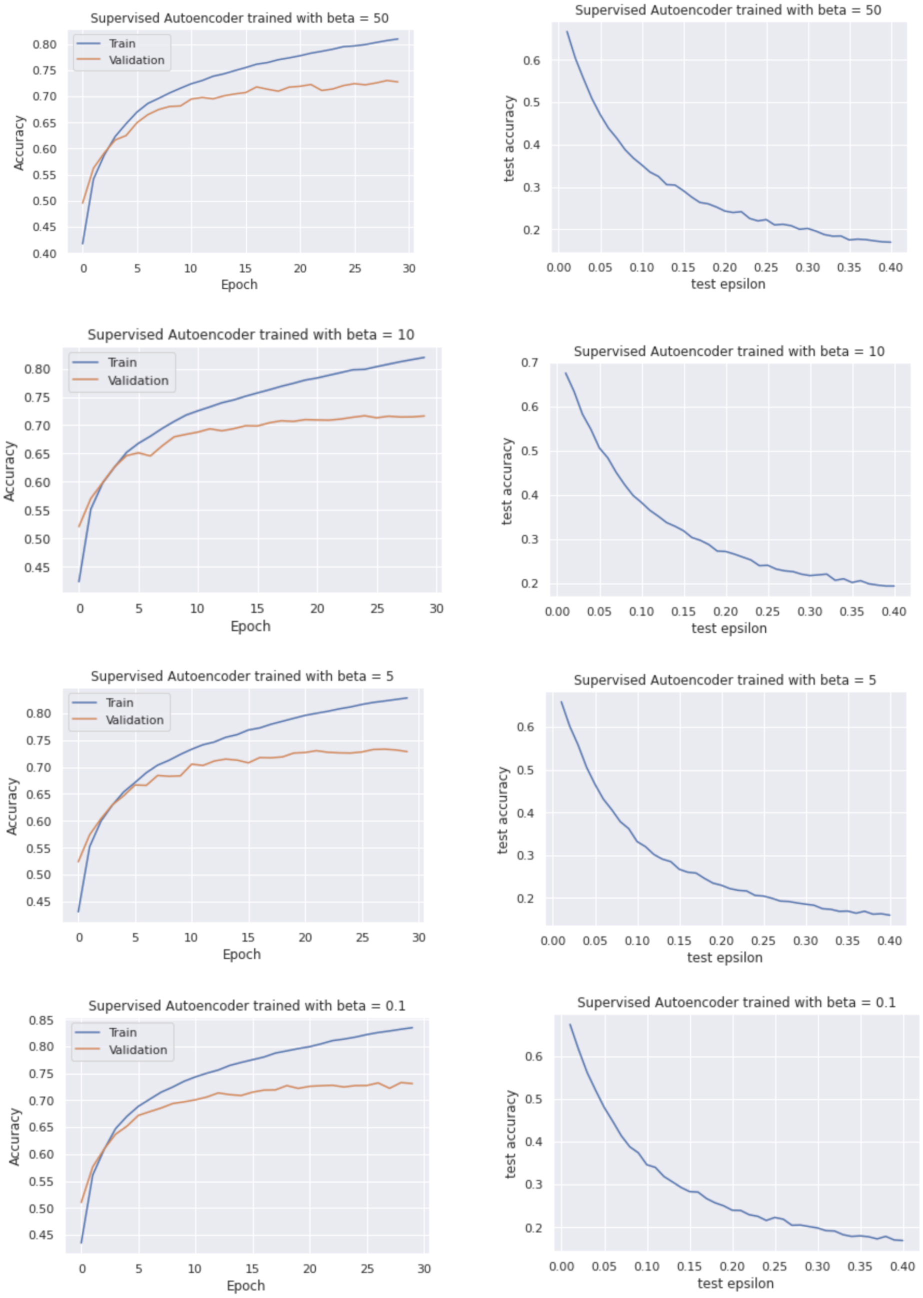

Figure 6: SAE trained with varying β, with varying zero-out noise applied to trained model

5 Discussion

Conclusion: Generally speaking, when train-time and test-time data matched, the SAE model had minimal to no performance advantage (in terms of image classification accuracy) over a standard convolutional network of the same computational capacity. However, a slight performance edge emerged when test-time noise shifted from that seen in training. This indicates that our initial intuition was justified: the introduction of reconstructive loss may assist with denoising in the presence of more extreme levels of noise. We found that the baseline convolutional network generally could handle simpler noise (contrary to our initial expectations), but for more challenging zero-out noise, a clear advantage emerged for the SAE.

Aside from the slightly improved robustness to noise, the SAE generally performed equally well as the baseline despite utilizing a larger, more computationally expensive architecture. We often found that the inclusion of reconstructive losses had an adverse or neutral affect on classification performance. These results largely seem to suggest that information relevant to classification and reconstruction do not necessarily overlap.

Empirically, training the SAE model on varying amounts of labeled and unlabeled data, did not result in improvement in the generalization of the model. This was true for both the training procedures we used (pre-training and iterative). The results of the first variation of the SAE model (which separated the learning stages with unsupervised followed by supervised), suggest that information needed to reconstruct an input image does not improve its classification performance. This SAE model, however, as indicated by the improved classification accuracy in our noise robustness tests, is more robust than the baseline convolutional classification network. Furthermore, while our experimentation with iterative SAE training was driven by the potential regularization benefits of this method, our results seem to suggest the opposite — iterative training causes the model to overfit, shows no improvement in robustness, and ultimately demonstrates the inadequacy of an image reconstruction task in regularizing image classification predictions.

Limitations and Future Work: Our experiments and results explored a broad range of environments in which we conjectured the SAE model may provide an advantage over standard convolutional networks. Due to the limited timeline of the project, we were unable to explore any one of these avenues as extensively as we would have liked, and we leave that up to future work. Exploring performance in the face of other types of challenging noise, playing more extensively with the model architecture, and testing on a wider range of datasets (such as transfer performance to CIFAR-100) were all directions that we considered, and could be pursued in the future. Motivated by the SAE performance in the face of test-time noise distribution shifts, we also conjecture that our model could provide some sort of edge for transfer learning tasks. The SAE performs better than the baseline when test-time noise diverges from that seen in train-time – this concept is foundational to transfer learning, and perhaps the inclusion of reconstructive losses could encourage representations that generalize to other datasets and tasks more effectively.

6 Contributions of Group Members

GitHub repository:

`https://github.com/TristanSaidi/Supervised-Denoising-Autoencoder`

Tristan Saidi (tls2160): I had the original idea for this project, and I implemented a significant amount of the code. I did the experimentation with noise, model architecture and the loss weighting. I also implemented dataloading for the labeled and unlabeled splits, and wrote the abstract, most of the method, half of the results section and a majority of the discussion section.

Sagarika Sharma (ss5946): My contribution to the project was in writing the code for the variations of the supervised autoencoder and convolutional network that were tested with varying amounts of labeled/unlabeled data. I conducted most of the experimentation related to this and wrote and created figures for the corresponding two sections, Labeled vs unlabeled data splits and iterative unsupervised and supervised learning, in the results and conclusion section. I also conducted a lot of experimentation with different model architectures, and generally was involved in many of the different attempts we made with our research topic.

Mitali Juneja (mj2944): My contribution to this project was in helping to debug much of our code, in order to make sure our models and training were properly implemented. Specifically, I helped to debug our convolutional network baseline (so that it was performing at reasonable levels), as well as our SAE training loop (so that the proper model inputs were being used, and loss and accuracy computations were being performed correctly). I also implemented some preliminary experiments with adversarial noise, and helped write the introduction and method section of our report.

Nikhilesh Belulkar (nb2953): I assisted with tuning the model architecture to generate a noticeable result: specifically experimenting with a shallower classification network. I also researched relevant papers so that we could gain a background understanding of the paper. I wrote the related works section and contributed to the introduction and the abstract of this paper.

References

[dCCN+16] Gabriel B. Paranhos da Costa, Welinton A. Contato, Tiago S. Nazare, João E. S. Batista Neto, and Moacir Ponti. An empirical study on the effects of different types of noise in image classification tasks, 2016.

[EBC+10] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(19):625–660, 2010.

[EMB+09] Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal Vincent. The difficulty of training deep architectures and the effect of unsupervised pre-training. In David van Dyk and Max Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 153–160, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.

[LBBH98] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LPW18] Lei Le, Andrew Patterson, and Martha White. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[RHAM18] Sudipta Singha Roy, Sk. Imran Hossain, M. A. H. Akhand, and Kazuyuki Murase. A robust system for noisy image classification combining denoising autoencoder and convolutional neural network. *International Journal of Advanced Computer Science and Applications*, 9(1), 2018.

[VLL+10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(110):3371–3408, 2010.

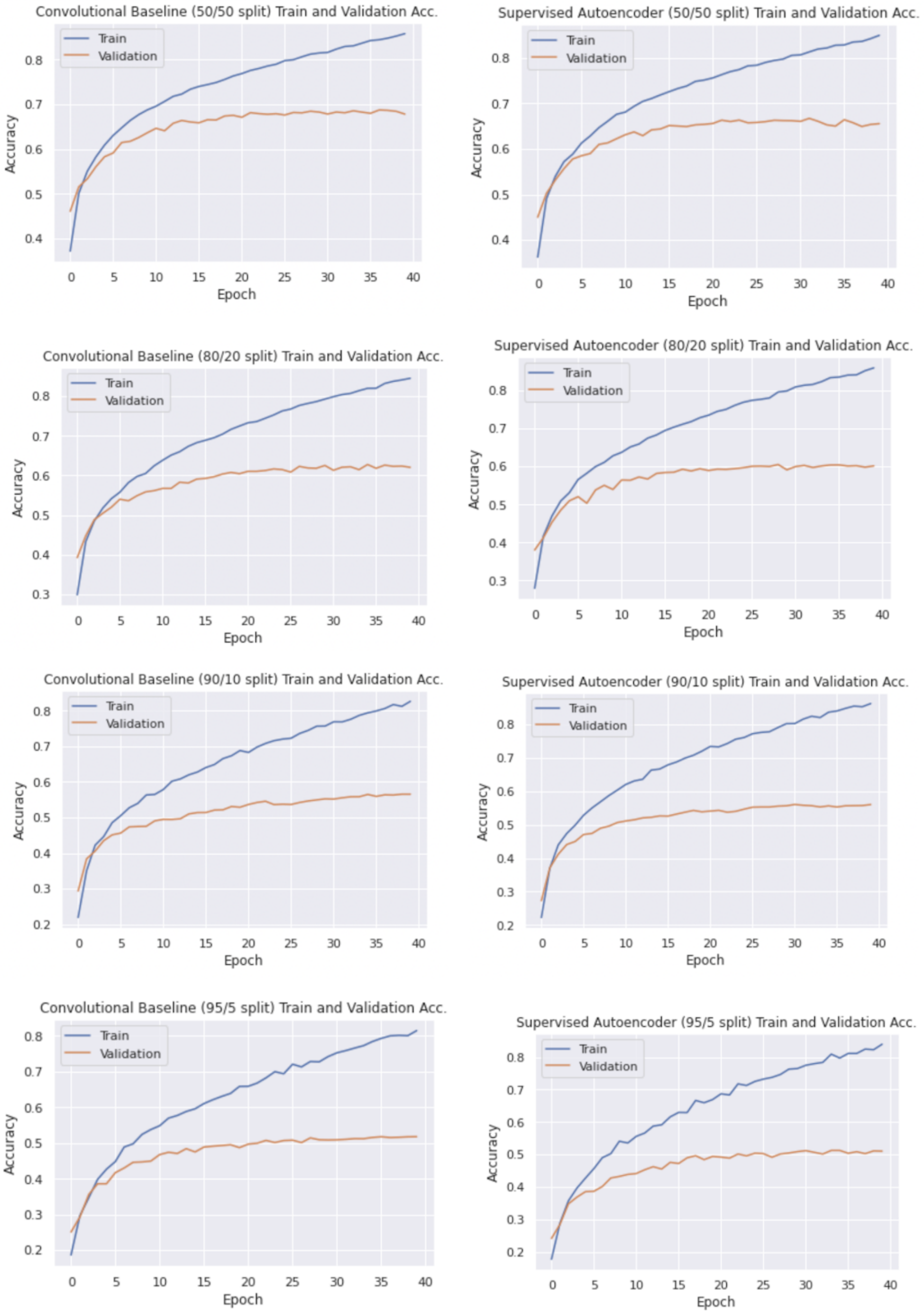

Figure 7: Convolutional baseline network (left) and SAE (right) performance with varying number of labeled data points and unlabeled data points seen during train time

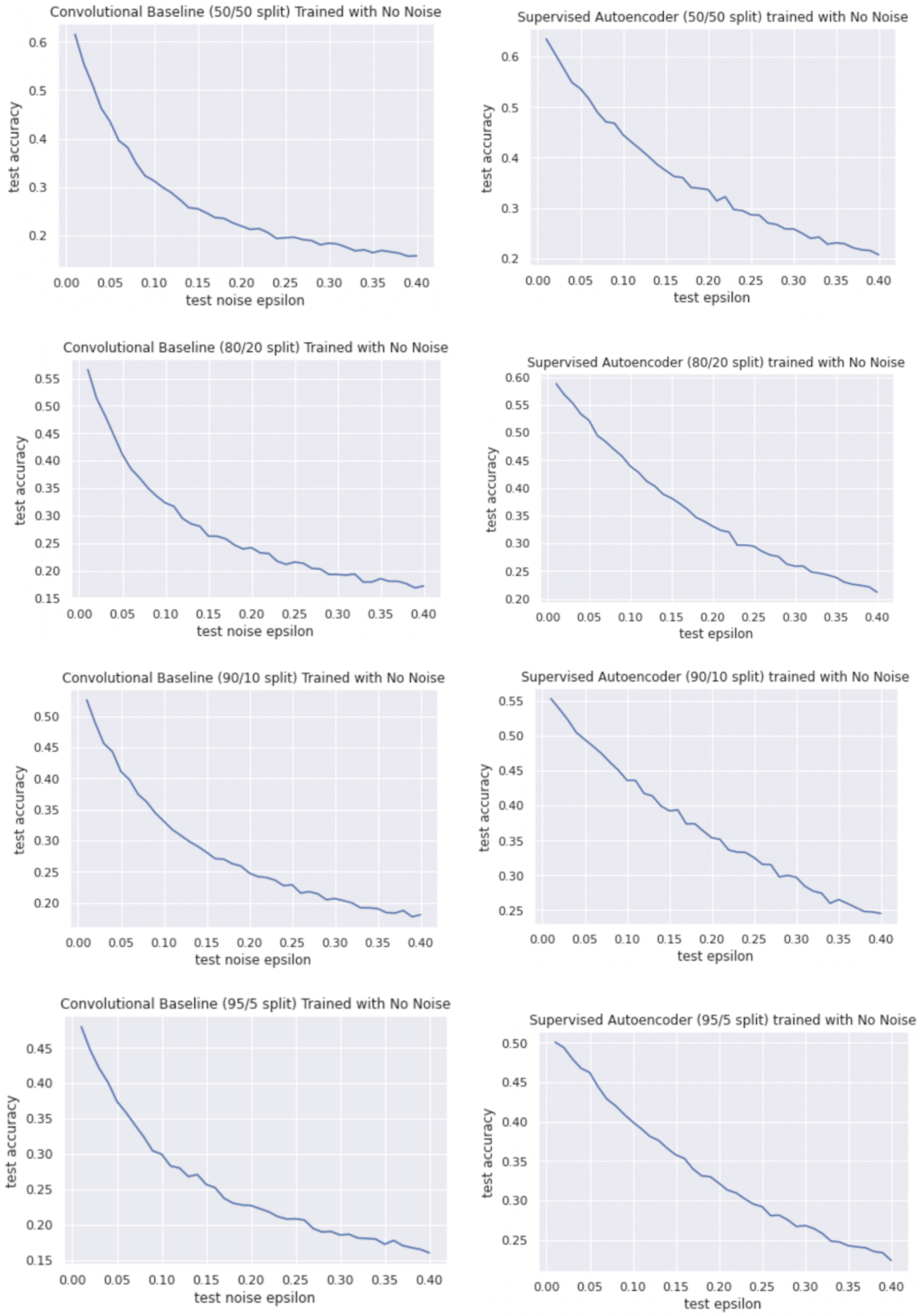

Figure 8: Convolutional baseline network (left) and SAE (right) performance with varying number of labeled data points seen during train time (each row) with varying levels of test time zero-out noise.

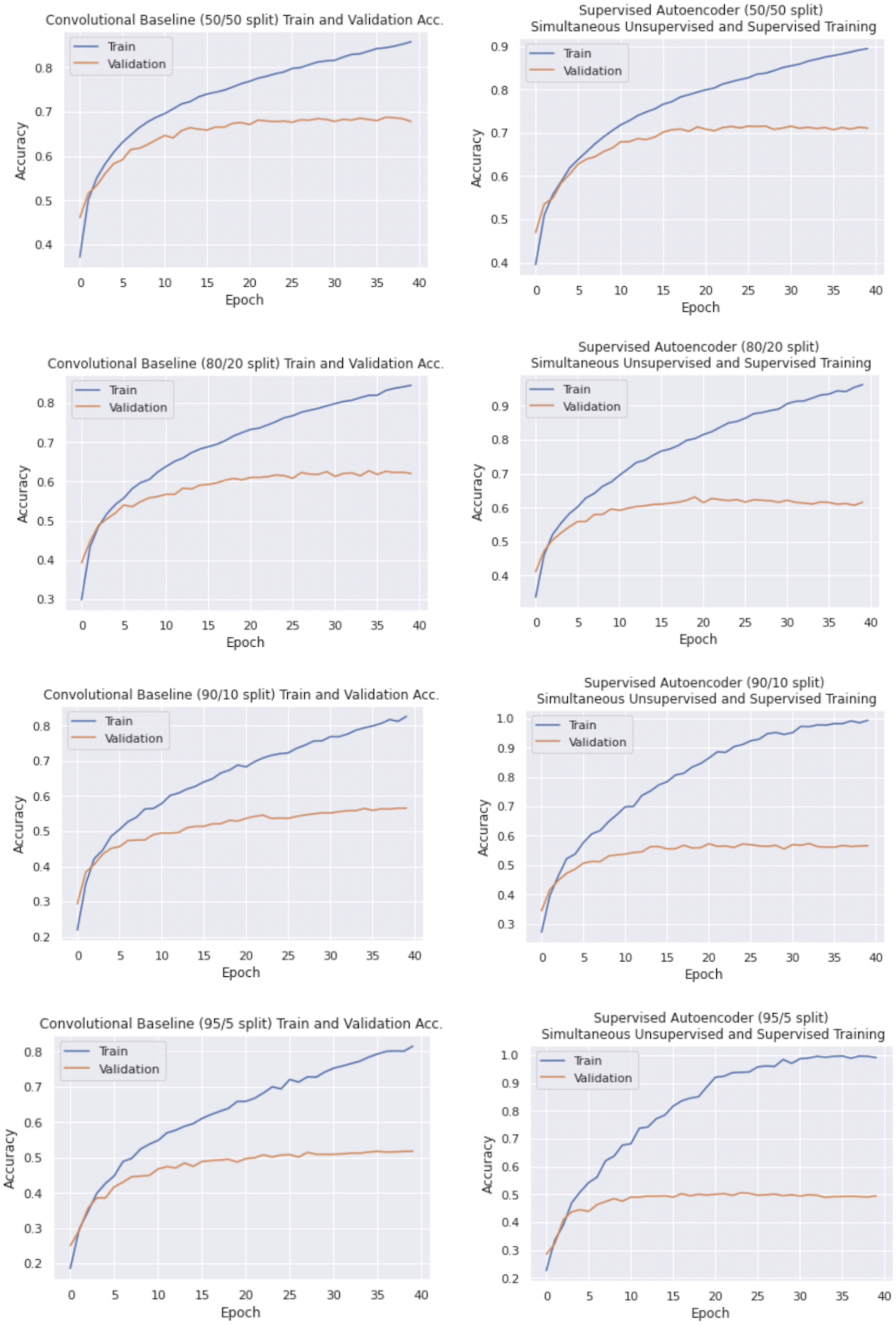

Figure 9: Convolutional baseline network (left) and SAE with simultaneous supervised and unsupervised learning (right) performance with varying number of labeled data points and unlabeled data points seen during train time

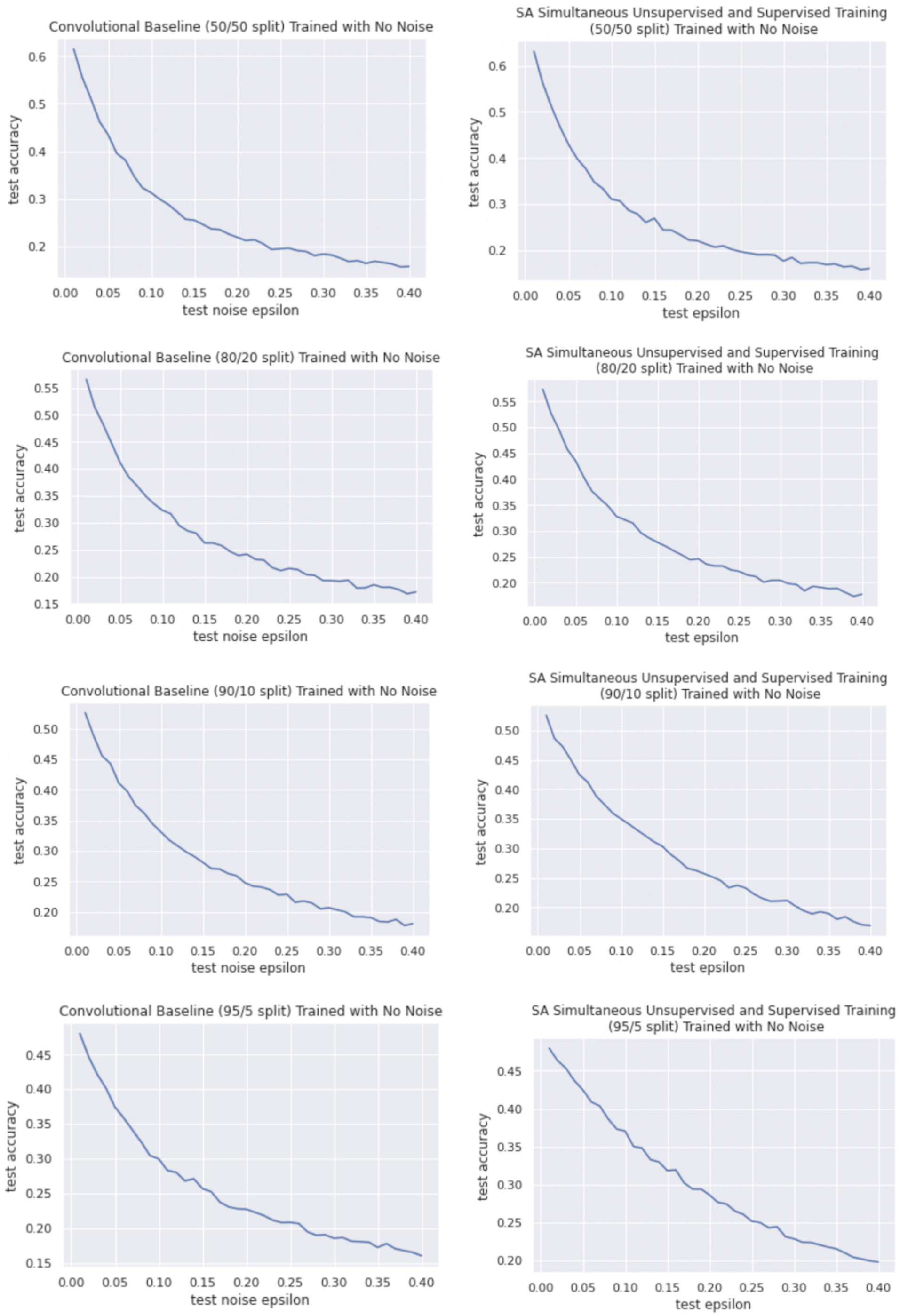

Figure 10: Convolutional baseline network (left) and SAE with simultaneous supervised and unsupervised learning (right) performance with varying number of labeled data points seen during train time (each row) with varying levels of test time zero-out noise.