Leveraging Diffusion Models for EEG-to-Image Conversion

Anugu Arun Reddy, Prajwal Singh, Prof. Shanmuganathan Raman, Prof. Krishna Prasad

Department of Computer Science and Engineering,
Indian Institute of Technology, Gandhinagar

Problem Definition

This research addresses the challenge of converting EEG signals into visually interpretable photorealistic images. Leveraging diffusion models, the aim is to enhance the accuracy and reliability of this conversion process. The ultimate goal is to improve the accessibility and interpretability of brain signals for various applications.

Introduction

The below points summarize the importance and challenges associated with this problem.

- Interpreting Brain Activity
- Challenges of Noisy EEG Signals
- Limited Data Availability
- High Training Time Requirements
- No pretrained encoders for EEG signals
- Multifaceted Complexity

Our Approach

We used latent diffusion models for the image generation, the main problem that was to be solved here was to get proper encoding for the EEG signals. There are no pretrained EEG encoders available that could do this task. However naively training a transformer based encoder along with the diffusion model is not feasible as the gradients to the transformer encoder are really small.

To solve this problem we first converted EEG signals into a space where the signals corresponding to similar images have similar embeddings. Following this training the diffusion model produces really good results.

Model

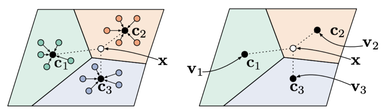


Figure 1. The EEG embeddings are seperated with the help of image supervision.

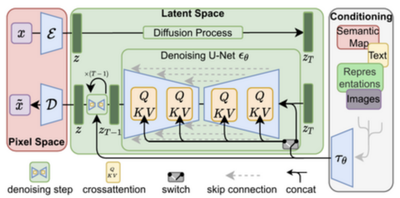


Figure 2. Latent diffusion model is used to generate images conditioned on EEG.

Results

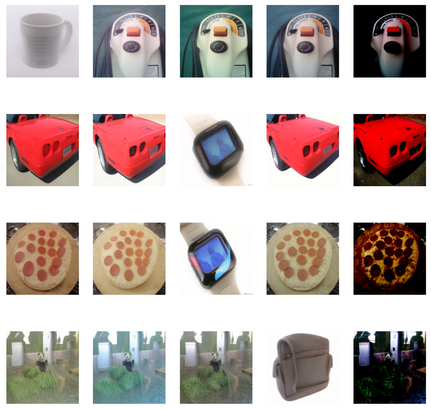


Figure 3. Results obtained on the test set after 50 epochs. The rightmost column consists of ground truth images for EEG signal, remaining columns are the generated images. Results could definitely be improved by further training. One epoch takes about 1 hour. SOTA models are trained for more than 300 epochs on many more GPUs.

(a) Generated Image at epoch 34. (b) Generated Image at epoch 35. (c) Generated Image at epoch 36.

Figure 4. The training exhibits the phenomenon of sudden convergence. Image obtained till epoch number 34 is inconsistent but the model suddenly improves within next 2 epochs.

Class	FID	I.S	LPIPS
Car	3.95 ± 0.07	210.67	0.012
Pizza	2.68 ± 1.02	198.23	0.093
Panda	1.04 ± 2.32	186.45	0.061
Train	2.04 ± 0.32	202.39	0.095
Iron	3.01 ± 0.57	192.78	0.031
Overall	4.01 ± 0.76	195.63	0.091

Table 1. Image Generation Evaluation Metrics.

Alternate Approaches

Comparsion of alternate approaches

Previous approaches include seperately training a transformer based EEG encoder using masked language modelling. This requires lot of EEG data and its also time consuming.

Jointly training EEG encoder along with diffusion model is not feasible as the gradients to the parameters of the EEG encoder model would be really small.

Advantage of our approach

Our approach incorporates the ideas of few shot learning and trains the encoder with very limited data and could also be adapted to other domains really easily. This is efficient both compute wise and time wise.

We trained out model with less than 12,000 image-EEG pairs. Not requiring lot of EEG data is the main advantage of our approach.

Future Work

In our approach we just finetuned all the weights of the model with small learning rate. We can also utilize many recent finetuning techniques to improve the training process.

Low Rank Adaptation.

Low-rank adaptation is a technique used in machine learning to efficiently adapt models to new tasks or datasets by constraining their parameters to low-rank structures. By reducing the dimensionality of the parameter space while preserving important information, low-rank adaptation facilitates faster learning and improved generalization, particularly in scenarios with limited computational resources or labeled data.

Quantization.

Quantization in large models involves reducing the precision of numerical values representing model parameters or activations. This process is used to decrease memory usage and computational requirements, making language models more efficient for deployment on resource-constrained devices. While quantization may introduce some loss of accuracy, optimization techniques can help mitigate this impact, ensuring that the quantized model maintains acceptable performance.

References

[1] Timo Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer.
Qlora: Efficient finetuning of quantized llms.
arXiv preprint arXiv:2305.14314, 5 2023.

[2] Edwin Jian Hu, Yelong Shen, Patrick Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Sida Wang, Liwei Wang, and Wenbo Chen.
Lora: Low-rank adaptation of large language models.
arXiv preprint arXiv:2106.09685, 6 2021.

[3] Robin Rombach, Alexander Blattmann, Dominic Lorenz, Patrick Esser, and Bjorn Ommer.
High-resolution image synthesis with latent diffusion models.
arXiv preprint arXiv:2112.10752, 12 2021.

[4] Jake Snell, Kevin Swersky, and Richard S Zemel.
Prototypical networks for few-shot learning.
arXiv preprint arXiv:1703.05175, 3 2017.

reddyanugu@iitgn.ac.in

Sinusoidally Bounded Neural Networks for Adversarial Robustness.

Anugu Arun Reddy, Zeel B Patel, Prof. Nipun Batra

Department of Computer Science and Engineering,
Indian Institute of Technology, Gandhinagar

Problem Definition

Neural networks are known to be susceptible to adversarial examples, wherein inputs closely resemble natural data but are incorrectly classified by the network.

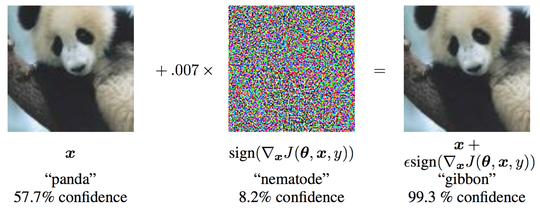


Figure 1. Small amount of noise added to the input can fool the network into making wrong predictions.

The problem of adversarial examples is crucial to address because it exposes vulnerabilities in machine learning models, particularly neural networks, which are increasingly integrated into critical systems like autonomous vehicles, healthcare diagnostics, and cybersecurity. Solving this problem is essential to ensure the trustworthiness and robustness of AI systems deployed in real-world scenarios.

Our approach

We propose that adversarial examples exist because of the unbounded nature of logits of the neural network. Hence we mitigate this by using a bounded activation function at the end of the neural network to bound the logits.

Our contribution is that using sine activation at the end of the neural network along with a special weight initialization scheme improves adversarial robustness.

The weight initialization scheme is important otherwise the gradients to the earliest layers of the neural network would be really small and the network would not learn anything at all.

The distribution shift in case of using sine is lesser than while using relu.

Special weight initialization scheme is as follows.

$$W \sim \mathcal{U}\left(-\frac{p}{6 \cdot \text{fan\_in}}, \frac{p}{6 \cdot \text{fan\_in}}\right)$$

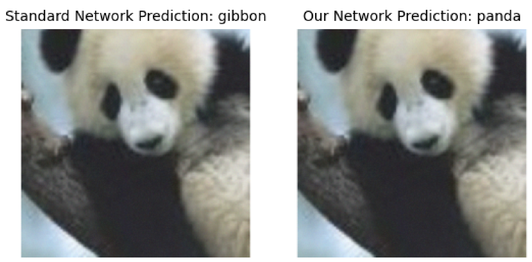
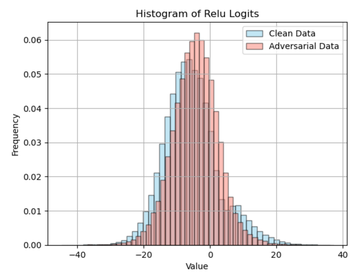
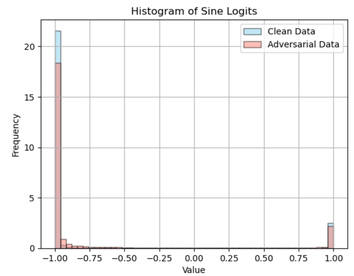


Figure 2. Our Network is robust to adversarial attacks.

Results

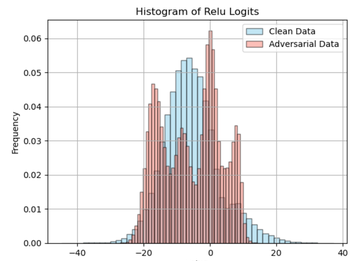


(a) Distribution of relu logits. The mode of the distribution has shifted causing neural network to make wrong predictions.

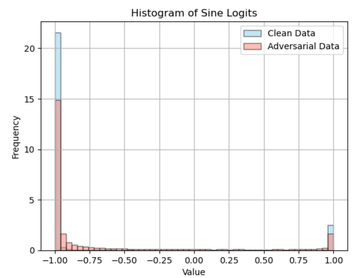


(b) Distribution of sine logits. The mode of the distribution has not shited. Hence neural network still makes good predictions.

Figure 3. Comparison of logit distribution of standard and our method.



(a) Distribution of relu logits. The mode of the distribution has deviated significantly causing neural network to make wrong predictions.



(b) Distribution of sine logits. The mode of the distribution has not shited. Hence neural network still makes good predictions.

Figure 4. Comparison of logit distribution of standard and our method on a much stronger attack.

Attack	Hyperparameter	Standard CNN	Ours
FGSM	$\epsilon = 0.1$	46.8	**48.25**
FGSM	$\epsilon = 0.2$	31.85	**35.88**
FGSM	$\epsilon = 0.3$	21.8	**28.57**
PGD	$\epsilon = 0.3$	25.82	**28.54**
PGD	$\epsilon = 0.5$	12.5	**20.1**
PGD	$\epsilon = 0.7$	5.82	**16.6**
A.A	$\epsilon = 0.2$	34.36	**34.92**
A.A	$\epsilon = 0.3$	25.51	**27.92**
A.A	$\epsilon = 0.4$	18.76	**23.73**
A.A	$\epsilon = 0.5$	14.26	**21.46**

(a) **CIFAR-10** dataset. Classification accuracies for Conventional CNN and Our method. Both the models are **adversarially trained**. Our method works significantly better in most cases. Even in the case when conventional model is better the gap between both is low. A.A is most widely accepted attack for benchmarking.

Attack	Hyperparameter	Standard CNN	Ours
FGSM	$\epsilon = 0.1$	91.73	**92.54**
FGSM	$\epsilon = 0.2$	65.75	**80.3**
FGSM	$\epsilon = 0.3$	29.93	**64.81**
PGD	$\epsilon = 0.3$	0.16	**13.77**
PGD	$\epsilon = 0.5$	0.0	**5.60**
PGD	$\epsilon = 0.7$	0.0	**3.32**
A.A	$\epsilon = 0.2$	30.59	**49.39**
A.A	$\epsilon = 0.3$	0.47	**10.44**
A.A	$\epsilon = 0.4$	0.2	**6.55**
A.A	$\epsilon = 0.5$	0.21	**5.38**

(b) **MNIST** dataset. Classification accuracies for Conventional CNN and Our method. Both the models are **adversarially trained**. Our method works significantly better in most cases. Even in the case when conventional model is better the gap between both is low. A.A is most widely accepted attack for benchmarking.

Alternate Approach

Adversarial Training.

- **Approach:** Adversarial training involves augmenting the training process by including adversarial examples during training. This helps the model learn to be robust against such examples by exposing it to a variety of perturbations.
- Cons:
  - **Computational Cost:** Adversarial training requires generating adversarial examples for each training iteration, which significantly increases computational cost.
  - **Overfitting:** There's a risk of the model overfitting to the specific adversarial examples used during training, potentially reducing its generalization performance on unseen data.

Robust Optimization

- **Approach:** Robust optimization techniques aim to directly optimize the model parameters to increase robustness against adversarial attacks. This may involve incorporating regularization terms that penalize sensitivity to small perturbations.
- Cons:
  - **Complexity:** Implementing robust optimization techniques often involves complex mathematical formulations, making them difficult to understand and implement.
  - **Trade-offs:** There's a trade-off between robustness and standard performance metrics (e.g., accuracy). Enhancing robustness may come at the cost of reduced performance on clean data, limiting the model's overall effectiveness.

Future Work.

Theoretical Foundation

We need to develop a rigorous mathematical foundation to support these empirical results. Investigate into the theoretical properties of activations that affect adversarial robustness.

Transferability of attacks

Investigate the transferability of adversarial attacks across different models, architectures, and domains. Understanding how adversarial examples generalize across different settings can provide insights into the fundamental properties of neural networks and inform the design of more robust models.

References

[1] Francesco Croce and Matthias Hein.
Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks.
arXiv preprint arXiv:2003.01690, March 2020.

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy.
Explaining and harnessing adversarial examples.
arXiv preprint arXiv:1412.6572, December 2014.

[3] Szilveszter Kotyan.
A reading survey on adversarial machine learning: Adversarial attacks and their understanding.
arXiv preprint arXiv:2308.03363, August 2023.

[4] Yen-Chen Liu, Xinyang Chen, Chang Liu, and Dawn Song.
Delving into transferable adversarial examples and black-box attacks.
arXiv preprint arXiv:1611.02770, November 2016.

[5] Vincent Sitzmann, Julius N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein.
Implicit neural representations with periodic activation functions.
arXiv preprint arXiv:2006.09661, 2020.

[6] Hu Xu, Yisen Ma, Hong Liu, Diptangshu Deb, Hong Liu, Jiliang Tang, and Anil K Jain.
Adversarial attacks and defenses in images, graphs and text: a review.
arXiv preprint arXiv:1909.08072, September 2019.

reddyanugu@iitgn.ac.in