Cross-modal Adversarial Reprogramming - Supplementary Material

1. Wall-clock inference time of reprogramming function

In Table 1, we report the wall clock inference time for the adversarial program and the benchmark text classifiers studied in our work for a sequence of length 500. Both the Bi-LSTM and CNN model use 256 hidden units and an embedding size of 256. We use a single layer Bi-LSTM network and 1-D CNN with convolution filters of size 3, 4 and 5 based on the architecture proposed in (Kim, 2014). For the adversarial program the patch size is $16X16$ and the output image size is $384X384$. We average the inference time for 100 sequences for these evaluations. It can be seen that the adversarial program is significantly faster than both Bi-LSTM and 1D-CNN models for both CPU and GPU implementations in PyTorch. The CPU used for these evaluations is Intel Xeon CPU and the GPU is an Nvidia Titan 1080i.

Model	CPU	GPU
Adversarial Program	7.9 ms	0.2 ms
Bi-LSTM	161.5 ms	13.9 ms
1D CNN	383.2 ms	2.2 ms

Table 1. Wall clock inference time (in miliseconds) for the adversarial program and the benchmark text classifiers studied in our work for a sequence of length 500.

2. Hyper-parameter details of benchmark classifiers

For training the benchmark neural-text classifiers, we use the Bi-LSTM and 1D-CNN model with a softmax classification head. For both of these models, the token embedding layer is randomly initialized and trained with the model parameters. We use Adam optimizer and perform mini-batch gradient descent using a batch size of 32 for both of these models for a maximum 200k mini-batch iterations. For the 1D-CNN models we use filters of size 3, 4 and 5 for the three convolutional layers. Other hyper-parameter details of these models are listed in Table 2.

Model	Hidden Units	Emb. Size	# Layers	LR
Bi-LSTM	256	256	1	1e-4
1D CNN	256	256	3	1e-4

Table 2. Hyper-parameter details for the neural sequence classifiers used as benchmark classifiers in our work. LR: Learning Rate. Emb. Size: Embedding Size.

3. Perturbation amount vs Accuracy in Bounded attacks

Figure 1 shows the performance of cross-modal adversarial reprogramming at different magnitudes of allowed perturbation in the bounded attack setting, while attacking the ViT model. We use the same base image x_c as used in all of our bounded attack experiments.

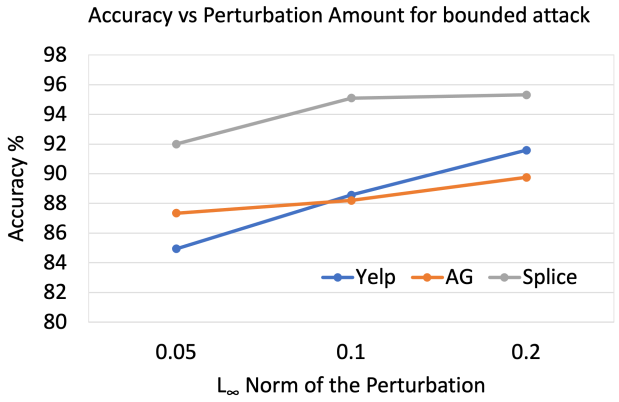

Figure 1. Accuracy vs L_∞ norm of the perturbation while reprogramming the ViT model for three target tasks covering emotion, topic and DNA sequence classification.

4. Assessing the importance of Victim Model

To assess the importance of the victim model for solving the target task, we perform an experiment to understand the extent to which the target task can be solved by using only the adversarial reprogramming function with a linear classification head on top. To perform this experiment, we take the mean of all token embeddings in a sequence and pass it as input to a linear layer that predicts the class scores for the target task. Not surprisingly, this classifier works well for

Dataset	Patch Size	Accuracy%
Yelp	16	89.86
IMDB	16	87.75
AG	16	91.39
DBPedia	32	97.01
Splice	48	50.41
H3	16	75.12

Table 3. Performance of a classifier that uses only an embedding layer and a classification head on the datasets used in our study.

sentiment and topic classification tasks which can be solved reliably using word-frequency based methods (as reported in Table 2 of the main paper). However, compared to our reprogramming methods (reported in Table 2 of the main paper), the classifier significantly underperforms on the two DNA sequence classification tasks that require understanding the underlying semantics of the sequence. This suggests that while the embedding layer of the adversarial program can sufficiently capture word-frequency statistics and independently solve tasks that require keyword detection, the pre-trained victim model is essential for tasks that require analysing the structure and semantics of the sequence.

References

Kim, Y. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.