Backdoor Training Parameters

Environment Settings
- Framework version (PyTorch version): 2.7.0
- CUDA version (if applicable):
- GPU used (Newt/Floo/local - specifics): Newt

Dataset Settings
- Data preprocessing techniques applied (if any):
- Data augmentation techniques (if any):

Model Architecture
- Base model: VGG16 pretrained on GTSRB
- Any modifications to the architecture: none

Backdoor Implementation
- Trigger pattern description in words (must be ≤16 pixels): 4x4 square on the very left, (vertical-wise) center of the image. The square is a split vertically between green and blue.
- Poisoning ratio (% of training data poisoned): ~50% (used random select on bool to decide whether to poison original data or not.

Training Hyperparameters
- Number of epochs: 20
- Batch size: 32
- Optimizer: Adam
- Learning rate: 1e-4
- Learning rate schedule (if any): no
- Weight decay: 0
- Loss function: CrossEntropyLoss()
- Early stopping criteria (if used): None

Random Seeds
- Random seed for model initialization (if applicable):
- Random seed for weight initialization (if applicable):
- Random seed for data augmentation (if applicable):

Additional Notes: please write any other techniques and methods used.

For backdoor training and validation set creation, I just iterated through the original train and test sets, and replaced 1/2 of source_class images with a triggered image and set its respective label to target_class.