

Deliverable

Process for getting PrimeVul helper script running with the PrimeVul Dataset

The following steps were conducted under similar conditions:

1. Windows 10 - latest update
2. WSL 2 - Running Ubuntu 22.04
3. Nvidia GPU

One of the systems is at the following versions:

```
WSL version: 2.4.11.0
Kernel version: 5.15.167.4-1
WSLg version: 1.0.65
MSRDC version: 1.2.5716
Direct3D version: 1.611.1-81528511
DXCore version: 10.0.26100.1-240331-1435.ge-release
Windows version: 10.0.19045.5487
```

With the overarching idea of the installation being put forth with the dependency start in given in the repo. Deploying the Conda environment in WSL with manually installing the specific DGL dependency and CUDA for pytorch.

1. Start with downloading and initializing the Conda environment. Instructions can be found here : [HERE](#). But summarized, get the miniforge conda version found here [HERE](#) for intel x86_64 and download it in WSL.
2. The script handles most of it but it may be recommended in leaving the base conda env on for the duration of the project to be able to quickly activate the primevul env. Or utilize:

```
eval "$($(~/miniforge3/bin/conda shell.bash hook)"
```

3. Now let's pull the github repo, cd into the repo, and run:

```
# WE WANT TO BE HERE FOR EVERYTHING ELSE AND IN THIS ENV AT ALL TIMES
conda env create -f environment.yml
```

```
# THIS WILL FAIL IT IS OK. IT WILL FAIL AT DGL
```

4. THIS WILL FAIL. It will hang up and abandon given the dgl package but will create the environment. So we will have to manually install it and then pip install -r the rest of the requirements. Make sure you are in the "primevul" conda env. If you are not in the conda env, run:

```
# make sure you're in the right env
conda activate primevul
```

```
# after this, manually install the package with pip
pip install "dgl==1.0.1" -f https://data.dgl.ai/wheels/cu113/repo.html
```

5. Let's download CUDA for WSL following this link: [HERE](#) Make sure its Linux, x86, WSL-Ubuntu, CUDA 12.8. Commands for reference:

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-wsl-ubuntu.pin
$ sudo mv cuda-wsl-ubuntu.pin /etc/apt/preferences.d/cuda-repository-pin-600
$ wget https://developer.download.nvidia.com/compute/cuda/12.8.0/local_installers/cuda-repo-wsl-ubuntu-12-8-local_12.8.0-1_amd64.deb
$ sudo dpkg -i cuda-repo-wsl-ubuntu-12-8-local_12.8.0-1_amd64.deb
$ sudo cp /var/cuda-repo-wsl-ubuntu-12-8-local/cuda-*-keyring.gpg /usr/share/keyrings/
$ sudo apt-get updatesudo apt-get -y install cuda-toolkit-12-8
```

6. Now we can install the rest of the requirements. Take the stripped requirements page to utilize in pip install -r requirements or strip them from the environment.yml with:

```
# Parse to requirements file
grep -A 1000 " - pip:" <PATH TO ENV.YAML> | tail -n +2 | sed 's/ - //g' > <PATH TO OUTPUT FILE>
```

```
# Install remaining pip packages with requirements
pip install -r requirements.txt
```

7. Now you should be able to run the helper bash script to run the desired model and settings. The helper.sh should be run in os_expr!
8. You shouldn't reach this but if any other errors appear they may be resolved.