Stochastic Deep Embedded Clustering (S-DEC)

Baseline: Deterministic DEC (for comparison only S-DEC itself is non-deterministic)
Required Comparison: Bayesian Deep Clustering Network (VaDE)

1) Prerequisites:

1. macOS / Linux / Windows
2. Python 3.9+
3. VS Code (recommended)
4. Git (optional)

2) Setup (First time):

Open a terminal (or PowerShell on Windows) in the project folder and run:

macOS / Linux:

```
python3 -m venv sdec_env
source sdec_env/bin/activate
pip install --upgrade pip   # optional
pip install torch torchvision scikit-learn matplotlib
```

Windows (PowerShell):

```
python -m venv sdec_env
.\sdec_env\Scripts\activate
python -m pip install --upgrade pip   # optional
pip install torch torchvision scikit-learn matplotlib
```

If VS Code asks for an interpreter, select:

- macOS/Linux: <project>/sdec_env/bin/python3
- Windows: <project>\sdec_env\Scripts\python.exe

3) Files:

File/Folder	Purpose
sdec_train.py	Trains **Baseline DEC** (deterministic) & **S-DEC** (stochastic) on MNIST
plot_results.py	Creates a bar chart for the last run
merge_results.py	Aggregates results from multiple seeds (mean ± std) & plots with error bars
results	Contains outputs (metrics.txt, metrics_bar.png, metrics_seed_*.txt, metrics_mean_std.csv, metrics_bar_err.png)
data	Auto-downloaded MNIST data

4) Single Seed (Quick Run):

macOS / Linux:

```
source sdec_env/bin/activate
python3 sdec_train.py
```

Windows:

```
.\sdec_env\Scripts\activate
python sdec_train.py
```

Outputs:

results/metrics.txt (ARI, NMI, Silhouette for Baseline + S-DEC)

Plot results:

```
python3 plot_results.py   # or `python` on Windows
```

Outputs:

results/metrics_bar.png

Optional faster/stronger runs:

```
python3 sdec_train.py --epochs_pre 3 --epochs_dec 6
python3 sdec_train.py --epochs_pre 10 --epochs_dec 20 --batch 256 --latent 10 --k 10
```

5) Stability Multiple Seeds (recomanded):

macOS / Linux:

```
source sdec_env/bin/activate
for s in 1 2 3 4 5; do
  python3 sdec_train.py --seed $s --epochs_pre 5 --epochs_dec 10
  cp results/metrics.txt results/metrics_seed_${s}.txt
done
python3 merge_results.py
```

Windows (PowerShell):

```
.\sdec_env\Scripts\activate
For ($s=1; $s -le 5; $s++) {
  python sdec_train.py --seed $s --epochs_pre 5 --epochs_dec 10
  Copy-Item results\metrics.txt results\metrics_seed_$s.txt
}
python merge_results.py
```

6) Code workflows (High Level):

1. Loads MNIST (via torchvision) and merges train+test for unsupervised clustering
2. Pretrains Autoencoder (few epochs) → latent features
3. Initializes KMeans cluster centers in latent space
4. DEC fine-tuning:

Baseline: deterministic encoder; optimizes KL(p || q)

S-DEC: stochastic latent $z = \mu + \sigma \odot \varepsilon$ with KL penalty on (μ, σ)

5. Reports clustering metrics: ARI, NMI, Silhouette
6. For stability: repeat with different seeds, aggregate mean ± std
7. Required Comparison: mention **VaDE** (Bayesian Deep Clustering Network) and compare conceptually with S-DEC

7) Common Commands:

Task	macOS / Linux	Windows
Activate env	source sdec_env/bin/activate	.\sdec_env\Scripts\activate
Run once	python3 sdec_train.py	python sdec_train.py
Plot results	python3 plot_results.py	python plot_results.py
Run 5 seeds + aggregate	(loop script above)	(PowerShell loop above)
Clean results	rm -rf results/*	Remove-Item results* -Recurse -Force

8) Troubleshooting:

- 'python: command not found' → use python3 (mac/Linux) or python (Windows)
- 'torch not found' → pip install torch torchvision (after activating env)
- MNIST download blocked → place IDX files under ./data/raw and modify loader
- Silhouette = NaN → try more epochs, different seed, or different β
- Slow training → check device printed at start (mps/cuda/cpu). MPS/CUDA are fastest.