Arabic Sign Language Experiments - Quick Start Checklist

Pre-Experiment Checklist

✅ Environment Setup

- [ ] Python 3.8+ installed
- [ ] CUDA-enabled GPU available (recommended)
- [ ] At least 8GB RAM
- [ ] At least 5GB free disk space

✅ Dependencies Installed

```bash
pip install torch torchvision opencv-python mediapipe numpy pyyaml matplotlib tqdm
```

- [ ] PyTorch installed with CUDA support
- [ ] All other dependencies installed
- [ ] Verified: `python -c "import torch; print(torch.cuda.is_available())"` → `True`

✅ Project Files

- [ ] `configs/arabic-asl.yaml` created
- [ ] `prepare_arabic_asl.py` updated (with label file copying fix)
- [ ] `utils.py` updated (with device handling fix)
- [ ] `main.py` updated (with timing and metrics)
- [ ] `train_loso.py` created
- [ ] All original files (model.py, dataset.py, etc.) present

✅ Dataset

- [ ] Videos located at: `data/MLR511-ArabicSignLanguage-Dataset-MP4/`
- [ ] Verified: 12 users (user01-user12)
- [ ] Verified: 10 gestures per user (G01-G10)
- [ ] Verified: 10 repetitions per gesture (R01-R10)
- [ ] Total expected videos: 1,200

Data Preparation Checklist

Step 1: Run Data Preparation

```bash
```

```
python prepare_arabic_asl.py
```

✅ Verify Output

- [ ] Processing completed without errors
- [ ] Message: "Processing complete: 1200 successful, 0 failed"
- [ ] `data/arabic-asl/all/` directory created
- [ ] `data/arabic-asl/label2id.json` exists
- [ ] `data/arabic-asl/id2label.json` exists
- [ ] `data/arabic-asl/meta.jsonl` exists

✅ Verify Sample Count

```bash
find data/arabic-asl/all -name "*.pkl" | wc -l
```

- [ ] Result: 1200 files

✅ Verify LOSO Splits

```bash
# Check directories exist
ls data/arabic-asl_LOSO_user01/
ls data/arabic-asl_LOSO_user08/
ls data/arabic-asl_LOSO_user11/
```

- [ ] All 3 LOSO directories created
- [ ] Each has `train/` and `test/` subdirectories
- [ ] Each has `label2id.json` and `id2label.json`

```bash
# Check sample counts
find data/arabic-asl_LOSO_user01/train -name "*.pkl" | wc -l
find data/arabic-asl_LOSO_user01/test -name "*.pkl" | wc -l
```

- [ ] Train: ~1,100 samples
- [ ] Test: ~100 samples
- [ ] Same for user08 and user11

Training Checklist

Option A: Train All Models (Recommended)

```bash
python train_loso.py --epochs 80 --lr 2e-4
```

Option B: Train Single Model (Quick Test)

```bash
python train_loso.py --holdout_only user01 --epochs 80 --lr 2e-4
```

✅ Training Started Successfully

- [ ] No errors during model initialization
- [ ] Message: "Experiment: arabic_asl_LOSO_user01"
- [ ] Message: "Device: cuda"
- [ ] Model parameters displayed (~736,010 parameters)
- [ ] Configuration displayed
- [ ] Progress bar appeared: "Training epoch 1/80"

✅ Monitor During Training

Every 10 minutes, check:

```bash
# Latest accuracy
tail -5 arabic_asl_LOSO_user01.log
```

- [ ] Training accuracy increasing
- [ ] Validation accuracy increasing
- [ ] No NaN losses

Check GPU usage:

```bash
nvidia-smi
```

- [ ] GPU utilization > 80%
- [ ] GPU memory used

Post-Training Checklist

✅ Verify Training Completed

- [ ] Message: "Training Complete!"
- [ ] Message: "Total training time: X.XX hours"
- [ ] Message: "Best train accuracy: X.XXXX"
- [ ] Message: "Best validation accuracy: X.XXXX"
- [ ] Message: "Plots saved to out-imgs/"

✅ Check Generated Files

Log Files:

```bash
ls -lh arabic_asl_LOSO_*.log
```

- [ ] `arabic_asl_LOSO_user01.log` exists
- [ ] `arabic_asl_LOSO_user08.log` exists (if trained)
- [ ] `arabic_asl_LOSO_user11.log` exists (if trained)

Checkpoints:

```bash
ls -lh checkpoints_arabic_asl_LOSO_user01/
```

- [ ] Checkpoint directory exists
- [ ] Multiple `.pth` files present
- [ ] Best model checkpoints saved

Plots:

```bash
ls -lh out-imgs/arabic_asl_LOSO_*
```

- [ ] Loss/accuracy plots exist (`.png`)
- [ ] Learning rate plots exist (`.png`)

✅ Verify Results Quality

Check final accuracies:

```bash
```

```
grep "Best validation" arabic_asl_LOSO_*.log
```

- [ ] LOSO user01: validation accuracy > 70%
- [ ] LOSO user08: validation accuracy > 70%
- [ ] LOSO user11: validation accuracy > 70%

Check training times:

```bash
grep "Total training time" arabic_asl_LOSO_*.log
```

- [ ] Each model: 2-4 hours on GPU (or 20-40 hours on CPU)

Results Collection Checklist

✅ Extract Key Metrics

Final Accuracies:

```bash
echo "=== Final Validation Accuracies ==="
for user in user01 user08 user11; do
    echo "LOSO $user:"
    grep "VAL" arabic_asl_LOSO_${user}.log | tail -1
done
```

- [ ] All accuracies recorded

Training Times:

```bash
echo "=== Training Times ==="
grep "Total training time" arabic_asl_LOSO_*.log
```

- [ ] All times recorded

Model Size:

```bash
echo "=== Model Parameters ==="
grep "Model Parameters" arabic_asl_LOSO_user01.log | head -1
```

- [ ] Parameter count: ~736,010
- [ ] Model size: ~2.81 MB

✅ Generate Comprehensive Report

```bash
python collect_results.py
```

- [ ] Report generated successfully
- [ ] All sections complete:
- [ ] Network architecture
- [ ] Training configuration
- [ ] Model size and parameters
- [ ] Recognition accuracy per test signer
- [ ] Training and inference times
- [ ] Summary table

Paper Requirements Checklist

✅ 1. Network Architecture

- [ ] Architecture diagram/description documented
- [ ] Layer specifications recorded
- [ ] Input/output dimensions specified
- [ ] Attention mechanism described
- [ ] Citation to original SignBart provided

✅ 2. Configuration and Hyperparameters

- [ ] All hyperparameters documented:
- [ ] Model dimensions (d_model, layers, heads)
- [ ] Learning rate: 2e-4
- [ ] Batch size: 1
- [ ] Epochs: 80
- [ ] Optimizer: AdamW
- [ ] Scheduler: ReduceLROnPlateau
- [ ] Dropout rates
- [ ] Augmentation settings

✅ 3. Model Size and Parameters

- [ ] Total parameters: ~736,010

- [ ] Trainable parameters: ~736,010
- [ ] Model size (MB): ~2.81 MB (FP32)

✅ 4. Recognition Accuracy per Test Signer

- [ ] LOSO user01: Top-1 accuracy: _____% (Top-5: _____%)
- [ ] LOSO user08: Top-1 accuracy: _____% (Top-5: _____%)
- [ ] LOSO user11: Top-1 accuracy: _____% (Top-5: _____%)
- [ ] Average