Work currently being performed

- **Scope/labels:** A–Z letters + approximately 20 MS-ASL words; establishing consistent label names and metadata schema.
- **Data curation:** MS-ASL clips trimmed/validated; preparing **Kaggle ASL Alphabet** for letters; building a single CSV/JSON manifest (paths, session, FPS, frames, label).
- **Personal recordings:** Recording **5 short sessions** under varied lighting/backgrounds (~20 takes per class); organizing **session-wise splits** (Train=S1–S3, Val=S4, Test=S5).
- **Pose pipeline setup:** Implementing **MediaPipe Holistic** for 2D landmarks → integrating a lightweight **2D→3D lifting** baseline; caching landmarks/3D joints to disk for fast iteration.
- **Quality checks:** Quick visualizations of landmark tracks, per-class counts, and spot-checks for label/file mismatches.
- **Tooling:** Set up repo structure, conda/env file, config YAMLs, reproducible seeds, and a simple viewer notebook for sequence sanity checks.

Work planned for the next couple weeks (before 10/30/2025)

- Finalize class list & naming; complete manifest v1 (MS-ASL + Kaggle); finish S1–S5 recordings and code the session split; generate first dataset stats (counts/durations).
- Finish 2D→3D integration + full feature cache; add smoothing (EMA/Savitzky–Golay) and light augmentations; build a minimal windowed dataloader.
- Run smoke tests end-to-end (video → landmarks → 3D → windowed tensors).
- Produce a short report: sample visualizations, example sequences, latency of each preprocessing step, and data stats.
- Implement a tiny **r**eal-time skeleton preview (webcam + top-1 stub) to validate I/O and FPS targets.

Changes / issues

- **Scope adjustment for a one-person project:** Limiting words to **~20** (plus A–Z) and using **session-wise splits** on personal recordings to emulate cross-condition testing without multi-signer collection.
- **Dataset alignment:** Create label map (MS-ASL ↔ Kaggle) and trim noisy MS-ASL clips.
- **Imbalance:** Some classes are sparse (some words have far fewer examples) → plan for sampling and light augmentations.
- **Performance:** Real-time constraints on CPU-only machines → mitigating with cached features, compact models, batch=1 streaming windows.