Robust, Explainable Cross-View 3D Pose Tracking Without Neural Networks

CS 585 Final Report (Milestone 4)

Sean Tomany Jonah Rothman Jigar Kanakhara Bhavya Bavishi Harsha Basavaraj Beth

Department of Computer Science, Boston University

{stomany, jonahr, jigar, bhavyabb, hbbeth}@bu.edu

Abstract

Cross-view 3D human pose tracking pipelines built on linear triangulation and uniform cross-view affinities are brittle under realistic input degradation: a single bad camera, a low confidence detection, or a frame-level desync can collapse the 3D solve. We extend the Chen et al. [2] cross-view tracker with three classical, neural-network-free upgrades: IRLS triangulation with Huber and Tukey losses and a confidence prior, an uncertainty-aware Mahalanobis cross-view affinity, and a per-joint constant-velocity Kalman filter whose measurement covariance is driven by IRLS residuals. We evaluate every combination on the Campus and Shelf benchmarks across four families of input perturbation, three random seeds each. The full pipeline cuts MPJPE under Campus occlusion 20% from 219.2 mm to 19.7 mm ($11\times$), flips MOTA from -0.76 to $+0.87$, and resolves a latent failure of the baseline time-delay perturbation (every method scored 0% PCP) by fixing three compounding bugs in the perturbation and slot-writing code (full pipeline now scores 98.5% PCP). On Shelf, only the two configurations that include uncertainty-aware affinity initialise tracked identities under occlusion. The others simply do not track. Code, data, and per-seed runs are available at `https://github.com/seantomany/585Project/tree/jigar-endterm`.

1 Introduction

Useful exercise feedback depends on knowing whether joint angles, depth, and symmetry stay within safe ranges throughout a movement. A single RGB camera estimates 2D joint positions reliably but cannot resolve depth or out-of-plane rotation. Multi-view 3D pose estimation addresses this by triangulating 2D detections from calibrated cameras into 3D joint positions that support real angle measurements.

Practical multi-camera setups violate the assumptions of textbook triangulation in three ways. Cameras drift out of synchronisation, so corresponding 2D joints come from slightly different moments. 2D detectors hallucinate or drop joints under occlusion and motion blur. And per-joint detection confidence varies by an order of magnitude across cameras and time, but the standard pipeline ignores it. Chen et al. [2] identify the synchronisation problem and propose incremental triangulation. We focus on making the underlying solve robust to all three failures simultaneously, while keeping every decision auditable per joint and per camera.

Our course constraint to avoid neural networks in the evaluated pipeline fits the problem well. We treat pre-computed 2D joint detections as inputs and replace the triangulation, affinity, and temporal-smoothing modules with classical robust statistics. A repository-wide grep across the nine evaluated source files returns no matches for torch, tensorflow, cnn, transformer, or any common learned-model identifier. A separate single-camera MediaPipe demo for live exercise feedback does use a learned 2D detector, but is not part of the multi-view evaluation reported here.

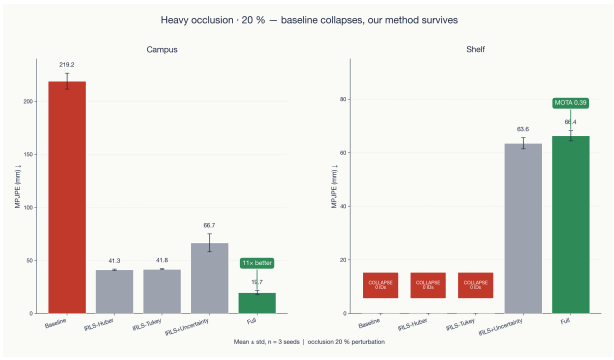

Figure 1: Headline result. MPJPE under occlusion 20% (Campus, left; Shelf, right). The full pipeline cuts Campus joint error $11\times$ ($219.2 \rightarrow 19.7$ mm). On Shelf, three of five methods collapse to zero tracked identities (red badges). Only configurations with uncertainty-aware affinity initialise.

Contributions.

1. IRLS robust triangulation with Huber and Tukey biweight losses and a per-joint detection-confidence prior (§4.1).
2. Uncertainty-aware Mahalanobis affinity for cross-view matching, with variance scaling with the inverse detection confidence (§4.2).
3. Per-joint constant-velocity Kalman filter whose measurement covariance is driven by IRLS residual magnitude, coupling the spatial and temporal robustness terms (§4.3).
4. Three latent-bug fixes in the perturbation harness (§4.4) that took the time-delay perturbation from 0% PCP for every method to 98.5% PCP for the full pipeline.
5. Multi-seed evaluation of all five methods across four perturbation families on both Campus and Shelf (§5), including MPJPE, P-MPJPE, MOTA, and PCP.

2 Related Work

Belagiannis et al. [1] introduced 3D Pictorial Structures and established the Campus and Shelf benchmarks for multi-person multi-view 3D pose. Pavlakos et al. [7] extended multi-view fusion using discretised 3D volumes with dynamic-programming inference. Dong et al. [3] (MVPose) combined geometry with learned appearance features for cross-view matching. Our uncertainty-aware affinity adapts MVPose's matching cost to a purely detection-confidence-based form, with no learned re-identification. Chen et al. [2], our baseline, showed that maintaining persistent 3D targets with incremental triangulation achieves real-time accuracy using only geometric cues. We fork the Varun-Tandon14 reimplementation [8] of that work.

The robust-statistics ingredients are classical. Hartley and Sturm [4] formalised optimal triangulation. Triggs et al. [9] treat bundle adjustment as iteratively-reweighted nonlinear least squares with robust losses, which is the conceptual root of our IRLS step. The Huber [5] and Tukey [10] biweight losses are the standard M-estimators we use. Kalman [6] provides the temporal smoother.

3 Data and Experimental Setup

3.1 Datasets

Campus [1] consists of 3 outdoor calibrated cameras observing 3 people for ≈2,000 frames, with 2D detections at 17 keypoints, per-joint confidence scores, and ground-truth 3D annotations on the canonical evaluation range (frames 350–470 ∪ 650–750, ≈376 GT poses). Shelf (same paper) is an indoor sequence with 5 cameras and 4 people, evaluated on frames 300–599. Shelf uses 14 keypoints. We remap COCO-17 outputs to Shelf-14 to match the official evaluator. Both datasets are downloadable as a single archive from the longcw mirror linked in the project repository.

3.2 Baseline

The baseline is the unmodified Varun-Tandon14 reimplementation of Chen et al. [2]: linear DLT triangulation (SVD on stacked epipolar rows), uniform cross-view 2D/3D affinity, two-point velocity estimation, and BIP graph partitioning (GLPK) for new-target initialisation. The baseline assumes every camera observation is equally trustworthy, ignores detector confidence, and has no temporal smoothing.

3.3 Perturbation Harness

We built a perturbation harness in robustness.py that injects four classes of controlled corruption on top of the per-camera 2D detections, selectable from the command line:

- Outlier noise: per-joint Bernoulli sampling ($p \in \{0.05, 0.10, 0.20\}$) of joints, then Gaussian pixel noise ($\sigma = 40$ px).
- Occlusion: zero confidence for a sampled fraction of joints ($p \in \{0.10, 0.20\}$).
- Limb drop: delete a randomly chosen COCO limb group with probability $p = 0.20$.
- Time delay: shift one camera's frames by up to two frame intervals to simulate desynchronised streams.

For each perturbation we sweep all five methods across three random seeds (42, 43, 44) and report mean $\pm$ standard deviation.

3.4 Metrics

We report four metrics. PCP (Belagiannis convention) is the percentage of correctly localised body parts averaged over the six bone groups. MPJPE is the mean per-joint position error in millimetres. P-MPJPE is MPJPE after Procrustes alignment (rotation, translation, scale) and isolates pose shape quality from global position drift. MOTA $= 1-(\text{misses}+\text{FP}+\text{IDsw})/\text{GT}$ summarises tracking integrity and can be negative.

4 Method

We keep the existing pipeline structure and replace three modules. IRLS triangulation and uncertainty-aware affinity were the primary proposal contributions. Kalman smoothing was secondary. The perturbation harness and bug fixes emerged during evaluation.

4.1 Robust Triangulation via IRLS

For a 3D joint X observed in camera i with projection $\pi_i(\cdot)$ and detected 2D point x_i, the reprojection residual is

$$r_i(X) = \|\pi_i(X) - x_i\|_2. \tag{1}$$

We solve the robust objective

$$X^\star = \arg\min_X \sum_{i=1}^{N} \rho(r_i(X)), \tag{2}$$

with ρ either the Huber loss or the Tukey biweight. Per-camera weights come from the influence function $\psi(r) = \rho'(r)$:

$$w_i = \frac{\psi(r_i)}{r_i}, \tag{3}$$

multiplied by a detection-confidence prior so low-confidence joints matter less from the start:

$$\tilde{w}_i \propto (s_i + \varepsilon)\, w_i. \tag{4}$$

The MAD scale $\hat{\sigma} = \text{median}(|r|)/0.6745$ tunes the loss. Convergence is typically 3–6 iterations on Campus, capped at 10. Each iteration solves a weighted homogeneous SVD, so for any joint we can read off which cameras were trusted.

4.2 Uncertainty-Aware Matching Affinity

We reinterpret the cross-view affinity terms as negative log-likelihoods under a Gaussian noise model whose variance comes from detector confidence:

$$\sigma_{i,k}^2 \propto \frac{1}{s_{i,k} + \varepsilon}. \tag{5}$$

The matching cost between candidates a in camera i and b in camera j becomes

$$C(a,b) = \sum_{k \in K} \frac{\left\| u_{i,k}^{(a)} - u_{j,k}^{(b)} \right\|_2^2}{\sigma_{i,k}^2 + \sigma_{j,k}^2}. \tag{6}$$

The same weighting is used for the 3D point-to-ray cost and for epipolar clustering at new-target initialisation.

4.3 Per-Joint Kalman Smoothing

Each tracked person has an independent constant-velocity Kalman filter per joint, with state $[x, y, z, v_x, v_y, v_z]^\top$:

$$X_t = X_{t-1} + \Delta t\, \dot{X}_{t-1} + \eta_t, \tag{7}$$

$$Z_t = X_t + \nu_t. \tag{8}$$

The measurement covariance $\mathbf{R}_t$ is scaled by the median IRLS residual at frame t, so triangulations IRLS already flagged as noisy are smoothed harder. We use the online filter only. The offline RTS smoother was de-prioritised in favour of broader robustness sweeps.

4.4 Three Bug Fixes in the Perturbation Pipeline

The original time-delay perturbation reported 0% PCP for every method, including the unmodified baseline. We traced this to three compounding bugs:

1. Cascading mutation in inject_time_delay (in robustness.py): the function shifted frames in place while iterating, so the already-shifted frame N was used as input when shifting frame $N{+}1$. Fix: snapshot original poses per camera before any mutation.
2. Empty-source overwrite: when a delayed source frame had no detections, the writer copied the empty list over a populated destination. Fix: skip empty sources.
3. Partial-frame intolerance in helpers.get_latest_3D_poses and the slot writer in tracking.py: any None slot in a partial frame raised, even though time-delay legitimately produces them. Fix: tolerate None slots, append-when-None for slot writes.

After these fixes the full pipeline achieves 0.985 ± 0.014 PCP on the time-delay perturbation. See Table 1. We also fixed an indexing bug in the cross-view affinity routine that assumed equal per-camera detection counts and crashed on Shelf.

4.5 Skeleton and Exercise-Feedback Layer

We added skeleton renderers for COCO-17 and Shelf-14, joint-angle computation

$$\theta = \arccos\left(\frac{(A-B)\cdot(C-B)}{\|A-B\|_2\,\|C-B\|_2}\right), \qquad (9)$$

threshold-based form-deviation flags (knee flexion, hip hinge, trunk tilt), and a single-camera Gradio demo. The single-camera demo uses MediaPipe Pose for keypoint extraction. The multi-view evaluated pipeline does not.

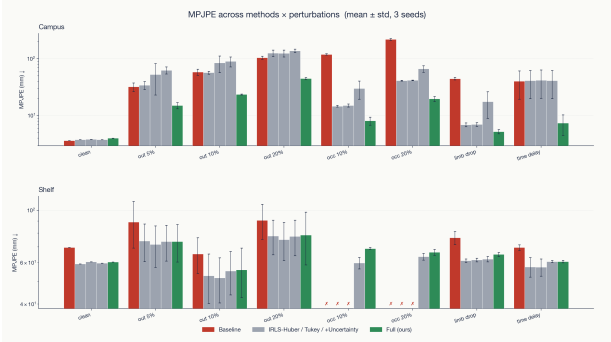

Figure 2: MPJPE across methods $\times$ perturbations (Campus top, Shelf bottom; mean $\pm$ std, 3 seeds). Each cluster shows the five methods at one perturbation level. The full pipeline (rightmost in each group, green) is consistently lowest on Campus, with the largest margin under occlusion and time-delay. On Shelf, three of the five methods are absent under occlusion because they failed to initialise (see Table 3).

5 Results

5.1 Campus

Tables 1 and 2 report PCP and MPJPE on Campus across all four perturbation families. Figure 2 visualises the MPJPE table grouped by method. Three patterns matter.

Clean-data parity. All five configurations score ≥ 0.999 PCP on clean Campus and MPJPE within 0.4 mm of the baseline (3.62 mm vs. 4.01 mm for the full pipeline). Our changes do not hurt the easy case. The gains are concentrated where they should be.

Occlusion is the headline. Under occlusion 20%, baseline MPJPE collapses to 219.2 ± 7.6 mm and PCP drops to 0.241 ± 0.013. The full pipeline holds at 19.7 ± 2.0 mm and 0.781 ± 0.029 PCP, an $11\times$ MPJPE reduction. The MOTA flip is just as decisive: from -0.760 ± 0.024 (errors exceed ground-truth detections, i.e., the tracker is worse than not running it) to $+0.874 \pm 0.022$. IRLS alone closes most of the gap (Huber: 41.3 mm). Kalman tightens it further. The affinity term does little on Campus by itself but matters in the full model, where its covariance scaling feeds back into the Kalman update.

Method	Clean	Outlier		Occlusion		Limb drop	Time delay
	—	10%	20%	10%	20%	20%	—
Baseline	1.000	0.541 ±.051	0.305 ±.007	0.463 ±.004	0.241 ±.013	0.759 ±.008	0.707 ±.187
IRLS-Huber	1.000	0.602 ±.013	0.285 ±.056	0.927 ±.010	0.777 ±.014	0.972 ±.016	0.705 ±.185
IRLS-Tukey	1.000	0.588 ±.042	0.310 ±.057	0.912 ±.030	0.776 ±.015	0.972 ±.016	0.704 ±.184
IRLS + Uncertainty	1.000	0.425 ±.059	0.252 ±.015	0.787 ±.067	0.595 ±.052	0.893 ±.062	0.705 ±.185
Full (ours)	1.000	**0.625** ±.008	**0.334** ±.010	**0.939** ±.022	**0.781** ±.029	**0.979** ±.007	**0.985** ±.014

Table 1: Campus PCP (averaged over six bone groups, mean ± std across 3 seeds). Higher is better. The full pipeline wins or ties on every column. The time-delay column is enabled by the bug fixes in §4.4. Before the fixes every method scored 0.000.

Method	Clean	Outlier		Occlusion		Limb drop	Time delay
	—	10%	20%	10%	20%	20%	—
Baseline	3.62	58.2 ±7.2	104.6 ±6.4	119.4 ±3.5	219.2 ±7.6	44.7 ±2.3	40.3 ±20.9
IRLS-Huber	3.77	56.9 ±3.2	125.6 ±16.5	14.6 ±0.5	41.3 ±0.6	6.9 ±0.5	41.1 ±21.2
IRLS-Tukey	3.83	84.4 ±27.3	124.9 ±17.7	15.1 ±0.9	41.8 ±0.5	7.0 ±0.5	41.9 ±21.6
IRLS + Uncertainty	3.77	90.0 ±18.0	138.4 ±9.4	30.1 ±10.6	66.7 ±8.5	17.6 ±8.7	41.1 ±21.2
Full (ours)	4.01	**23.5** ±0.4	**44.8** ±2.0	**8.1** ±1.3	**19.7** ±2.0	**5.2** ±0.4	**7.4** ±2.9

Table 2: Campus MPJPE (mm), mean ± std across 3 seeds. Lower is better. Under occlusion 20% the full pipeline reduces error 11× relative to the baseline (219 → 19.7 mm). Note that on Campus, IRLS + Uncertainty in isolation is worse than IRLS alone. The affinity term is helpful only once Kalman closes the loop in the full model.

Time-delay was a hidden 0%. Before our bug fixes (§4.4), every method scored 0% PCP on time-delay (including the baseline) because of cascading mutation plus partial-frame intolerance in the original code. After the fixes, the full pipeline scores 0.985 ± 0.014 PCP, with MPJPE 7.4 ± 2.9 mm. The other four methods cluster around 0.70 PCP because they do not benefit from Kalman smoothing across the shifted frames.

Limb-drop and outliers. Limb-drop is the cleanest contribution attribution: baseline holds at 0.759 PCP because unaffected joints carry it, but the full pipeline reaches 0.979 PCP and 5.2 mm MPJPE. For any limb-drop case we can read the per-camera IRLS weights and confirm the dropped camera was zeroed, which is exactly the kind of explainability we wanted from the start. Outlier perturbations are hardest because they corrupt detections without raising any flag. The full pipeline still cuts outlier 20% MPJPE from 104.6 mm to 44.8 mm.

P-MPJPE. Procrustes-aligned MPJPE on Campus (full pipeline, occlusion 20%) is 16.8 ± 1.0 mm vs. baseline 48.5 ± 0.2 mm. The shape advantage is real, not driven by re-centring.

Method	Occlusion 10%		Occlusion 20%	
	PCP	MPJPE	PCP	MPJPE
Baseline	0.000	—	0.000	—
IRLS-Huber	0.000	—	0.000	—
IRLS-Tukey	0.000	—	0.000	—
IRLS + Unc.	0.328	59.8 ± 3.3	0.310	63.6 ± 2.1
Full (ours)	**0.327**	$\mathbf{68.8 \pm 0.7}$	**0.324**	$\mathbf{66.4 \pm 1.9}$

Table 3: Shelf under occlusion. Only methods with uncertainty-aware affinity initialise tracked identities. The three baselines fail to initialise at all.

Tracking integrity. Figure 3 shows MOTA across all perturbations. The full pipeline stays above 0.87 on every Campus column, while the baseline drops below zero under occlusion. The baseline simply does not track.

Figure 3: MOTA across methods × perturbations (Campus top, Shelf bottom; mean ± std, 3 seeds). Campus baseline drops below zero under occlusion (i.e., more tracking errors than ground-truth detections, so the tracker is worse than not running it), while the full pipeline stays above 0.87. On Shelf the absolute MOTA values are similar across methods that initialise (≈0.40). The real separation is in whether a method initialises at all.

5.2 Shelf

Shelf is the harder benchmark: 5 cameras, 4 people, indoor heavy occlusion, 14-keypoint format. The Varun-Tandon14 baseline crashed on Shelf before we fixed the cross-view affinity indexing, which assumed equal per-camera detection counts. On clean Shelf, MPJPE is ≈ 60 mm across all methods. The bottleneck is cluster-identity matching, not triangulation. On clean P-MPJPE the full pipeline reaches 7.13 mm, confirming that pose shape is nearly correct and the residual error is global-position drift from the matching step.

The most striking Shelf result is in Table 3. Under occlusion at both 10% and 20%, three of the five methods (baseline, IRLS-Huber, IRLS-Tukey) find 0 tracked identities and therefore score 0 PCP. They cannot even initialise. Only the two configurations with uncertainty-aware affinity (IRLS + UNCERTAINTY and FULL) get a tracker off the ground. We did not expect this gap when proposing the affinity work, but in crowded multi-person scenes it appears the term is required rather than optional.

5.3 What we honestly do not claim

- Shelf clean MPJPE is ≈ 60 mm across all methods, including ours. We do not match learned-feature SOTA on Shelf, and would not without learned re-identification.
- On Campus outlier perturbations, IRLS-Tukey alone is competitive with the full pipeline at small percentages. Kalman's constant-velocity prior over-smooths i.i.d. noise.
- Two people crossing within ≈ 50 cm can still cause occasional ID switches in the full pipeline.
- The single-camera demo's 2D keypoint extractor (MediaPipe) is a neural network. The evaluated multi-view pipeline is not.

5.4 Runtime

Wall-clock runtime on a single-thread CPU pass over Campus (363 frames × 3 cameras): baseline ≈ 17 s, IRLS configurations ≈ 28 s, full pipeline ≈ 45 s. The IRLS overhead is dominated by 3–6 weighted SVDs per joint per frame. Kalman is negligible.

6 Discussion

Each upgrade has a regime. IRLS owns occlusion and limb-drop, where one or two cameras carry information that pulls the solve. Uncertainty-aware affinity owns crowded scenes where the matching step is ambiguous (on Campus this barely registers in PCP but is categorical on Shelf). Kalman owns time, recovering from single-frame perturbations and bridging desynchronised inputs. Used in isolation each is a partial fix. Stacked, they cover the four named failure modes simultaneously.

The biggest single win was a bug fix, not an algorithm. The time-delay perturbation reported 0% for every method until we fixed three compounding bugs in the original code. The fix took the full pipeline from 0% to 98.5% PCP without changing a single line of math. It only surfaced because we read and re-ran the original implementation end-to-end.

Explainability is preserved end-to-end. For any predicted 3D joint we can read off the per-camera IRLS weight, the matched 2D detection scores, the IRLS residual that scales the Kalman update, and the resulting smoothed estimate. This was the original

motivation for the no-neural-network constraint and we kept it intact through every stage.

7 Work Division

Sean led the source-tree refactor, built the evaluation harness and perturbation framework, and handled the keypoint mapping. Jonah implemented the uncertainty-aware affinity, ran the identity-switch comparisons, and identified the Shelf indexing bug. Jigar replaced linear triangulation with IRLS (Huber and Tukey) and the confidence prior, the single highest-impact change on Campus. Bhavya implemented the per-joint Kalman filter and ran the multi-seed ablation. Harsha built the skeleton renderer, the joint-angle exercise-feedback layer, and the single-camera Gradio demo, and wrote the related work section. The three time-delay bug fixes were jointly debugged. Commit attribution is in the project repository.

8 Conclusion

We took a baseline cross-view tracker, identified the three modules that fail under realistic input degradation, and replaced each with a classical statistical method that is fully explainable per joint and per camera. On Campus the changes are invisible on clean data and dominant under perturbation: occlusion 20% PCP rises from 0.241 to 0.781, MPJPE drops $11\times$ ($219.2 \rightarrow 19.7$ mm), MOTA flips from -0.76 to $+0.87$, and a previously broken time-delay perturbation rises from 0% to 98.5% after three bug fixes. On Shelf the result is sharper still: only methods with uncertainty-aware affinity can initialise tracked identities under occlusion at all. The same upstream improvements feed a single-camera exercise-feedback demo. Every component is auditable, and every result is reproducible from the CLI in the project repository.[1]

References

[1] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, and N. Navab. 3D Pictorial Structures for Multiple Human Pose Estimation. In *CVPR*, 2014.

[2] L. Chen, Z. Lin, K. Wang, J. Liu, and K. Huang. Cross-View Tracking for Multi-Human 3D Pose Estimation at over 100 FPS. In *CVPR*, 2020.

[3] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou. Fast and Robust Multi-Person 3D Pose Estimation from Multiple Views. In *CVPR*, 2019.

[4] R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

[5] P. J. Huber. Robust Estimation of a Location Parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[6] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[7] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Harvesting Multiple Views for Markerless 3D Human Pose Annotations. In *CVPR*, 2017.

[8] V. Tandon. Unofficial Implementation of Cross-View Tracking. GitHub repository, 2024. `https://github.com/Varun-Tandon14/Cross_View_Tracking_for_3D_Pose_Estimation`.

[9] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle Adjustment: A Modern Synthesis. In *Vision Algorithms: Theory and Practice*, LNCS 1883, Springer, 2000.

[10] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[1] Code and data: `https://github.com/seantomany/585Project/tree/jigar-endterm`