

STSR 2026 — Technical Report (Team JLShen)

Team: JLShen (Codabench user `junlong`). Tasks entered: Task 1, Task 2, Task 3.
Method names: HiFiTooth (Task 1) · XJawReg (Task 2) · CoverRec (Task 3).

All numbers below are official validation-phase scores returned by the challenge server.

Task 1 — Semi-supervised tooth & pulp instance segmentation in CBCT

Method: HiFiTooth — mirror-free ResEnc nnU-Net with high-fidelity label export

Model. nnU-Net v2 `3d_fullres` with the **ResEnc-L** planner (`nnUNetResEncUNetLPlans` , patch 192×128×224, 0.30 mm isotropic, per-image z-score, 65 output classes = background + 32 tooth FDI + 32 pulp), trained 250 epochs on the 30 labelled cases. Single fold (fold 0) is the submitted container; a 5-fold ensemble was also evaluated.

Mirroring is disabled. The metric is *per-FDI-label*, so a left–right flip does not preserve the target: mirror augmentation improves foreground overlap but scrambles tooth identity. Removing it was the single largest design decision; a mirror-trained variant lost IA 0.627 → 0.598 on held-out data.

High-fidelity export (the main engineering contribution). The usual memory-frugal path — argmax at model resolution, then nearest-neighbour upsampling to the native grid — silently caps accuracy. Resampling the **softmax** to native resolution and taking argmax there gained **+0.013 Dice_Image / +0.011 mean-metric on the server with identical weights**. To keep this export inside the 8 GB host-RAM evaluation cap, the ~16 GB fp16 65-channel logit tensor is kept **GPU-resident** and reduced per channel with a running argmax, so host RAM only ever holds the input volume and a 1-channel label map.

No external data, no pretrained weights, no TTA, no post-processing — all post-processing variants (largest-component, drop-tiny, morphological closing, anatomy-based pulp rules) were measured and none improved the server score.

Validation results (20 hidden cases):

entry	Dice_Img	mIoU_Img	NSD_Img	Dice_Inst	mIoU_Inst	NSD_Inst
single fold (submitted)	.9624	.9279	.9956	.7131	.6514	.8023
5-fold						

5-fold ensemble	.9638	.9305	.9959	.7110	.6542	.7949
organiser baseline	.9565	.9169	.9964	.3849	.2838	.5421

Container: 4.11 GB image, 3.3–22.5 s/case, peak host RSS 8.39 GB under `-m 8G`, peak GPU 21.2 GB (RTX 3090-compatible).

Negative result worth reporting. Every accuracy lever that improved 5-fold CV on the 30 labelled cases **regressed on the held-out server**: offline self-training on the 299 unlabelled cases (CV IA +0.052 → server IA −0.018, worst of all entries), pulp class re-weighting (CV IA +0.048 → server IA ±0.00), left–right mirror augmentation, 10-model ensembling (−0.013 mean), a larger ResEnc-XL backbone, and 0.25 mm training (rare-class collapse: all 16 upper-pulp channels died mid-training). A paired bootstrap confirmed the CV gains were statistically real on those 30 cases — they were *selection* overfitting to a 30-case set, not noise. Our final entry is therefore purely supervised on the 30 labelled cases; we could not make the unlabelled pool pay on held-out data. The residual bottleneck is **pulp**: 391 of 431 missed labels are pulp channels (teeth ≈95 % correctly identified, pulp ≈35 %), a thin-structure / resolution limit at 0.30 mm.

Task 2 — IOS ↔ CBCT rigid registration

Method: XJawReg — surface registration with a handedness prior and cross-jaw decoy repair

Predicts one 4×4 matrix per jaw mapping IOS crown coordinates to the CBCT world frame.

- Handedness prior.** $\det(R)$ is a scanner signature: CBCT volumes with exactly 0.3 mm isotropic spacing require a reflection ($\det = -1$), all others $\det = +1$. Verified 30/30 on labelled data and on all 380 available CBCT headers, with no near-threshold case. Both jaws of a patient share the sign and are decided jointly. This removed the dominant failure mode.
- Target surface.** Marching cubes on CBCT intensity > 800 restricted to within 8 mm of enamel cores (> 2400), i.e. the crown envelope the IOS actually sees.
- Global search, not ICP-from-identity.** Rotation candidates = de-duplicated training-GT rotations plus a canonical cap grid within the $\det$ group; per candidate a coarse-then-fine translation search, then a small-radius point-to-plane ICP; selection by a tight bidirectional surface score.
- Per-tooth correspondence.** Pretrained tooth-segmentation models (ToothGroupNetwork on the IOS mesh, a CBCT FDI segmenter on the volume) give FDI-to-FDI centroid Procrustes poses. These are decoy-free and excellent on **lower** jaws; a GT-free gate

(residual < 4 mm and canonical distance < 40) accepts 17/50 of them. On upper jaws cross-modal FDI labels do not agree (0–5 of 14 matched) and this route was abandoned.

5. **Cross-jaw decoy repair — the largest single gain.** Upper and lower jaw share the head pose (agreement $\leq 6.16^\circ$ / ≤ 6.5 mm across all 30 labelled patients). Where an upper estimate disagrees with the reliable lower pose by > 10 mm it is a mirror/flip decoy, so the lower pose is copied onto it. This cut mean translation error **11.94 mm → 9.90 mm**.
6. **Metric decoupling.** The scorer reads R and t independently, so for borderline uppers we take the *rotation* from the reliable lower jaw while keeping the *translation* from the own estimate: $3.5171^\circ \rightarrow 3.4583^\circ$ at unchanged translation.

Final validation score: 9.9045 mm mean translation error / 3.4583° mean rotation error (over 100 jaw poses). Local leave-out CV on the 30 labelled cases: mean 8.09 mm / 3.95° , **median 3.34 mm / 1.26°** — the mean is carried by ~10 outlier jaws, all of them "common-mode" cases where both jaws fail together, which the cross-jaw prior cannot repair.

Ruled out (measured, all worse on the held-out server): pretrained GeoTransformer + TEASER++ cross-modal registration (works on 5–10 % of jaws, unreliable per case); semi-supervised pseudo-labelling with the 300 unlabelled cases (injects the classical method's own decoy bias); a learned pose verifier trained leave-one-patient-out; trust-region decoy re-registration. Detection of a failed jaw is easy; *re-registering* it is not, because the failure is a segmentation/correspondence failure that any re-run inherits.

Disclosure: Task 2 uses publicly pretrained dental segmentation models (permitted by the rules; no external training data was used).

Task 3 — MMDental report generation

Method: CoverRec — a metric-calibrated, coverage-optimised record prior

The task asks for a 7-field dental record per CBCT case, scored by an undisclosed `Weighted_Score` blending text metrics (BLEU 1–4, METEOR, ROUGE-L, CIDEr), entity precision/recall/F1 (tooth notations, diagnosis codes, treatment actions, medications), diagnosis exact match, and field completion.

Since the scorer was not public, we **reverse-engineered it from the server**: a dependency-light pure-Python proxy scorer plus nine calibration submissions whose per-component breakdowns were downloaded and fitted. Findings: GT records are consolidated as *first non-empty* visit value; empty-vs-empty entity comparisons score 1.0; CIDEr is on a 0–1 scale; and the blend is dominated by **text overlap (≈ 0.47) plus coverage/recall terms**, not per-case entity precision.

That analysis says the optimum is a *single* record that maximises overlap with the common vocabulary of real records — and the coverage curve confirms a clear interior

optimum (over-stuffing degrades ROUGE-L faster than recall rises). The submitted method emits one train-derived, coverage-tuned 7-field record for every case. It is stdlib-only Python, needs no GPU, and runs in under a second per case.

Validation Weighted_Score:

variant	score
minimal generic text, no entities	0.1856
medoid record (per field, short form)	0.2769
coverage-tuned, light / medium	0.3088 / 0.3501
coverage-tuned, heavy (submitted)	0.3517
over-stuffed	0.3435

The honest finding is negative and, we think, the useful one for this task. Every image-conditioned method we tried scored *below* the image-free constant: nearest-neighbour retrieval on CBCT features (1762-dim, LOO-CV 0.21–0.22 vs 0.287 for the constant), an image→record-archetype classifier (below the majority baseline at every K), and per-case anatomy→diagnosis prediction (AUC 0.52–0.75, and touching only low-weight components). At the feature level we could reach, the CBCT volume carries essentially no recoverable signal about the patient's textual record, and the metric rewards vocabulary coverage over per-case specificity. We report this as a calibration result about the benchmark rather than as a report-generation model.

Reproducibility

Both Docker deliverables (`JLShen.tar.gz` , Task 1 and Task 3) run offline as specified by the organisers (`docker run --gpus all -m 8G -v ../inputs -v ../outputs jlshen:latest`), with the build recipe and a full 20/20-case validation log included alongside each image. Task 2 is a CPU-only offline pipeline; its predicted matrices reproduce bit-exactly from the released scripts. Code and weights are available on