ZEROSYNC: LEARNING MOTION MATCHING FROM PAIRED RELATIVE TIMESTAMP SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Estimating unknown time offsets between heterogeneous sensors requires matching motion observed at different rates and with varying reliability. Sparse or repeated motion can produce ambiguous alignment peaks, making fixed correlation scores unreliable. We introduce ZeroSync, which learns temporal matching through controlled timestamp shifts in a common motion representation. Sensor-specific front ends convert observations into angular-velocity sequences for a shared scoring network. Shifting only one stream in time preserves the recorded motion and supplies supervision without absolute clock labels. Shared branches refine component- and block-specific correlations, while the known shift supervises their relative offset distribution. A confidence head learns recovery success using observable and motion-erased examples. At evaluation, the difference between refined branch offsets estimates the correction that reverses the injected shift. Experiments on camera–IMU and LiDAR–IMU robot recordings show that learned residual scoring improves paired recovery without confidence selection. The improvement persists under sequence-equal averaging in each test dataset. These findings support learning a shared motion-matching rule across sensor pairs from known relative timing changes.

1 INTRODUCTION

Mobile robots combine camera or LiDAR measurements with an IMU for state estimation (Qin et al., 2018; Shan et al., 2020; Xu et al., 2022). Timestamp mismatches associate observations with the wrong inertial intervals, motivating temporal calibration (Qin & Shen, 2018). Our application goal is to estimate unknown sensor time offsets. We study how to learn the underlying motion-matching rule without absolute hardware-clock labels.

Rotational motion provides a common signal across sensors. Cameras and LiDAR estimate rotation between observations, while gyroscopes measure angular velocity directly. Prior work estimates timing jointly with spatial calibration or visual–inertial states (Furgale et al., 2013; Li & Mourikis, 2014; Taylor & Nieto, 2016; Qin & Shen, 2018); motion-based synchronization also uses correlation and window-level reliability assessment (Faizullin et al., 2021; Wodtko et al., 2024). We study learned matching when the motion evidence is sparse or ambiguous.

Frame and scan rotations summarize motion over intervals, whereas gyroscopes sample angular velocity more densely. Tracking failures, repeated turns, and weak motion can produce ambiguous correlation peaks. We test whether learning from separate motion components and temporal blocks improves matching over their fixed average.

ZeroSync uses controlled timestamp shifts as supervision for learning this matching rule. Sensor-specific front ends convert observations into angular-velocity sequences in a common body frame. We duplicate a recorded window and shift only the non-reference timestamps, keeping motion values and the IMU reference unchanged. The estimated offset should change by the injected shift. This trains the response of the matcher to timing changes without requiring the original offset as a label.

Shared branches learn bounded corrections to the mean correlation, extending learning over explicit matching hypotheses (Kendall et al., 2017; Chang & Chen, 2018; Teed & Deng, 2020) to sensor timing. Paired supervision targets the relative offset distribution, while a confidence head predicts

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

recovery success. Our evaluation measures the known offset change from the difference between branch estimates, rather than accuracy against an absolute clock reference.

We make three contributions. We formulate relative timing recovery through common motion sequences and known shifts of one stream, linking paired supervision to timestamp correction. We develop a shared residual scorer over component- and block-specific correlations, with separate objectives for offset recovery and confidence. We evaluate both sensor pairs without confidence selection, compare a simpler learned scorer, and examine sparse observations, selective prediction, and continuous shifts.

2 Related Work

Temporal calibration from sensor motion. Temporal calibration is often integrated with spatial calibration (Furgale et al., 2013; Taylor & Nieto, 2016) or visual–inertial state estimation (Li & Mourikis, 2014; Qin & Shen, 2018). ZeroSync assumes known rotational extrinsics and learns motion-matching scores without jointly estimating scene geometry or extrinsic parameters.

Correlation is a classical approach to delay estimation (Knapp & Carter, 1976). Twist-n-Sync optimizes correlations of high-rate gyroscope signals (Faizullin et al., 2021). Wodtko et al. (2024) use rotational motion in sliding windows to estimate time-varying offsets, assess uncertainty, and correct timestamps. We build on motion-based synchronization by learning how to score component- and block-specific correlations. Our comparisons evaluate external methods on shared motion inputs under the protocol described below.

Learning over matching volumes. GC-Net and PSMNet learn over stereo cost volumes (Kendall et al., 2017; Chang & Chen, 2018); RAFT uses correlations to update optical flow (Teed & Deng, 2020). ZeroSync applies this principle to candidate time offsets, learning corrections to a fixed motion-correlation prior. Comparisons with the residual disabled isolate the contribution of learned scoring under the same readout.

Supervision from temporal relationships. Temporal structure can provide training signals without semantic annotations. Shuffle and Learn learns visual representations by verifying frame order (Misra et al., 2016). ZeroSync uses timestamp shifts of known magnitude to supervise the relationship between two offset distributions. A separate confidence objective predicts recovery success, including on examples with erased motion.

3 Method

Figure 1 follows the method from common motion sequences through timestamp shifts, shared scoring, and paired supervision to timing recovery. Each branch receives an IMU sequence paired with camera or LiDAR motion. Before resampling, the original and shifted inputs differ only in the sensor timestamps.

3.1 From heterogeneous observations to motion sequences

Let r be the IMU reference and s a camera or LiDAR rigidly attached to the same robot. Known rotational extrinsics map their motion into a common body frame. Each front end produces $\mathcal{M}_s = \{(t_i, \omega_s(t_i), q_s(t_i))\}_i$, containing timestamps, angular velocities, and quality scores. Camera motion uses Shi–Tomasi features (Shi & Tomasi, 1994), pyramidal Lucas–Kanade tracking, and RANSAC essential-matrix rotation recovery. Recordings configured as near-pure rotation use a calibrated homography. LiDAR motion uses ICP/GICP registration between adjacent scans (Segal et al., 2009). For relative rotation $R_{i\to i+1}$ and sensor-to-body rotation R_{Bs}, we compute

$$\omega_s(\bar{t}_i) = R_{Bs}\,\frac{\mathrm{Log}(R_{i\to i+1}^{-1})^{\vee}}{t_{i+1}-t_i}, \qquad \bar{t}_i = \frac{t_i + t_{i+1}}{2}. \tag{1}$$

The inverse follows the front-end convention; $(\cdot)^{\vee}$ maps the matrix logarithm to a rotation vector. Equation 1 approximates midpoint angular velocity, with error under varying motion. IMU readings retain their acquisition timestamps.

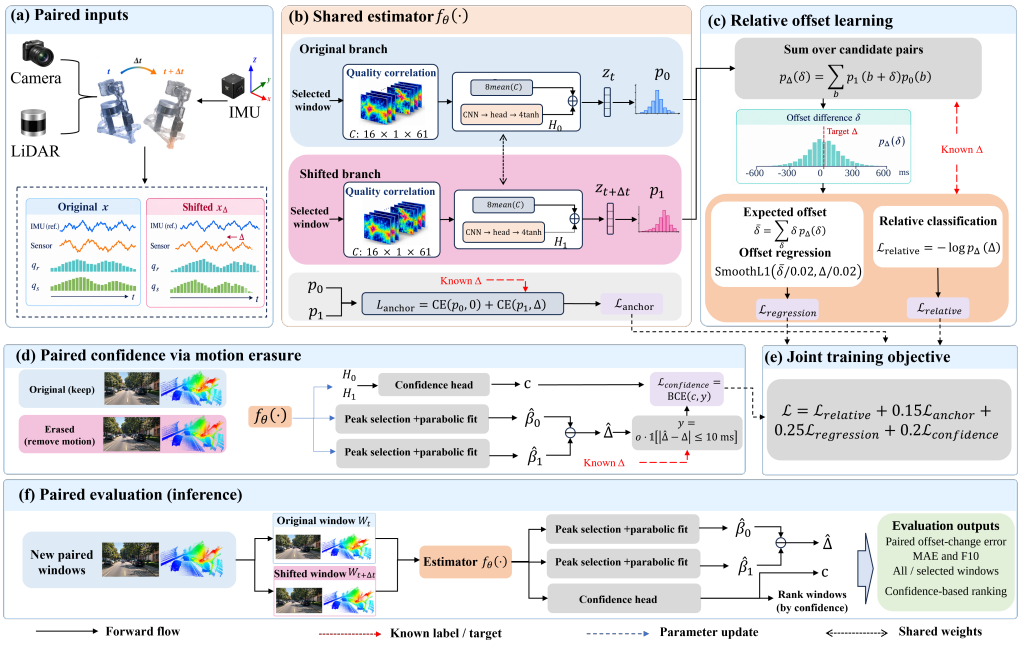

Figure 1: Paired supervision and timing recovery. Camera or LiDAR motion is paired with an IMU reference. Only sensor timestamps shift between inputs. Shared branches refine correlations under relative offset supervision. Observable samples train offset recovery; confidence also uses motion-erased samples (Eq. 9). Refined branch peaks yield the relative timestamp correction. Sensor pairs are alternative inputs; curves and volumes are schematic.

Each stream also carries a quality score $q_s(t) \in [0, 1]$. Camera quality combines tracking support, inlier ratio, and blur response; LiDAR quality combines registration fitness and residual error. Failed estimates receive a quality score of zero. We use four motion channels, $v_s(t) = [\omega_{s,x}, \omega_{s,y}, \omega_{s,z}, \|\omega_s\|_2]^\top$: the components retain direction, while the norm supplies angular-speed evidence.

This representation lets the same matcher process both sensor pairs while keeping image tracking and scan registration in their respective front ends.

3.2 Shifting one motion sequence to construct supervision

A deterministic angular-speed correlation estimate compensates the native sensor delay relative to the IMU before we construct supervision (Appendix A.2). It defines the starting time coordinates, not a measured clock reference.

An original window x contains the reference and sensor motion sequences. We form x_Δ by changing only the sensor timestamps to $t'_s = t_s - \Delta$. Motion values and quality scores remain attached to the same samples; the IMU sequence is unchanged. Under the constant-offset relation $t_r = t_s + \beta$, this shift increases the required sensor-to-reference correction by Δ.

For a matcher f_θ that returns a branch offset, the recovered timestamp change is

$$\widehat{\Delta} = f_\theta(x_\Delta) - f_\theta(x). \tag{2}$$

For example, advancing sensor timestamps by 100 ms requires a 100 ms increase in the estimated correction. The known Δ supplies supervision even when the original absolute offset is unknown. We evaluate Δ within ± 200 ms with zero clock drift, using both windows.

We linearly resample 40 s windows at 50 Hz. Each channel is median-centered and scaled by its interquartile range over valid samples, with standard deviation as a fallback and a 10^{-6} floor. Values are clipped to $[-8, 8]$; invalid samples are filled with zero and excluded through quality weights. This controls outliers and scale differences before correlation.

3.3 COMPARING MOTION UNDER CANDIDATE TIME CORRECTIONS

Each branch asks how closely the two motion sequences agree under each candidate time correction. Let $u_r(t_n)$ and $u_s(t_n)$ denote the normalized motion signals. A candidate $b_j \in \mathcal{B}$ assigns sensor time t_s to reference time $t_s + b_j$. We therefore compare the reference at t_n with sensor motion and quality interpolated at $t_n - b_j$. We use $\mathcal{B} = \{-0.30, -0.29, \ldots, 0.30\}$ s, giving $J = 61$ candidates. To reduce the influence of unreliable observations, each comparison receives weight

$$w_{nj} = q_r(t_n)\, q_s(t_n - b_j)\, m_{nj}, \tag{3}$$

where m_{nj} is one within valid interpolation support and zero otherwise. A low quality score in either stream reduces the contribution of that time point.

We divide the window into $K = 4$ contiguous blocks $\mathcal{T}_k$ to retain local matching evidence. For channel d, write $x_{nd} = u_{r,d}(t_n)$ and $y_{ndj} = u_{s,d}(t_n - b_j)$. Within each block, weighted means center the two signals before their correlation is computed:

$$W_{kj} = \max\left(\sum_{n\in\mathcal{T}_k} w_{nj}, \eta\right), \quad \mu^r_{kdj} = \frac{\sum_{n\in\mathcal{T}_k} w_{nj} x_{nd}}{W_{kj}}, \quad \mu^s_{kdj} = \frac{\sum_{n\in\mathcal{T}_k} w_{nj} y_{ndj}}{W_{kj}}. \tag{4}$$

These means depend on the candidate offset because the weights and overlapping samples change with b_j. The weighted correlation then measures whether the centered signals rise and fall together:

$$C_{kdj} = \frac{\sum_{n\in\mathcal{T}_k} w_{nj}(x_{nd} - \mu^r_{kdj})(y_{ndj} - \mu^s_{kdj})}{\max\Big(\sqrt{\sum_{n\in\mathcal{T}_k} w_{nj}(x_{nd} - \mu^r_{kdj})^2 \sum_{n\in\mathcal{T}_k} w_{nj}(y_{ndj} - \mu^s_{kdj})^2}, \eta\Big)}. \tag{5}$$

The denominator normalizes by the variation in both signals; $\eta = 10^{-6}$ prevents division by zero, and correlations are clipped to $[-1, 1]$. The four blocks and four channels yield 16 score curves, forming $C \in \mathbb{R}^{16\times 1\times J}$ with a singleton drift axis. Keeping the curves separate preserves local evidence for competing offsets.

3.4 LEARNING A SHARED RESIDUAL SCORING RULE

The network learns how to combine the matching evidence in C. Shared branches process the original and shifted volumes, producing distributions p_0 and p_1 over candidate corrections. Two padded 3×3 convolutions with 32 channels and GELU activations map C to features $H_\theta(C) \in \mathbb{R}^{32\times 1\times J}$. A 1×1 head g_θ predicts a score adjustment for each candidate. The corrected logits and branch distribution are

$$\bar{C}_j = \frac{1}{16}\sum_{k,d} C_{kdj}, \qquad z_j = 8\bar{C}_j + 4\tanh g_\theta(C)_j, \qquad p(b_j) = \frac{\exp z_j}{\sum_{\ell=1}^{J} \exp z_\ell}. \tag{6}$$

The convolutions combine block and component evidence across neighboring offsets. The mean correlation provides an initial score; the bounded residual adjusts it using the full volume and can change the preferred peak. Parameter sharing applies the same matching rule to both branches and sensor pairs.

3.5 SUPERVISING OFFSET RECOVERY AND CONFIDENCE

Training requires the estimated correction to increase by the injected shift. A candidate b in the original branch and $b + \delta$ in the shifted branch together support a change of δ. We multiply their probabilities and sum over all such candidate pairs:

$$p_\Delta(\delta) = \sum_{b\in\mathcal{B}} p_1(b + \delta) p_0(b), \qquad \bar{\delta} = \sum_{\delta\in\mathcal{B}-\mathcal{B}} \delta\, p_\Delta(\delta). \tag{7}$$

Values outside $\mathcal{B}$ have zero probability. The resulting relative distribution has $2J-1$ entries at 10 ms spacing, and $\bar{\delta}$ is its expected offset change. It retains all candidate pairs during training, including those associated with competing branch peaks. We apply offset supervision to windows that pass

motion-excitation and quality thresholds, termed observable windows below. For each such window, the offset loss is

$$\ell_{\text{off}} = -\log p_{\Delta}(\Delta) + 0.15[-\log p_0(0) - \log p_1(\Delta)] + 0.25\,\text{SmoothL1}(\bar{\delta}/0.02, \Delta/0.02). \tag{8}$$

The first term assigns probability to the known difference Δ. Auxiliary anchors target zero and Δ in the precompensated coordinates, rather than independent hardware-clock labels. Regression penalizes expected offset error at a 0.02 s scale. Evaluation instead uses refined peaks, as specified below.

The confidence head predicts whether the paired estimate meets an error tolerance. We flatten and concatenate H_0 and H_1, then apply an MLP with a 32-unit hidden layer, GELU, and sigmoid output c_i. Its target is $y_i = o_i \mathbf{1}[|\widetilde{\Delta}_i - \Delta_i| \le \tau_c]$, where o_i is the observability label and $\tau_c = 0.01$ s. The detached branch difference $\widetilde{\Delta}_i$ is evaluated before output clipping. A positive target therefore requires an observable window and error within 10 ms. Artificial motion erasure retains quality channels but sets $o_i = 0$. With sample weights a_i, the batch objective is

$$\mathcal{L} = \frac{\sum_i a_i o_i \, \ell_{\text{off},i}}{\max(\sum_i a_i o_i, \eta)} + 0.2 \frac{\sum_i a_i \, \text{BCE}(c_i, y_i)}{\max(\sum_i a_i, \eta)}. \tag{9}$$

The first term averages offset losses over observable windows; the second averages confidence losses over all windows, including erased examples. The coefficient 0.2 balances the objectives. Sample weights a_i weight whole windows, unlike the pointwise quality weights in Eq. 3. Observability is a threshold-based training label, while c_i learns recovery success and ranks windows for selective evaluation.

3.6 Recovering motion-sequence timing

We recover the injected timing change by refining and subtracting the two branch offsets. For each branch, we select $i = \arg\max_j z_j$. A three-point parabolic fit around an interior peak gives a sub-grid displacement at spacing $h = 0.01$ s:

$$D_i = z_{i-1} - 2z_i + z_{i+1}, \qquad \hat{\beta} = b_i + h \,\text{clip}\left(\frac{z_{i-1} - z_{i+1}}{2D_i}, -\tfrac{1}{2}, \tfrac{1}{2}\right). \tag{10}$$

Boundary indices are clamped to the nearest interior index before fitting; displacement is zero when $D_i \ge -10^{-6}$. Subtracting the refined offsets gives $\widehat{\Delta}$ in Eq. 2; evaluation reports $\widehat{\Delta}_{\text{eval}} = \text{clip}(\widehat{\Delta}, -0.2, 0.2)$ s.

The corrected timestamps are $t_s^{\text{restored}} = t'_s + \widehat{\Delta}_{\text{eval}}$. Exact recovery restores the pre-injection timestamps without changing angular velocities. We evaluate the resulting timing error, $\widehat{\Delta}_{\text{eval}} - \Delta$.

Subtracting the branch offsets cancels their shared additive delay, although estimation errors can still depend on that delay. This recovers a relative change, not an absolute clock reference.

4 Experiments

We test residual scoring, deterministic baselines, sampling rates, and confidence selection on camera–IMU and LiDAR–IMU recordings.

4.1 Sensor pairs and evaluation protocol

Robot recordings and data splits. Camera–IMU evaluation uses EuRoC (Burri et al., 2016) sequences S01–S10. LiDAR–IMU evaluation uses urban M2DGR (Yin et al., 2022) sequences S11–S16 and off-road RELLIS-3D (Jiang et al., 2021) sequences S17–S21. The same scoring network processes both sensor pairs, each with an IMU reference.

Training uses TartanAir (Wang et al., 2020) P000/P001 and M2DGR rotation_01, street_02, and street_03; validation uses TartanAir P002 and M2DGR street_05. TartanAir training motion is obtained from pose pairs. Appendix A.2 specifies the training protocol, and Appendix A.3 gives the test sequence key. One MS^2 (Shin et al., 2023) camera–IMU sequence provides an additional transfer evaluation (Appendix B.5).

From recordings to test pairs. Before motion extraction, we decimate camera observations to approximately 20, 10, 5, and 2 Hz and LiDAR observations to 10, 5, and 2 Hz. Sections 3.1–3.2 describe extraction, precompensation, and pair construction. Resampling 40 s windows at 50 Hz does not restore removed observations; decimation also changes the rotation-estimation intervals.

Every method estimates offsets for the original and shifted inputs; their difference is compared with the known injection. The rate sweep and matched-grid tests inject offsets at 10 ms spacing within ± 200 ms and search within ± 300 ms. Matched-grid comparisons use 10 ms candidate spacing for all methods. Continuous-offset tests vary weighted L1 spacing from 10 to 1 ms. We exclude 1 Hz settings and recordings with native-delay estimates near the search boundary.

The rate sweep covers 21 sequences and 73 configurations. A separate comparison uses 700 windows per operating point: S01/S02 and S11/S15 at 2 Hz, and S17 at 10 Hz. Appendix A.1 specifies these distinct evaluation populations.

Baselines and matched inputs. We compare angular-speed correlation, weighted vector L2, Huber, weighted and unweighted L1, the upstream Basalt alignment objective, Twist-n-Sync (Faizullin et al., 2021), and GCC-PHAT (Knapp & Carter, 1976). Basalt denotes the alignment component of the Basalt project (Usenko et al., 2020). Weighted L1 is our quality-weighted, interpolated absolute-deviation implementation, labeled `basalt_l1`, and is distinct from upstream Basalt. Methods share motion inputs but retain their own interpolation, refinement, and invalid-output handling. Equal grid spacing therefore does not imply equal computational cost.

Within each experiment, methods receive identical input pairs and motion erasures. Primary comparisons retain all windows; selective prediction retains the top 70% within each source, ranked by ZeroSync confidence. We also analyze non-erased windows and alternative selectors. The pooled study contains 25,900 windows. Continuous-offset comparisons retain 490 of 700 windows drawn at stride eight from 5,600 predictions, with membership fixed across L1 search resolutions.

Metrics. For absolute offset-change error $e_j = |\widehat{\Delta}_{\mathrm{eval},j} - \Delta_j|$ and retained set $\mathcal{S}$, we report

$$\kappa = \frac{|\mathcal{S}|}{N}, \qquad \mathrm{MAE}_{\mathcal{S}} = \frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}} e_j, \qquad F_\tau = \frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}} \mathbf{1}[e_j \geq \tau]. \tag{11}$$

MAE is reported in ms and F_{10} in percent. Appendix A.4 also reports success at 20 ms. Success uses $e_j \leq \tau$, so it overlaps with failure at the exact threshold. At 2 Hz, sequence-level comparisons use the lowest rival F_{10}. Results are descriptive because windows may overlap.

4.2 Recovery across camera–IMU and LiDAR–IMU pairs

Full and Prior only use identical windows, correlation volumes, and peak readout. Prior only disables the learned residual and differs from the angular-speed Correlation baseline. Across all 25,900 windows, pooled MAE is 8.581 ms for Full, 8.704 ms for Prior only, and 9.447 ms for weighted L1 (Table 3).

The 2,011 motion-erased windows have the same pooled MAE of 96.171 ms for all three estimators. On all 23,889 non-erased windows, Full reduces MAE by 9.91% relative to Prior only and 43.74% relative to weighted L1 (Table 1). This subset uses only the synthetic-erasure flag, without observability or confidence filtering.

Sequence-equal averages in Table 1 preserve this ordering for camera–IMU, urban LiDAR–IMU, and off-road LiDAR–IMU. Full has the highest error on urban LiDAR recordings. These groups differ in scenes, motion, and sampling rates, so their errors cannot be attributed to sensor modality alone.

Appendix B.1 reports all internal objectives; Appendix B.3 gives dataset macro averages. Full also has lower all-window MAE than a separately trained pointwise scorer on both canonical test and generalization splits (Appendix B.2).

Table 1: Offset-change recovery without confidence selection on all 23,889 non-erased windows across 73 configurations. The upper block pools windows; the lower block reports MAE averaged equally over rates within each sequence and then over sequences in each group. Lower is better.

Metric	Full	Prior only	Weighted L1
Window-weighted pooled results			
MAE (ms)	**1.208**	1.341	2.147
F_{10} (%)	**1.77**	1.91	10.51
Sequence-equal MAE (ms) by sensor pair and environment			
Camera–IMU	**1.094**	1.196	1.941
Urban LiDAR–IMU	**1.570**	1.733	1.878
Off-road LiDAR–IMU	**1.132**	1.312	3.062

4.3 Full-coverage comparison with deterministic alignment

Across the five operating points, ZeroSync has the lowest F_{10} among nine matchers on all 700 windows, including motion erasures (Table 2). Failures decrease by 3.86–11.43 percentage points relative to weighted L1. MAE rankings differ: weighted L1 leads on S02 and S11/S15, while ZeroSync leads on S01 and S17 (Table 11). ZeroSync has lower MAE and F_{10} than all three external implementations under this shared-input protocol.

Table 2: Full-coverage F_{10} (%) for all nine methods, with 700 windows per point and no confidence filtering. S01/S02/S11/S15: 2 Hz; S17: 10 Hz. Lower is better; column minima are bold.

	Camera–IMU		Urban LiDAR–IMU		Off-road LiDAR–IMU
Method	S01	S02	S11	S15	S17
ZeroSync	**12.71**	**12.00**	**14.57**	**16.86**	**9.14**
Correlation	29.86	29.14	28.14	37.57	16.29
L2	26.43	24.86	27.14	45.29	17.43
Huber	24.14	25.57	26.43	37.86	16.86
Weighted L1	24.14	23.14	18.43	21.86	14.86
Unweighted L1	25.57	23.43	21.43	28.29	15.00
GCC-PHAT	90.57	86.71	88.71	87.14	47.29
Twist-n-Sync	64.14	62.86	70.43	68.29	54.43
Basalt	39.71	41.14	36.57	38.43	33.43

Basalt and Twist-n-Sync are evaluated on shared inputs as alignment components. Appendix A.4 documents their readout and fallback behavior and reports confidence-selected comparisons and tolerance sensitivity.

4.4 Selective prediction across sensors and sampling rates

Selective comparisons use the same windows for every method, ranked by ZeroSync confidence. The subset is therefore model-dependent; Sections 4.2–4.3 report results without confidence selection.

Figure 2 compares ZeroSync with the lowest failure rate among all five internal deterministic objectives at each configuration. Table 5 and Appendix A.3 report every method and sequence under this coverage rule.

At 2 Hz and 70% coverage, ZeroSync lowers F_{10} relative to the best internal rival on all 10 camera sequences and 10 of 11 LiDAR sequences (Table 5). Reductions range from 2.85 to 27.35 percentage points. Median F_{10} is approximately 1.84% for camera–IMU, 7.14% for urban LiDAR–IMU, and 3.67% for off-road LiDAR–IMU, versus 13.06%, 12.45%, and 22.04% for the rival envelope. On S13, Correlation instead achieves 6.53%, compared with 8.16% for ZeroSync.

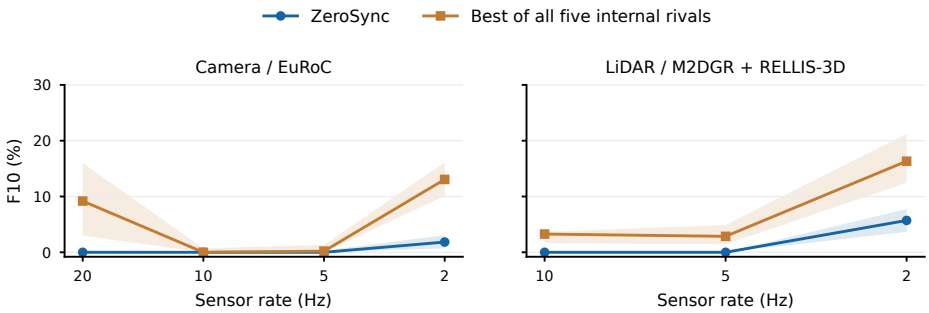

Figure 2: Sampling-rate sensitivity at 70% coverage ranked by ZeroSync confidence. Lines show median F_{10}; bands show across-sequence interquartile ranges. The rival envelope takes the minimum over the five internal deterministic objectives per configuration. Camera and LiDAR panels contain 10 and 11 sequences.

At 5 and 10 Hz, ZeroSync records no threshold failures on the tested LiDAR sequences under the same coverage rule. A camera rival with zero failures outperforms ZeroSync on S07 at 5 and 10 Hz and S06 at 5 Hz. Appendix D illustrates two effects of sampling: sparse observations can miss brief motion, and frame intervals affect camera rotation recovery.

Coverage and selector comparisons. Full has the lowest pooled MAE under full coverage and all four 70% selection rules (Table 3). Confidence selection reduces its MAE from 8.581 to 1.008 ms, whereas random selection gives 8.726 ms. This reduction reflects a change in retained windows, distinct from the residual gain on matched windows.

Table 3: Window-weighted pooled MAE (ms) across 73 configurations. No selection retains all windows; the other rows retain 70%. Prior only disables the residual.

Selector	n	Full	Prior only	Weighted L1
none (100% of windows)	25900	**8.581**	8.704	9.447
model confidence	18130	**1.008**	1.145	2.025
motion excitation (model-free)	18130	**8.355**	8.470	9.230
input quality (model-free)	18130	**8.275**	8.413	9.243
random control	18130	**8.726**	8.851	9.584

Figure 3 summarizes the residual and L1 comparisons for these populations and the continuous-offset test.

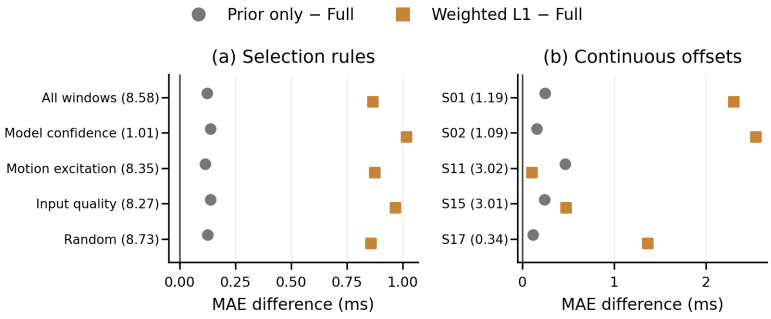

Figure 3: MAE differences favoring Full when positive; parentheses give Full MAE (ms). (a) Pooled selectors (Table 3). (b) Continuous offsets, 70% confidence coverage, 1 ms L1 (Table 4). Points are aggregates; panel scales and populations differ.

On S15 at 2 Hz and 70% coverage, two label-free prior-peak selectors retain MAE reductions of 10.38% and 3.41% relative to Prior only, with lower F_{10} (Appendix B.4). The residual benefit on this sequence thus also occurs without learned confidence selection.

4.5 Continuous offsets and fine deterministic search

We sample offsets continuously and uniformly within ± 200 ms, independently of the model grid. S01/S02 and S11/S15 use 2 Hz inputs; S17 uses 10 Hz. Table 4 compares Full, Prior only, and weighted L1 at 1 ms spacing under confidence selection. Appendix A.6 reports the complete 10, 5, 2, and 1 ms sweep.

Table 4: Continuous-injection MAE (ms) / F_{10} (%) at 70% coverage. Bold marks minima among the methods shown; the full sweep is in Appendix A.6. Runs differ from Table 7.

Point	Full	Prior only	L1, 1 ms
S01	**1.187 / 0.4**	1.432 / 0.8	3.487 / 4.9
S02	**1.093 / 0.8**	1.253 / **0.8**	3.634 / 5.9
S11	**3.019 / 5.1**	3.487 / 6.5	3.126 / 5.5
S15	**3.007 / 5.7**	3.247 / 6.9	3.485 / **5.7**
S17	**0.341 / 0.0**	0.457 / **0.0**	1.705 / **0.0**

At 70% coverage, Full has the lowest MAE at all five points, but 5 ms L1 has lower F_{10} on S11 (4.5% versus 5.1%). The 1 ms L1 ties Full on S15, as do several resolutions on S17. Appendix A.6 reports the different threshold rankings under discrete injections and timing details.

4.6 Shared-delay sensitivity and anchor ablation

On S15, $+50$ ms shared delay raises MAE from 2.976 to 4.111 ms with confidence reselection (Appendix B.6). Retraining without the anchor changes mean MAE by $+0.001$ ms over 25 conditions, without proving equivalence. Main results retain the anchor.

5 Conclusion

ZeroSync learns shared motion matching from paired timestamp shifts and uses the recovered change to correct sequence timing. Residual scoring reduces mean recovery error on fixed windows for both sensor pairs; continuous-offset and sampling-rate tests characterize this advantage.

Limitations. We evaluate controlled shifts, not absolute calibration or downstream fusion. Windows may overlap; shared-delay tests reselect them. Fine search remains competitive. Pointwise cross-seed stability and interval-integrated motion matching remain untested.

AI use statement

Generative AI tools assisted manuscript drafting, revision, translation, LaTeX formatting, and the preparation of figure-generation prompts and the method illustration.

Reproducibility statement

Sections 3–4.1 describe the method and evaluation protocol; Appendices A–D give training settings, expanded comparisons, and implementation diagnostics. We plan to release code and experiment configurations.

REFERENCES

Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W. Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016. doi: 10.1177/0278364915620033.

Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5410–5418, 2018. doi: 10.1109/CVPR.2018.00567.

Marsel Faizullin, Anastasiia Kornilova, Azat Akhmetyanov, and Gonzalo Ferrer. Twist-n-Sync: Software clock synchronization with microseconds accuracy using MEMS-gyroscopes. *Sensors*, 21(1):68, 2021. doi: 10.3390/s21010068.

Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1280–1286, 2013. doi: 10.1109/IROS.2013.6696514.

Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. RELLIS-3D dataset: Data, benchmarks and analysis. In *IEEE International Conference on Robotics and Automation*, pp. 1110–1116, 2021. doi: 10.1109/ICRA48506.2021.9561251.

Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *IEEE International Conference on Computer Vision*, pp. 66–75, 2017. doi: 10.1109/ICCV.2017.17.

C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, 1976. doi: 10.1109/TASSP.1976.1162830.

Mingyang Li and Anastasios I. Mourikis. Online temporal calibration for camera–IMU systems: Theory and algorithms. *The International Journal of Robotics Research*, 33(7):947–964, 2014. doi: 10.1177/0278364913515286.

Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pp. 527–544, 2016. doi: 10.1007/978-3-319-46448-0_32.

Tong Qin and Shaojie Shen. Online temporal calibration for monocular visual-inertial systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3662–3669, 2018. doi: 10.1109/IROS.2018.8593603.

Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018. doi: 10.1109/TRO.2018.2853729.

A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Robotics: Science and Systems*, 2009. doi: 10.15607/RSS.2009.V.021.

Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5135–5142, 2020. doi: 10.1109/IROS45743.2020.9341176.

Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994. doi: 10.1109/CVPR.1994.323794.

Ukcheol Shin, Jinsun Park, and In So Kweon. Deep depth estimation from thermal image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1043–1053, 2023. doi: 10.1109/CVPR52729.2023.00107. URL https://openaccess.thecvf.com/content/CVPR2023/html/Shin_Deep_Depth_Estimation_From_Thermal_Image_CVPR_2023_paper.html.

Zachary Taylor and Juan Nieto. Motion-based calibration of multimodal sensor extrinsics and timing offset estimation. *IEEE Transactions on Robotics*, 32(5):1215–1229, 2016. doi: 10.1109/TRO.2016.2596771.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pp. 402–419, 2020. doi: 10.1007/978-3-030-58536-5_24.

Vladyslav Usenko, Nikolaus Demmel, David Schubert, Jörg Stückler, and Daniel Cremers. Visual-inertial mapping with non-linear factor recovery. *IEEE Robotics and Automation Letters*, 5(2):422–429, 2020. doi: 10.1109/LRA.2019.2961227.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4909–4916, 2020. doi: 10.1109/IROS45743.2020.9341801.

Thomas Wodtko, Alexander Scheible, and Michael Buchholz. Self-assessment and correction of sensor synchronization. *arXiv preprint arXiv:2409.20266*, 2024. URL https://arxiv.org/abs/2409.20266.

Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. FAST-LIO2: Fast direct LiDAR-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022. doi: 10.1109/TRO.2022.3141876.

Jie Yin, Ang Li, Tao Li, Wenxian Yu, and Danping Zou. M2DGR: A multi-sensor and multi-scenario SLAM dataset for ground robots. *IEEE Robotics and Automation Letters*, 7(2):2266–2273, 2022. doi: 10.1109/LRA.2021.3138527.

A Additional experimental setup and results

This appendix expands the evaluation in Section 4. Camera–IMU uses S01–S10; urban and off-road LiDAR–IMU use S11–S16 and S17–S21. Each experiment keeps the IMU reference fixed and measures recovery of a known shift in the sensor motion timestamps. We describe the evaluation populations and training protocol before presenting the additional comparisons.

A.1 Evaluation populations

Experiments with different window populations are reported separately, using the sequence IDs from the main text. The rate sweep covers 73 sequence–rate configurations across 21 sequences. The pooled selector study includes 350 windows for each of 72 configurations and 700 windows for S17 at 10 Hz, totaling 25,900. At 70% coverage, these configurations retain 245 or 490 windows each, totaling 18,130. The five-point matched-grid comparison uses a separate set of 700 windows per point and retains 490; S01/S02/S11/S15 use 2 Hz and S17 uses 10 Hz. Repeat runs sample different windows, and the S15 erasure study uses 350 windows. Observable-window residual ablations select samples by motion and quality labels, not confidence.

The continuous-offset evaluation generates 5,600 predictions per point. Taking every eighth sample yields 700 comparison windows, of which confidence ranking retains 490. S01/S02 use camera–IMU at 2 Hz, S11/S15 use LiDAR–IMU at 2 Hz, and S17 uses LiDAR–IMU at 10 Hz. Non-erased results and macro averages are recomputed from the matched per-window records.

F_{10} counts errors ≥ 10 ms; tolerance success counts errors $\leq \tau$. The resulting rates need not sum to unity at the boundary. All MAEs are in milliseconds. Paired counts below are descriptive: overlapping windows are not independent experimental replications.

A.2 Training and checkpoint protocol

Training uses AdamW with micro-batches of 384 pairs and two-step gradient accumulation, giving a nominal effective batch size of 768. The learning rate is 5×10^{-6}, weight decay is 10^{-4}, and gradients are clipped at 5. Two warmup epochs precede a cosine schedule with a minimum learning rate of 10^{-6}. The budget is 80 epochs with an early-stopping patience of 12. The recorded history spans epochs 0–62.

The reported model corresponds to the lowest validation selection score in the saved training history. The main tables aggregate errors across evaluation windows for this model, rather than across training seeds. Appendix B.3 reports sequence-equal macro averages.

Each training epoch samples 56,000 paired windows. The validation, test, and generalization configurations each specify 5,600 windows. TartanAir training motion comes from pose pairs. Sampling assigns LiDAR sources weight 2 and other sources weight 1, with weights normalized across sources. LiDAR examples with injected offset magnitude of at least 100 ms receive loss weight 2; other examples receive weight 1. Weighted losses are normalized by the sum of weights.

Artificial motion erasure occurs with probability 0.1 and removes motion channels while retaining quality channels. The observability rule is applied to valid reference, original-sensor, and shifted-sensor segments. Each must have a motion-magnitude standard deviation of at least 0.01 and mean quality of at least 0.05. Erased examples are marked unobservable. Offset losses use observable examples; confidence uses all examples with detached success labels, as in Eq. 9.

Native-delay precompensation uses quality-weighted angular-speed correlation over the common sensor interval. The reference is interpolated at 50 Hz, and the search covers ± 300 ms at 2 ms spacing. Valid overlap must contain at least 20 samples and 25% of the common grid. The estimated correction is added to sensor timestamps before experimental shifts are injected; it is not an independent clock reference. For the 70% comparisons, Full confidence ranks windows within each source, and every method uses the top $\mathrm{round}(0.7N)$. The saved 25% confidence-calibration setting controls inference flags, not selection in these tables.

A.3 COMPLETE RATE SWEEP

Table 5: Sequence comparison at 2 Hz on the shared 70% confidence-selected subset. Entries are 10 ms failure rates (%, lower is better); the lowest value in each row is bold. All listed methods cover all 21 sequences. Basalt, GCC-PHAT, and Twist-n-Sync are compared on a separate five-point population in Table 7.

ID	Correlation	L2	Huber	Weighted L1	L1	ZeroSync
EuRoC						
S01	22.45	14.29	13.88	13.47	12.24	**0.41**
S02	19.18	9.80	11.43	9.39	10.20	**1.63**
S03	16.73	17.14	17.14	13.88	17.14	**2.04**
S04	20.41	10.61	9.80	9.80	10.61	**1.63**
S05	11.43	7.76	9.39	10.61	8.16	**0.00**
S06	23.27	40.41	16.73	15.51	38.37	**3.27**
S07	16.33	37.96	24.90	24.08	40.41	**8.16**
S08	20.82	13.88	11.02	12.24	14.29	**0.00**
S09	22.45	20.41	19.18	17.55	20.82	**2.45**
S10	38.37	37.55	35.51	34.29	44.49	**6.94**
M2DGR						
S11	15.51	17.96	17.96	7.76	12.65	**3.67**
S12	13.88	10.61	10.20	15.92	14.69	**7.35**
S13	**6.53**	10.61	10.20	12.24	11.43	8.16
S14	35.10	30.61	22.45	16.33	17.55	**8.16**
S15	28.98	34.29	25.71	14.69	20.00	**5.71**
S16	40.41	33.47	26.12	20.41	26.53	**6.94**
RELLIS-3D						
S17	15.51	14.69	15.92	21.63	21.22	**3.67**
S18	22.04	25.31	27.76	23.67	25.31	**3.67**
S19	25.71	24.90	29.80	35.10	34.29	**8.98**
S20	19.18	19.59	21.63	23.27	23.67	**2.04**
S21	26.53	26.12	28.98	26.12	26.53	**4.90**

Figure 4 shows individual sequences alongside the median trend in each dataset. Many objectives have elevated failure rates at 2 Hz, particularly on LiDAR sequences. Camera baselines also have elevated failures at 20 Hz, indicating a nonmonotonic relationship with sampling rate. Decimation changes both observation spacing and rotation-estimation intervals, and the plot combines these effects.

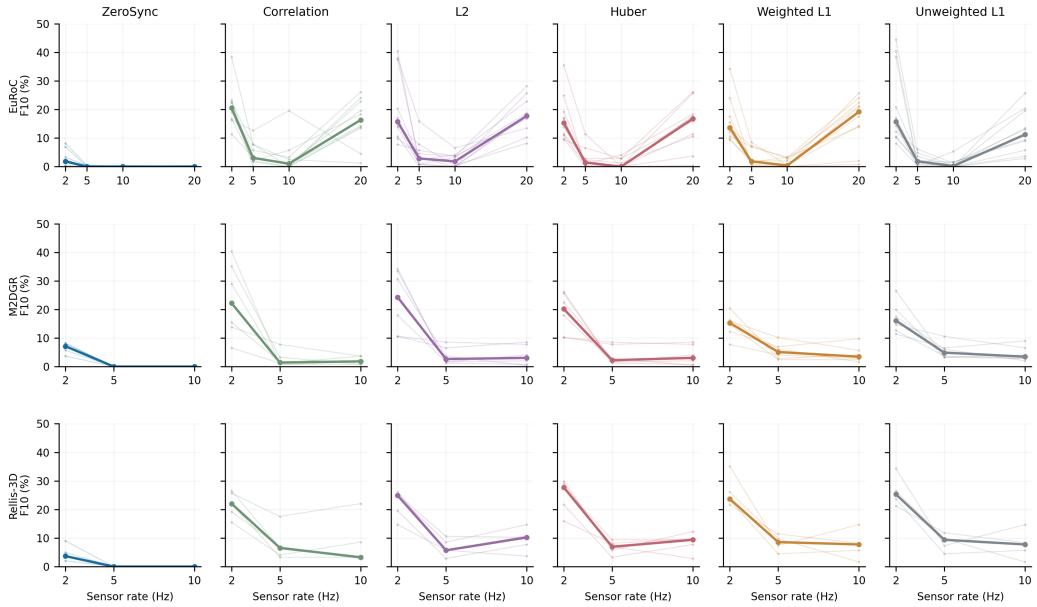

Figure 4: Sampling-rate sensitivity for six internal methods across 73 sequence–rate configurations. Thin lines show individual sequences; thick lines show the median across sequences at each available rate. Rows correspond to EuRoC, M2DGR, and RELLIS-3D. All panels report F_{10} on the shared 70% confidence-selected population within each configuration. Across-sequence variation is not a confidence interval, and no 20 Hz LiDAR result is implied.

The following table reports all internal objectives at every available rate. External implementations appear in the separate five-point comparison. The best rival is selected independently for each configuration, so the identity of the comparator can change with sequence and rate.

Table 6: Complete rate sweep: confidence-selected F_{10} (%).

ID	Hz	Full	W. L1	Corr.	Huber	U. L1	L2
S01	20	0.41	21.22	1.22	10.61	13.06	8.16
S01	10	0.00	2.86	2.45	1.22	1.22	0.41
S01	5	0.00	7.35	7.76	2.45	6.12	7.76
S01	2	0.41	13.47	22.45	13.88	12.24	14.29
S02	20	0.00	13.88	14.29	15.92	8.98	16.73
S02	10	0.00	3.27	0.82	2.86	1.63	3.67
S02	5	0.00	1.63	2.86	2.86	2.04	4.49
S02	2	1.63	9.39	19.18	11.43	10.20	9.80
S03	20	0.00	17.55	13.88	17.55	5.71	17.55
S03	10	0.00	0.41	0.00	0.00	0.00	0.00
S03	5	0.41	1.63	2.04	0.41	2.45	0.82
S03	2	2.04	13.88	16.73	17.14	17.14	17.14
S04	20	0.00	20.82	26.12	17.55	19.59	17.96
S04	10	0.00	0.00	3.27	0.00	0.00	0.00
S04	5	0.00	2.04	5.71	0.00	1.63	0.00
S04	2	1.63	9.80	20.41	9.80	10.61	10.61
S05	20	0.00	24.08	19.59	26.12	20.41	28.16
S05	10	0.00	3.27	1.22	2.86	5.31	3.67
S05	5	0.00	6.94	3.27	6.53	0.41	5.71
S05	2	0.00	10.61	11.43	9.39	8.16	7.76
S06	20	0.00	25.71	22.86	25.71	25.71	25.71

ID	Hz	Full	W. L1	Corr.	Huber	U. L1	L2
S06	10	0.00	0.00	0.00	0.00	0.00	0.00
S06	5	0.41	0.41	1.63	0.00	0.00	0.82
S06	2	3.27	15.51	23.27	16.73	38.37	40.41
S07	20	0.00	2.04	4.49	3.67	2.86	13.47
S07	10	0.41	0.00	19.59	0.00	0.00	6.53
S07	5	0.41	0.82	12.65	0.00	0.82	15.92
S07	2	8.16	24.08	16.33	24.90	40.41	37.96
S08	20	0.00	22.45	18.37	18.37	9.39	18.78
S08	10	0.00	2.04	5.71	4.08	1.63	3.27
S08	5	0.00	2.04	2.04	0.82	0.00	0.82
S08	2	0.00	12.24	20.82	11.02	14.29	13.88
S09	20	0.00	0.82	13.47	0.41	3.67	10.20
S09	10	0.00	0.00	0.00	0.00	0.00	0.00
S09	5	0.00	1.63	0.00	2.04	3.67	3.27
S09	2	2.45	17.55	22.45	19.18	20.82	20.41
S10	20	0.00	14.29	24.08	11.43	13.47	22.86
S10	10	0.00	0.41	0.00	0.00	0.41	4.08
S10	5	0.00	8.57	7.76	11.43	4.90	2.45
S10	2	6.94	34.29	38.37	35.51	44.49	37.55
S11	10	0.00	2.86	2.04	2.04	2.86	2.04
S11	5	0.00	4.08	1.63	1.63	3.67	1.63
S11	2	3.67	7.76	15.51	17.96	12.65	17.96
S12	10	0.00	5.71	3.67	7.76	6.53	7.76
S12	5	0.00	10.20	7.76	8.57	10.61	8.57
S12	2	7.35	15.92	13.88	10.20	14.69	10.61
S13	10	0.00	9.80	3.67	8.57	8.98	8.57
S13	5	0.00	6.94	1.22	7.76	6.53	6.53
S13	2	8.16	12.24	6.53	10.20	11.43	10.61
S14	10	0.00	1.63	0.82	0.41	2.04	0.41
S14	5	0.00	6.12	3.27	2.86	6.12	3.67
S14	2	8.16	16.33	35.10	22.45	17.55	30.61
S15	10	0.00	2.45	0.82	0.82	2.86	0.82
S15	5	0.00	2.45	0.82	1.22	3.27	1.22
S15	2	5.71	14.69	28.98	25.71	20.00	34.29
S16	10	0.00	4.08	1.63	4.08	4.08	4.08
S16	5	0.00	2.86	0.82	1.63	3.27	1.63
S16	2	6.94	20.41	40.41	26.12	26.53	33.47
S17	10	0.00	8.16	8.57	9.80	8.37	10.82
S17	5	0.00	11.43	4.08	8.16	11.84	5.71
S17	2	3.67	21.63	15.51	15.92	21.22	14.69
S18	10	0.00	7.76	3.27	9.39	7.76	10.20
S18	5	0.00	8.57	3.27	9.39	9.39	10.61
S18	2	3.67	23.67	22.04	27.76	25.31	25.31
S19	10	0.00	14.69	22.04	12.24	14.69	14.69
S19	5	0.00	7.35	17.55	5.71	7.35	8.57
S19	2	8.98	35.10	25.71	29.80	34.29	24.90
S20	10	0.00	5.71	3.27	7.76	5.71	7.76
S20	5	0.00	4.49	6.94	3.27	4.49	2.86
S20	2	2.04	23.27	19.18	21.63	23.67	19.59
S21	10	0.00	1.63	2.86	2.86	1.63	3.67
S21	5	0.00	9.80	6.53	6.94	9.80	5.71
S21	2	4.90	26.12	26.53	28.98	26.53	26.12

Sequence key: EuRoC MH_01_easy (camera) = S01; EuRoC MH_02_easy (camera) = S02; EuRoC MH_03_medium (camera) = S03; EuRoC MH_04_difficult (camera) = S04; EuRoC V1_01_easy (camera) = S05; EuRoC V1_02_medium (camera) = S06; EuRoC V1_03_difficult (camera) =

Table 7: Matched 10 ms search grid: F_{10} (%, lower is better) on a shared 70% confidence-selected population. S15, S11, S01, S02: 2 Hz; S17: 10 Hz. Basalt and Twist-n-Sync use upstream implementations on shared motion inputs.

Method	S15	S11	S01	S02	S17
ZeroSync	**6.12**	**3.88**	**0.61**	**1.02**	**0.00**
Correlation	28.16	17.76	22.65	19.18	8.57
L2	35.31	16.73	12.86	11.02	10.82
Huber	28.37	17.55	11.22	12.45	9.80
Weighted L1	13.67	8.57	13.67	10.41	8.16
Unweighted L1	18.78	12.45	12.65	10.82	8.37
GCC-PHAT	86.73	87.55	89.59	84.69	41.22
Twist-n-Sync	67.14	67.14	60.82	61.43	51.02
Basalt	29.80	30.41	31.43	34.08	25.92

S07; EuRoC V2_01_easy (camera) = S08; EuRoC V2_02_medium (camera) = S09; EuRoC V2_03_difficult (camera) = S10; M2DGR street_01 (LiDAR) = S11; M2DGR street_04 (LiDAR) = S12; M2DGR street_06 (LiDAR) = S13; M2DGR street_07 (LiDAR) = S14; M2DGR street_08 (LiDAR) = S15; M2DGR street_09 (LiDAR) = S16; Rellis-3D 00000 (LiDAR) = S17; Rellis-3D 00001 (LiDAR) = S18; Rellis-3D 00002 (LiDAR) = S19; Rellis-3D 00003 (LiDAR) = S20; Rellis-3D 00004 (LiDAR) = S21.

A.4 Selective matched-grid comparisons and tolerance sensitivity

All tables in this subsection use 70% coverage ranked by ZeroSync confidence. The retained windows are shared across methods, but their selection is model-dependent.

Basalt uses a nearest-sample objective with 20 ms readout granularity on 50 Hz signals. Twist-n-Sync raises exceptions in 134 of 1,400 calls per operating point; non-finite paired outputs receive a zero-change fallback. These comparisons evaluate shared-input alignment components rather than complete systems.

Table 8: Five-point matched-grid results: F_{10} (%) / MAE (ms). Dashes denote unreported MAEs for internal objectives; no values are inferred from rounded differences.

Method	S15	S11	S01	S02	S17
Correlation	28.16 / 8.612	17.76 / 4.388	22.65 / –	19.18 / –	8.57 / –
L2	35.31 / 7.245	16.73 / 3.816	12.86 / –	11.02 / –	10.82 / –
Huber	28.37 / 5.714	17.55 / 3.490	11.22 / –	12.45 / 2.776	9.80 / –
Weighted L1	13.67 / **2.755**	8.57 / **2.204**	13.67 / 2.714	10.41 / 2.204	8.16 / 1.898
Unweighted L1	18.78 / 3.673	12.45 / 2.776	12.65 / –	10.82 / –	8.37 / –
GCC-PHAT	86.73 / 96.286	87.55 / 90.653	89.59 / 90.204	84.69 / 74.204	41.22 / 10.204
Twist-n-Sync	67.14 / 30.858	67.14 / 26.839	60.82 / 17.895	61.43 / 17.016	51.02 / 10.676
Basalt	29.80 / 6.265	30.41 / 6.122	31.43 / 5.673	34.08 / 6.735	25.92 / 5.163
ZeroSync	**6.12** / 2.976	**3.88** / 2.570	**0.61 / 1.142**	**1.02 / 1.133**	**0.00 / 0.324**

Table 9: Success at 10 / 20 ms (%) on the same 490 retained windows per point.

Method	S15	S11	S01	S02	S17
Correlation	71.8 / 89.2	82.2 / 95.3	77.3 / 96.5	80.8 / 97.1	91.4 / **100.0**
L2	64.7 / 88.8	83.3 / 95.3	87.1 / 97.8	89.0 / 99.6	89.2 / **100.0**
Huber	71.6 / 93.1	82.4 / 96.5	88.8 / 99.0	87.6 / 99.4	90.2 / **100.0**
Weighted L1	86.3 / **98.8**	91.4 / **99.4**	86.3 / 99.4	89.6 / **99.8**	91.8 / **100.0**
Unweighted L1	81.2 / 98.2	87.6 / 98.4	87.3 / 99.4	89.2 / 99.4	91.6 / **100.0**
GCC-PHAT	13.3 / 20.2	12.4 / 20.8	10.4 / 18.6	15.3 / 27.8	58.8 / 86.1
Twist-n-Sync (upstream)	32.9 / 56.5	32.9 / 53.9	39.2 / 67.8	38.6 / 66.7	49.0 / 68.0
Basalt (upstream)	70.2 / 96.3	69.6 / 97.1	68.6 / 99.0	65.9 / 95.9	74.1 / **100.0**
ZeroSync	**93.9** / 97.8	**96.1 / 99.4**	**99.4 / 100.0**	**99.0** / 99.6	**100.0 / 100.0**

Table 10: Repeated sampling of different urban LiDAR windows: F_{10} (%) / MAE (ms).

Method	S11	S15
Basalt	29.90 / 6.092	30.71 / 6.306
GCC-PHAT	87.04 / 94.204	87.55 / 95.684
Twist-n-Sync	65.00 / 25.978	66.94 / 30.402
ZeroSync	**5.92 / 3.361**	**8.88 / 4.014**

A.5 Full-coverage comparison at five operating points

All 700 windows at each operating point are evaluated without confidence selection, including the original motion erasures. S01/S02 and S11/S15 use 2 Hz inputs; S17 uses 10 Hz. Tables 2 and 11 report all nine methods. ZeroSync has the lowest F_{10} at every point and lower MAE than GCC-PHAT, Twist-n-Sync, and Basalt under the shared-input protocol. Some internal objectives achieve lower MAE, as discussed in the main text.

Table 11: Full-coverage MAE (ms) on the same windows. Lower is better; column minima are bold.

	Camera–IMU		Urban LiDAR–IMU		Off-road LiDAR–IMU
Method	S01	S02	S11	S15	S17
ZeroSync	**12.228**	13.045	12.551	13.124	**10.206**
Correlation	13.871	13.957	16.071	19.314	11.314
L2	14.200	13.571	13.843	17.471	11.471
Huber	13.114	13.343	13.414	15.986	11.400
Weighted L1	12.900	**12.814**	**12.086**	**12.714**	11.214
Unweighted L1	13.557	13.257	12.657	13.886	11.229
GCC-PHAT	95.886	83.457	94.986	95.957	18.900
Twist-n-Sync	26.614	26.087	32.708	39.487	19.355
Basalt	16.086	16.314	15.429	15.829	14.571

A.6 Continuous offsets, search resolution, and timing

Offsets are sampled uniformly within ± 200 ms. Each operating point retains 490 of 700 comparison windows drawn at stride eight from 5,600 saved predictions. Window identities, model predictions, and selection by Full confidence remain fixed across L1 spacings of 10, 5, 2, and 1 ms. Prior only predictions are matched by source and window start time and evaluated on the same retained windows. Only the deterministic L1 search spacing varies; model parameters and the candidate grid remain fixed.

Finer deterministic search narrows some MAE gaps without consistently improving every metric (Figure 5). Full has lower MAE than every displayed L1 setting at each operating point. On S11, however, 5 ms L1 achieves $F_{10} = 4.5\%$, compared with 5.1% for Full. Horizontal lines show fixed model results across the L1 search resolutions.

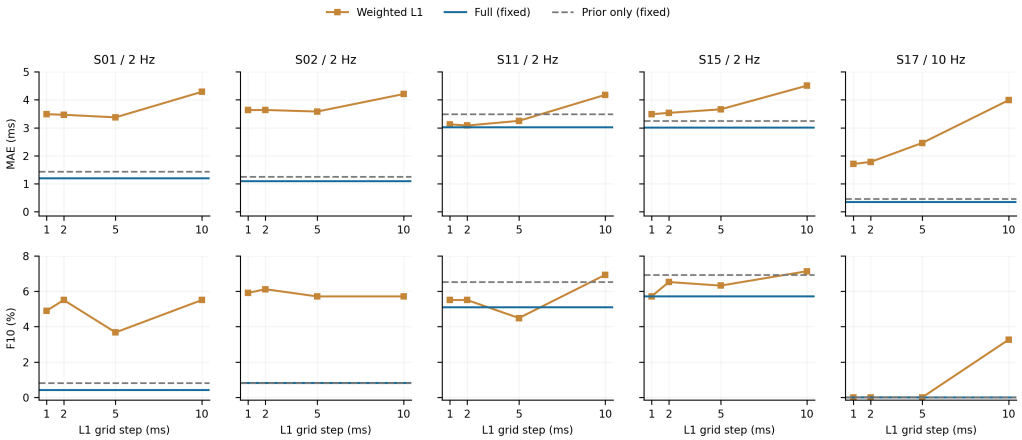

Figure 5: Continuous-offset recovery as weighted L1 search spacing varies. Columns show five sequences; rows show MAE and F_{10}. Full and Prior only provide fixed reference results. Points reproduce the aggregate continuous-injection results under confidence selection. S01/S02 and S11/S15 use 2 Hz inputs; S17 uses 10 Hz. Each point retains 490 of 700 comparison windows, with fixed selection across search resolutions.

The table below reports the complete off-grid sweep within ± 200 ms. Bold marks the minimum for each metric in a row, including ties. On S01 and S02, finer L1 spacing does not monotonically reduce MAE; the aggregate results do not establish the cause.

Table 12: Continuous injections: MAE (ms) / F_{10} (%) under confidence selection.

Point	Full	Prior only	L1, 10 ms	L1, 5 ms	L1, 2 ms	L1, 1 ms
S01	**1.187 / 0.4**	1.432 / 0.8	4.288 / 5.5	3.373 / 3.7	3.466 / 5.5	3.487 / 4.9
S02	**1.093 / 0.8**	1.253 / **0.8**	4.209 / 5.7	3.581 / 5.7	3.638 / 6.1	3.634 / 5.9
S11	**3.019** / 5.1	3.487 / 6.5	4.179 / 6.9	3.249 / **4.5**	3.081 / 5.5	3.126 / 5.5
S15	**3.007 / 5.7**	3.247 / 6.9	4.508 / 7.1	3.660 / 6.3	3.534 / 6.5	3.485 / **5.7**
S17	**0.341 / 0.0**	0.457 / **0.0**	3.989 / 3.3	2.459 / **0.0**	1.778 / **0.0**	1.705 / **0.0**

In a recorded comparison run, L1 runtime rises from 199 s at 10 ms spacing to 1,879 s at 1 ms spacing, a factor of 9.44. Hardware specifications, timing boundaries, and repeated measurements are unavailable, so these timings do not establish a latency advantage for ZeroSync. In the separate discrete-injection experiment, 2 ms L1 achieves $F_{10} = 4.90\%$ on S15 and 3.47% on S11, compared with 6.12% and 3.88% for ZeroSync. These results use a different injection distribution from the continuous experiment.

B Ablation and Robustness Studies

B.1 Pooled Comparison with Internal Objectives

Table 13 extends the main pooled comparison to all five internal deterministic objectives. All methods use the same matched windows across 73 configurations, without confidence selection. The all-window population contains 25,900 windows; the non-erased subset contains 23,889 windows identified solely by the synthetic-erasure flag. Full has the lowest pooled MAE and F_{10} among these

methods in both populations. Non-erased windows can still contain weak or ambiguous motion, so this subset is distinct from the observable training population.

Table 13: Pooled internal comparisons without confidence selection. Each entry is MAE (ms) / F_{10} (%). The two columns use different populations; methods within each column use matched windows.

Method	All windows	Non-erased windows
Full	**8.581 / 9.108**	**1.208 / 1.771**
Prior only	8.704 / 9.236	1.341 / 1.909
Weighted L1	9.447 / 17.166	2.147 / 10.507
Huber	9.447 / 17.189	2.146 / 10.532
Unweighted L1	9.593 / 17.514	2.305 / 10.884
L2	9.876 / 18.653	2.612 / 12.119
Correlation	10.400 / 18.243	3.180 / 11.675

B.2 Comparison with a pointwise learned scorer

We compare the residual CNN with a separately trained pointwise scorer that uses two 1×1 convolutions. The pointwise scorer mixes channels at each candidate offset without combining neighboring offsets. Both models use the same correlation input, prior, loss, candidate grid, sub-grid readout, data split, and training seed. This comparison tests a simpler scoring architecture in addition to the residual-off comparison.

Table 14: Learned scoring architectures on the canonical test and generalization splits. All entries are MAE (ms). These evaluations are separate from the 73-configuration sweep.

Split	Coverage	Full CNN	Pointwise scorer
Test	All windows	**11.886**	12.098
	70%	**1.727**	1.942
Generalization	All windows	**11.744**	12.330
	70%	**1.327**	1.966

Full reduces all-window MAE from 12.098 to 11.886 ms on the test split and from 12.330 to 11.744 ms on the generalization split. The reported MAEs at 70% coverage also favor Full. Because retained-window identities are unavailable in the comparison summary, we base the architecture comparison on the all-window results. These favor the CNN over this pointwise alternative, but do not isolate neighborhood size from parameter count. The comparison uses one training seed and does not include the complete 73-configuration sweep for the pointwise scorer.

B.3 Sequence-equal macro averages

To give each sequence equal weight, we first compute MAE and F_{10} within each configuration. We then average equally over the tested rates within each sequence and over sequences within each dataset. Table 15 reports all-window and non-erased results without confidence selection. EuRoC contributes 10 camera–IMU sequences; M2DGR and RELLIS-3D contribute 6 urban and 5 off-road LiDAR–IMU sequences. The non-erased MAEs also appear in the sensor-pair block of Table 1. These means across rates differ from the 2 Hz medians reported in the main comparison.

Table 15: Sequence-equal macro MAE (ms) / F_{10} (%) across tested rates, without confidence selection. Bold denotes the lowest value for each metric within a row.

Dataset	Population	Full	Prior only	Weighted L1
EuRoC	All	**8.409 / 8.450**	8.503 / 8.464	9.191 / 16.157
	Non-erased	**1.094 / 1.107**	1.196 / 1.122	1.941 / 9.458
M2DGR	All	**8.849 / 10.587**	8.999 / 10.762	9.133 / 15.984
	Non-erased	**1.570 / 3.423**	1.733 / 3.612	1.878 / 9.271
RELLIS-3D	All	**8.610 / 9.086**	8.776 / 9.467	10.390 / 21.429
	Non-erased	**1.132 / 1.672**	1.312 / 2.085	3.062 / 15.055

Full has lower macro MAE and F_{10} than both comparators in every dataset and population. These descriptive averages give each sequence equal weight; they do not quantify uncertainty across training runs.

B.4 Prior-peak selection on cached S15 windows

We test whether the residual benefit persists under selection without learned confidence. This exploratory analysis uses 700 cached S15 windows at 2 Hz, including motion erasures. S15 was chosen because complete score surfaces were available. Full and Prior only predictions are recomputed from the cached scores without retraining or new network inference. Recomputed Full predictions differ from the originals by at most 0.010665 ms.

Both alternative selectors use correlation logits from Prior only, without CNN residuals or recovery-success labels. The top-two gap is the difference between the largest and second-largest logits divided by the logit range. The separated-peak rule excludes candidates within 20 ms of the main peak before finding the strongest competitor. Flat surfaces receive a score of zero; the smaller of the two branch scores is used for selection. The candidate grid spans ± 300 ms at 10 ms spacing. Ties are broken by ascending window index (`dataset_index`). Each rule retains 490 windows for both estimators, although different rules retain different subsets. The 20 ms exclusion was fixed for this audit without test-error tuning or independent validation-set selection.

Table 16: S15 at 70% coverage ($n = 490$ per selector). Lower is better; bold compares Full with Prior only within each selector, not across subsets. F_{10} is the percentage with absolute error ≥ 10 ms.

	MAE (ms)		F_{10} (%)	
Selector	Full	Prior only	Full	Prior only
Learned confidence	**2.976**	3.263	**6.122**	7.959
Prior top-two gap	**3.714**	4.144	**8.776**	10.204
Prior separated peak (20 ms)	**3.297**	3.413	**7.143**	7.551

Full has lower MAE and F_{10} under both prior-peak rules, with relative MAE reductions of 10.38% and 3.41%. The selectors still share our motion representation and correlation prior; this single-sequence result does not establish rankings across datasets or other coverage levels.

B.5 Selection, erasure, and residual ablations

The selector study changes the retained population while keeping estimators fixed. Full coverage and all four 70% selection rules yield pooled MAE ordered as Full $<$ Prior only $<$ weighted L1. The non-erased analysis retains 23,889 windows satisfying `synthetic_motion_erased=0`, without observability or confidence filtering. Full and weighted L1 records are matched to Prior only records and erasure flags by source and window start time rounded to six decimal places. Table 1 reports the resulting pooled metrics.

Table 17: Pooled MAE and retained window counts across 73 configurations.

Selector	n	Full	Prior only	Weighted L1
none (100% of windows)	25900	**8.581**	8.704	9.447
model confidence	18130	**1.008**	1.145	2.025
motion excitation (model-free)	18130	**8.355**	8.470	9.230
input quality (model-free)	18130	**8.275**	8.413	9.243
random control	18130	**8.726**	8.851	9.584

Table 18: Residual ablation with model parameters fixed: MAE by observability population. Percentage reductions use the displayed values and may differ slightly from values computed before rounding.

Test condition	No residual	Full model	Reduction
A. Observable windows			
S15: standard	0.506	**0.399**	21.1%
S15: identifiable	0.503	**0.397**	21.1%
S03 + S11	2.142	**1.926**	10.1%
MS^2: transfer	2.059	**1.418**	31.1%
B. All windows			
S15: standard	10.646	**10.550**	0.9%
S15: identifiable	0.503	**0.397**	21.1%
S03 + S11	12.081	**11.886**	1.6%
MS^2: transfer	12.318	**11.744**	4.7%

The MS^2 transfer experiment uses cached camera–IMU motion from sequence 2021-08-06-10-59-33, with 5,600 sampled windows, including 5,013 observable windows. It does not evaluate LiDAR, thermal, or near-infrared inputs. The extraction configuration specifies RGB, but documentation of the cached image stream is incomplete. The standard S15 and combined S03+S11 experiments each contain 5,600 windows, including 5,053 observable windows. The identifiable S15 set contains 5,600 observable windows. Full and Prior only use identical network parameters and differ only in residual scale. Sampled windows may overlap.

In the selector study, 1,944 erased windows across 72 configurations have a reported MAE of 95.926 ms; the remaining 67 have an MAE of 103.284 ms. The rounded weighted mean is 96.171 ms over 2,011 windows. Thus 7.764% of all 25,900 windows are erased, contributing approximately 7.47 ms to the overall MAE of each estimator. Equal errors are observed for these estimators under these corruptions; they do not imply that every estimator must return zero.

Table 19: Separate 350-window S15 set: all-window MAE, confidence-selected MAE, and F_{10}.

Method	All MAE (ms)	Ret. MAE (ms)	F_{10} (%)
Correlation	17.629	8.653	28.98
L2	15.143	7.224	34.29
Huber	13.686	5.714	25.71
Weighted L1	**10.086**	**2.449**	14.69
Unweighted L1	11.514	3.633	20.00
ZeroSync	10.822	2.952	**5.71**

Of these 350 windows, 27 are erased, with MAE 95.926 ms for all six methods. They contribute 7.400 ms, or 68.4% of the 10.822 ms all-window MAE of ZeroSync. This population is separate

from the 700-window comparison and pooled selector study. The residual audit records a change in the selected peak in 25.6% of cases, a median prior top-two gap of approximately 0.001 logits, and a median absolute residual of 1.72. Residual magnitude alone does not determine whether the selected peak changes.

B.6 Shared residual delays and anchor removal

Let a be the residual native delay shared by both branches after precompensation. Writing $f_\theta(x) = a + e_0$ and $f_\theta(x_\Delta) = a + \Delta + e_1$ gives $\widehat{\Delta} - \Delta = e_1 - e_0$ before output clipping. The shared additive term cancels, but the estimation errors may depend on a. Recovery of the injected translation therefore does not establish absolute timing accuracy for either branch.

Figure 6 reports MAE and its change from the zero-delay condition. MAE varies little on S01, S02, and S17 over the tested range, but rises by 1.135 ms on S15 at $+50$ ms. S11 is more affected at -50 ms than at positive delays. A shared delay cancels algebraically in the target difference, but its effect on estimation error need not cancel.

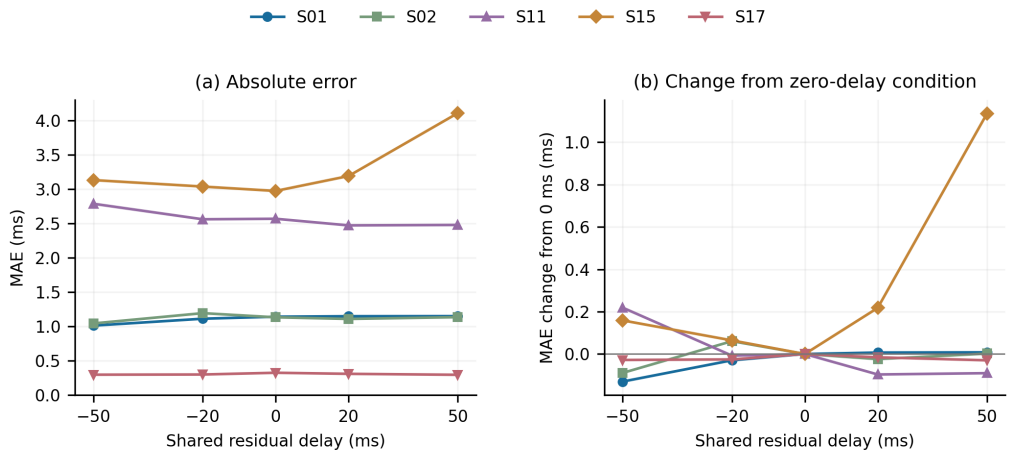

Figure 6: Shared-delay sensitivity of the paired pipeline. (a) Confidence-selected MAE. (b) Change from zero-delay MAE. Confidence ranking is recomputed at each delay; curves therefore include changes in retained membership. Lines connect tested conditions. The S17 -50 ms condition additionally has a physical-window mismatch, described below.

Table 20: Residual delay shared by both branches: MAE (ms) after confidence reselection at each condition. Zero-delay values match the matched-grid reference.

Point	0 ms	+20 ms	−20 ms	+50 ms	−50 ms
S01	1.142	1.149	1.112	1.150	1.012
S02	1.133	1.109	1.194	1.135	1.044
S11	2.570	2.474	2.563	2.480	2.789
S15	2.976	3.193	3.040	4.111	3.134
S17	0.324	0.309	0.299	0.295	0.296

At each delay, the top 490 of 700 windows are selected again. Between 474 and 490 windows overlap with the zero-delay selection; S15 at $+50$ ms retains 474. At this setting, F_{10} rises from 6.1% to 11.6%, and MAE rises by 38.1%. After subtracting the applied delay from window start times, all 700 S17 windows at -50 ms remain mismatched, with a maximum difference of 24.214 ms. Other audited conditions match within numerical precision. The curves therefore describe

the full selection procedure under perturbation, rather than an effect measured on fixed physical windows.

Table 21: Auxiliary anchor comparison under matched architecture, data, and schedule. MAEs are in ms.

Quantity	Anchored	No anchor
Training time (h)	10.24	10.18
Initial validation MAE	11.2845	11.2845
Best validation MAE	11.1916	11.1915
Best epoch	50	51
S15, $+50$ ms: selected MAE	4.111	4.109

The recorded 25-condition summary gives a signed mean change (no anchor minus anchored) of $+0.001$ ms and a largest increase of $+0.042$ ms. Anchoring reduces MAE in 15 conditions and increases it in 10; on S15 at $+50$ ms, anchored and no-anchor MAEs are 4.111 and 4.109 ms. The small average change does not establish equivalence between the two variants or explain the remaining S15 degradation.

C Paired error patterns and external implementation audit

The following tables use the 490 retained windows at each operating point. Column A counts windows where ZeroSync fails and the comparator succeeds; column B counts the reverse. Reported p values rounded to 0.000 are neither treated as exact zeros nor used to claim significance, because the analysis does not account for overlapping windows.

Table 22: Paired discordant counts at the 10 ms threshold.

Point	Rival	A	B
S15	Unweighted L1	20	82
S15	Huber	23	132
S15	Correlation	18	126
S15	Basalt	17	133
S11	Unweighted L1	14	56
S11	Huber	16	83
S11	Correlation	15	83
S11	Basalt	15	145
S01	Unweighted L1	2	61
S01	Huber	3	55
S01	Correlation	3	111
S01	Basalt	3	154
S02	Unweighted L1	3	51
S02	Huber	4	60
S02	Correlation	3	92
S02	Basalt	2	164
S17	Unweighted L1	0	41
S17	Huber	0	48
S17	Correlation	0	42
S17	Basalt	0	127
S15	Weighted L1	27	64
S11	Weighted L1	16	39

Table 23: Error-series correlation, counts of windows with an error advantage exceeding 5 ms, and MAE difference (model minus rival, ms).

Point	Rival	Correlation	Model better	Rival better	Δ MAE
S15	Unweighted L1	0.275	123	46	-0.70
S15	Huber	0.100	177	43	-2.74
S15	Correlation	0.253	174	44	-5.64
S15	Basalt	0.135	227	33	-3.29
S11	Unweighted L1	0.075	107	53	-0.21
S11	Huber	0.046	126	49	-0.92
S11	Correlation	0.133	137	46	-1.82
S11	Basalt	-0.011	236	36	-3.55
S01	Unweighted L1	0.125	109	11	-1.31
S01	Huber	0.146	113	11	-1.41
S01	Correlation	0.157	175	8	-3.23
S01	Basalt	0.078	257	7	-4.53
S02	Unweighted L1	0.215	110	9	-1.40
S02	Huber	0.058	124	10	-1.54
S02	Correlation	0.229	159	11	-2.70
S02	Basalt	0.167	285	5	-5.60
S17	Unweighted L1	0.594	94	0	-1.59
S17	Huber	0.650	96	0	-1.63
S17	Correlation	0.397	82	0	-1.35
S17	Basalt	0.597	253	0	-4.84

The reimplemented Basalt objective matches the reported upstream results at the tested operating points. This checks that component only, rather than every baseline implementation. Twist-n-Sync uses two calls per pair, with a zero-change fallback for non-finite outputs. Removing all retained zero-change estimates is a sensitivity analysis because it also excludes valid zero estimates.

Table 24: Exception audit: 1,400 calls and 700 pairs per point; 134 exceptions correspond to 9.6%.

Point	Exceptions	Affected pairs	Selected	F_{10} all / filtered
S01	134	78 / 700	10	60.82 / 62.08
S02	134	71 / 700	4	61.43 / 61.93
S11	134	75 / 700	7	67.14 / 68.12
S15	134	78 / 700	10	67.14 / 68.54
S17	134	70 / 700	2	51.02 / 51.23

D Motion-front-end diagnostics

D.1 Observation availability and temporal detail

Figure 7 shows motion extraction in two separate 12 s excerpts. The camera excerpt contains 69/240, 63/120, 52/60, and 24/25 finite rotation estimates at 20, 10, 5, and 2 Hz. The corresponding fractions rise from approximately 28.8% to 52.5%, 86.7%, and 96.0%, although the number of finite estimates falls from 69 to 24. A higher valid fraction therefore need not provide denser temporal evidence, and finite estimates may still be inaccurate.

In the LiDAR excerpt, the IMU and 10 Hz traces contain a brief angular-speed dip that the 2 Hz samples do not resolve. Interpolation cannot reconstruct a feature absent from the samples. These excerpts illustrate the distinction between observation availability and temporal detail; they do not measure learned alignment or sequence-wide sampling effects.

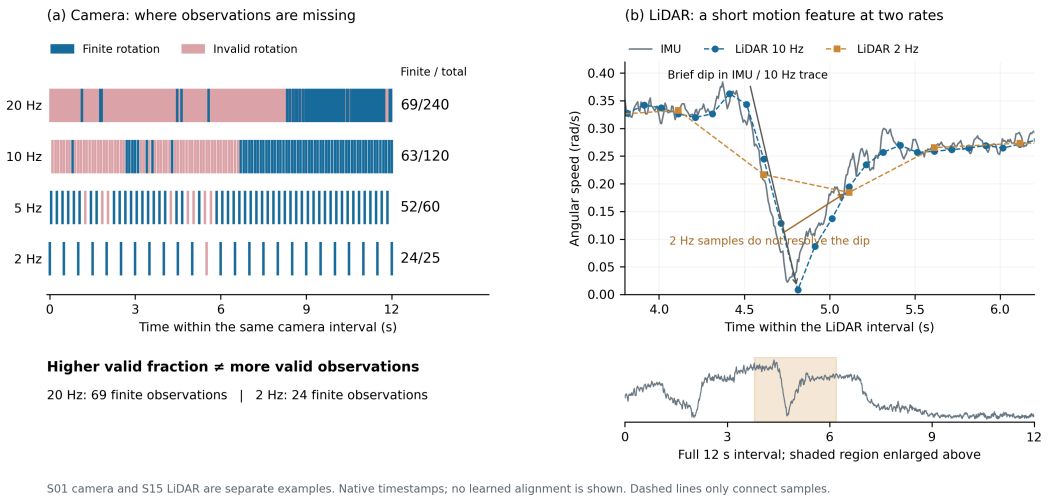

Figure 7: Local motion-front-end examples from S01 camera and S15 LiDAR. (a) Locations and counts of finite versus invalid camera rotation estimates in the displayed interval. (b) A short angular-speed feature at two LiDAR rates, with the full interval below for context. Native timestamps are retained; dashed lines only connect samples. These local counts are distinct from the aggregate camera fractions reported below. Exact recording start times and extraction settings are needed to reproduce the excerpts.

D.2 Aggregate front-end statistics

The following fractions measure the availability of valid camera-rotation estimates, not synchronization accuracy. Longer observation intervals can improve rotation recovery while reducing temporal sampling density. The estimated native delay also depends on the sampling rate and extracted motion representation.

Table 25: Valid camera-rotation fractions.

Sequence	20 Hz	10 Hz	5 Hz	2 Hz
S01	38%	65%	82%	86%
S02	44%	75%	88%	91%
S03	59%	78%	84%	88%
S04	46%	67%	80%	84%
S05	43%	80%	92%	94%
S06	80%	92%	92%	90%
S07	68%	85%	90%	84%
S08	39%	79%	90%	92%
S09	75%	90%	94%	94%
S10	71%	89%	92%	90%

Table 26: Estimated native delay (ms) at different acquisition rates.

Source	10 Hz	5 Hz	2 Hz
S15	-40.0	-40.0	+18.0
S11	-40.0	-40.0	-28.0
S13	-42.0	-40.0	-38.0
S17	+58.0	+66.0	+72.0

The evaluation excludes 1 Hz configurations and near-boundary native-delay cases; the rotation_02 estimate is 298 ms against a ± 300 ms search bound. A motion-reconstruction check gives maximum error 2.4×10^{-7} against a 10^{-4} tolerance.