# FlashCap: Millisecond-Accurate Human Motion Capture via Flashing LEDs and Event-Based Vision

## Supplementary Material

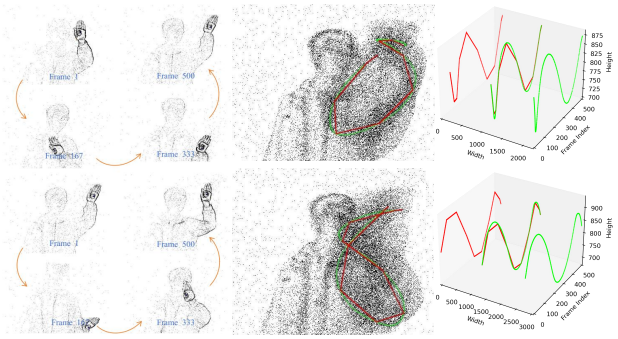

Figure 1. **Qualitative Comparison of Interpolation vs. 1000Hz Ground Truth.** We visualize the trajectory of rapid hand-waving (Sequence 1 & 2). The Green line represents the 1000 Hz ground truth, capturing fine-grained micro-dynamics. The Red line represents the trajectory interpolated from 20 Hz labels, which fails to capture rapid direction changes.

## A. Why are 1000 Hz labels required?

While millisecond-level accuracy is known to be critical for Olympic athletes (see the Introduction section), its relevance for the general population requires validation. We therefore benchmark standard interpolation techniques for upsampling low-frequency motion data of ordinary individuals. Our analysis first demonstrates the qualitative and quantitative failure of upsampling from 20 Hz data, and then quantitatively shows that even 100 Hz sources are insufficient for this purpose.

### A.1. Evaluation of Standard 20 Hz Video

In the main text, we highlighted the trajectory deviation when interpolating from 20 Hz labels. Here, the 20 Hz labels are obtained by downsampling our 1000 Hz ground truth, and interpolation is then applied to these downsampled ground-truth labels (not model predictions). As qualitatively visualized in Fig. 1, the interpolated trajectory (Red) smoothens out critical high-frequency details captured by our ground truth (Green). Below, we provide the quantitative breakdown.

**Failure of Linear Interpolation.** We first utilized linear interpolation on these 500-ms sequences. Tab. 1 compares the upsampled data against our ground truth, revealing high MPJPE and extremely low PCK scores.

**Comparison with Spline Interpolation.** Is advanced interpolation sufficient? We tested Spline interpolation on the same sequences. As shown in Tab. 2, while Spline interpolation reduces the mean error compared to Linear methods, the precision (PCK) remains unacceptable for precise analysis.

Table 1. **Quantitative Evaluation of Linear Interpolation (20Hz → 1000Hz).** Results on two rapid hand-waving sequences. **Metric Definitions:** MPJPE: Mean Per Joint Position Error; **ME**: Max Error; **PCK@t**: Proportion of Correct Keypoints within threshold $t$. (Errors are in pixels).

| Sequence | MPJPE↓ | ME↓ | PCK@0.3↑ | PCK@0.5↑ |
|---|---|---|---|---|
| Sequence 1 | 8.76 | 21.63 | 0.03 | 0.03 |
| Sequence 2 | 9.94 | 26.40 | 0.03 | 0.03 |

Table 2. **Linear vs. Spline Interpolation (20Hz → 1000Hz) on Hand-Waving Sequences.** While Spline interpolation reduces the MPJPE compared to Linear methods, the precision (PCK) remains low. *Metrics (MPJPE, ME) are defined in Tab. 1.*

| Seq | Method | MPJPE↓ | ME↓ | PCK@1↑ | PCK@2↑ |
|---|---|---|---|---|---|
| Seq 1 | Linear | 8.76 | 21.63 | 0.03 | 0.09 |
| | Spline | **2.98** | **7.07** | 0.01 | 0.15 |
| Seq 2 | Linear | 9.94 | 26.40 | 0.03 | 0.09 |
| | Spline | **3.80** | **12.04** | 0.01 | 0.19 |

Table 3. **Evaluation of Spline Interpolation (20Hz → 1000Hz) across Action Categories.** Note the catastrophic Max Error (ME) in *Jumping* (**74.01 px**) and *Swinging Racket*, confirming that 20Hz sources cannot resolve fast dynamics. *Metrics are defined in Tab. 1.*

| Action | MPJPE↓ | ME↓ | PCK@1↑ | PCK@2↑ |
|---|---|---|---|---|
| Walking | 1.36 | 7.07 | 0.020 | 0.872 |
| Running | 2.06 | 12.04 | 0.062 | 0.619 |
| Crossing Hands | 1.42 | 3.16 | 0.070 | 0.695 |
| Kicking | 1.84 | 6.08 | 0.094 | 0.575 |
| Punching | 1.93 | 6.40 | 0.108 | 0.479 |
| Jumping | 7.80 | **74.01** | 0.054 | 0.329 |
| Rotating Hands | 2.99 | **19.00** | 0.111 | 0.527 |
| Waving | 3.04 | 11.18 | 0.084 | 0.439 |
| Swinging Racket | 3.70 | **31.62** | 0.023 | 0.252 |

**Catastrophic Errors in Dynamic Actions.** To assess the generality of this finding, we evaluated spline interpolation across the entire dataset. As Tab. 3 indicates, dynamic actions are particularly prone to substantial errors. A notable example is *Jumping*, which exhibits a Maximum Error (ME) of **74.01 pixels**—a deviation too large for accurate motion estimation. This result confirms that 20 Hz source data is fundamentally insufficient for capturing rapid dynamics, regardless of the interpolation technique employed.

Table 4. **Evaluation of Spline Interpolation with High-Speed Sources (100Hz → 1000Hz).** Even with 100Hz input, highly dynamic actions like *Jumping* and *Rotating Hands* maintain significant Max Errors (**28.50 px**, **11.20 px**), validating the need for FlashCap's native 1000Hz resolution. *Metrics are defined in Tab. 1.*

| Action | MPJPE↓ | ME↓ | PCK@1↑ | PCK@2↑ |
|---|---|---|---|---|
| Walking | 0.85 | 4.12 | 0.245 | 0.910 |
| Running | 1.25 | 7.84 | 0.186 | 0.755 |
| Crossing Hands | 0.92 | 2.45 | 0.210 | 0.820 |
| Kicking | 1.15 | 3.95 | 0.195 | 0.710 |
| Punching | 1.10 | 4.15 | 0.205 | 0.685 |
| Jumping | 3.50 | **28.50** | 0.092 | 0.450 |
| Rotating Hands | 1.65 | **11.20** | 0.155 | 0.650 |
| Waving | 1.75 | 6.80 | 0.140 | 0.580 |
| Swinging Racket | 2.10 | **15.50** | 0.085 | 0.405 |

Table 5. Quantitative validation of PMT error for manually labeled high-speed RGB baselines . Errors are in milliseconds (ms).

| Baseline System | Kicking (ms) ↓ | Punching (ms) ↓ | Jumping (ms) ↓ |
|---|---|---|---|
| High-Speed RGB (100FPS) | $5.6 \pm 3.01$ | $5.25 \pm 2.38$ | $5.92 \pm 2.66$ |
| High-Speed RGB (200FPS) | $2.33 \pm 1.16$ | $2.75 \pm 1.71$ | $2.17 \pm 1.44$ |
| **FlashMotion (1000Hz) [Ours]** | **< 1ms (by design)** | | |

## A.2. Evaluation of High-Speed 100 Hz Sources

A critical question remains: would using a high-speed RGB camera (e.g., 100 Hz) suffice? To investigate this, we downsampled our 1000 Hz ground truth to 100 Hz and then upsampled it back using spline interpolation.

As shown in Tab. 4, despite the 5× increase in sampling rate, significant deviations persist in highly dynamic actions:

- **Jumping:** Exhibits a substantial Max Error (ME) of **28.50 pixels**.
- **Rotating Hands:** Maintains a high ME of **11.20 pixels**.
- **Swinging Racket:** Shows an ME of **15.50 pixels**.

These quantitative results validate that even 100 Hz sources cannot fully resolve the micro-dynamics (such as impact tremors and rapid reversals). This strictly justifies the need for FlashCap's native 1000 Hz resolution.

## A.3. Limitations of High-Speed RGB for PMT

Precise Motion Timing (PMT) in professional sports, such as the Olympic Games, typically requires specialized hardware and good lighting conditions. For example, the timing for a 100-meter race requires starting blocks equipped with force sensors, along with high-speed cameras and infrared light curtains. This necessitates a complicated setup and precise synchronization. Our goal is to achieve comparable precision using accessible vision sensors.

To benchmark against traditional vision solutions, we collected human motion sequences using an industrial camera at 100 FPS and 200 FPS. We manually annotated the specific timing frames to serve as the baseline. As shown in Tab. 5, even with 200 FPS input, the timing error remains significant. In contrast, our event-based solution offers millisecond-level resolution by design, validating its advantage for fine-grained temporal analysis.

# B. FlashCap System Implementation Details

In this section, we detail the hardware specifications and algorithmic formulations that enable FlashCap to achieve millisecond-level precision.

## B.1. Active Marker Configuration

Table 6. LED configuration, On-time ($t_p$), Off-time ($t_n$), and Flicker Period ($T = t_p + t_n$) in microseconds ($\mu s$).

| LED Location | Description | $t_p$ | $t_n$ | $T = t_p + t_n$ |
|---|---|---|---|---|
| LF | Left Foot | 250 | 200 | 450 |
| RDL | Right Lower Leg | 100 | 150 | 250 |
| LUL | Left Upper Leg | 100 | 200 | 300 |
| RH | Right Hand | 100 | 250 | 350 |
| LFA | Left Forearm | 100 | 300 | 400 |
| RUA | Right Upper Arm | 150 | 100 | 250 |
| LS | Left Shoulder | 150 | 150 | 300 |
| HIP | Hip | 150 | 200 | 350 |
| Neck | Neck | 150 | 250 | 400 |
| Head | Head | 150 | 300 | 450 |
| RS | Right Shoulder | 200 | 100 | 300 |
| LUA | Left Upper Arm | 200 | 150 | 350 |
| RFA | Right Forearm | 200 | 200 | 400 |
| LH | Left Hand | 200 | 250 | 450 |
| RUL | Right Upper Leg | 200 | 300 | 500 |
| LDL | Left Lower Leg | 250 | 100 | 350 |
| RF | Right Foot | 250 | 150 | 400 |

Each LED is positioned at a distinct keypoint on the human body. Functioning as an active marker, it emits flashing light at a configurable frequency, generating a stream of events that is easily detected by the event camera.

**Frequency Configuration.** To facilitate unique identification, we configure each LED with a distinct flicker period $T$, which consists of a specific on-time ($t_p$) and off-time ($t_n$). Specifically, the flicker period $T$ is configured to values ranging from 250$\mu$s to 500$\mu$s, with individual $t_p$ and $t_n$ settings between 100$\mu$s and 300$\mu$s. Crucially, since the maximum flicker period ($500\mu s$) is strictly smaller than our annotation time window (1 ms), this design guarantees that LED events are present within every single event frame. The detailed setup is provided in Tab. 6.

**Wavelength Selection.** Our selection is driven by both theoretical and empirical evidence. Theoretically, green light has the highest Quantum Efficiency (QE) among visible wavelengths (see Tab. 7), implying superior detectability. Empirically, we installed Green, Blue, and Red LEDs into the FlashCap joints for comparative testing. Results confirm that Green LEDs are significantly easier to detect and track than Red or Blue variants.

Table 7. **Quantum Efficiency (QE) of LEDs.** Comparison of peak wavelengths and sensor efficiency. Green offers the highest detectability.

| Color | Range (nm) | Peak (nm) | QE (%) |
|---|---|---|---|
| Green | 495–570 | 530 | **98** |
| Blue | 450–495 | 475 | 90 |
| Red | 620–750 | 650 | 91 |

## B.2. Multi-Sensor Synchronization

We employ a hybrid synchronization strategy consisting of hardware synchronization and software synchronization.

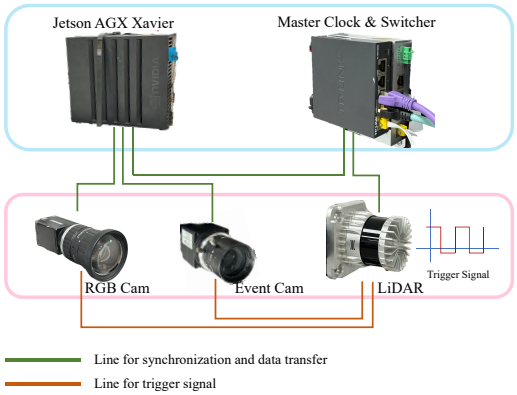


Figure 2. **Hardware Synchronization Schematic.** The visual sensors and LiDAR are synchronized via an **Auto66 Master Clock** and integrated with an **NVIDIA Jetson AGX Xavier** for high-throughput data capture.

**Hardware Synchronization (Visual & LiDAR).** As illustrated in Fig. 2, we implemented a robust hardware triggering system. The RGB Camera, Event Camera, and LiDAR are connected to an *Auto66* master clock. To ensure high-bandwidth data throughput and synchronization, these sensors interface with an NVIDIA Jetson AGX Xavier via dedicated switches and PCIe network cards. This architecture ensures that all optical frames and point clouds are stamped with a unified microsecond-level time base.

**Software Synchronization (IMU).** Since the Xsens IMU system operates on an internal clock without external hardware trigger support, we employ a trajectory-based software alignment. Before data collection, participants perform a high-velocity vertical arm swing. We developed a peak-detection algorithm to identify the apex of the arm trajectory in both the IMU data and the visual data. The temporal offset is then calculated by aligning these kinematic peaks, compensating for the constant system latency.

## B.3. Spatial Calibration Strategy

**Event-RGB Alignment.** Achieving pixel-perfect alignment between asynchronous events and synchronous RGB frames is non-trivial. We utilize a **Beam Splitter** rig to ensure both sensors share the exact same optical center. To correct for minor mechanical misalignments (rotation/translation) caused by the mounting interface, we reconstruct grayscale images from events using E2VID and compute a homography matrix to spatially register them against the RGB frames.

**LiDAR-RGB Alignment.** For 3D spatial calibration, we first separate the human point cloud from the raw LiDAR data. To ensure robust alignment, we utilize a coarse calibration matrix (derived from IMU-LiDAR extrinsic) as initialization. Subsequently, we manually annotate corresponding keypoints on the human body in both the dense LiDAR point cloud and the RGB image to solve the Perspective-n-Point (PnP) problem. This pipeline ensures precise transformation from the LiDAR coordinate system to the RGB camera frame.

## B.4. Tracking Algorithm Formulation

To maintain temporal consistency and recover clusters missed due to transient noise, we employ a bipartite matching tracker. For each event frame $t$, we explicitly denote the set of detected clusters as $\mathcal{C}_t = \{c_t^1, c_t^2, \dots, c_t^n\}$, and for the previous frame $t-1$, as $\mathcal{C}_{t-1} = \{c_{t-1}^1, c_{t-1}^2, \dots, c_{t-1}^m\}$.

We compute the cost matrix $D_{ij}$ as the Euclidean distance between the centroids of all cluster pairs $(c_{t-1}^i, c_t^j)$:

$$D_{ij} = \left\| \boldsymbol{\mu}_t^j - \boldsymbol{\mu}_{t-1}^i \right\|_2 \tag{B.1}$$

where $\boldsymbol{\mu}_t^j$ and $\boldsymbol{\mu}_{t-1}^i$ denote the spatial centroids of clusters $c_t^j$ and $c_{t-1}^i$, respectively.

We then solve for the optimal association $\pi$ by minimizing the total transport cost:

$$\min_{\pi} \sum_{i} D_{i,\pi(i)}, \quad \text{s.t.} \quad D_{i,\pi(i)} < \tau \tag{B.2}$$

where $\tau$ is a spatial threshold used to reject unreliable matches.

**Occlusion Handling.** Crucially, if a cluster $c_{t-1}^i$ cannot be matched in the current frame but maintains spatial consistency across previous history, we retain it as a potential candidate. This "memory" mechanism bridges gaps when an LED is momentarily occluded or fails to trigger sufficient events, ensuring stable ID tracking over time.

# C. System Robustness and Reliability Evaluation

Beyond hardware setup, we rigorously validated the system's reliability under physical stress and environmental interference.

Table 8. **Mechanical Stability Test Results.** Marker displacement across 4 volunteers performing intense actions for 1 minute. The low overall deviation (**1.6 mm**) confirms the effectiveness of our high-friction mounting strategy. (Unit: mm).

| Action | Avg. Dev. | Max Dev. | Min Dev. |
|---|---|---|---|
| Jumping | 2.0 | 3.0 | 1.0 |
| Punching | 1.3 | 2.0 | 1.0 |
| Kicking | 1.5 | 2.0 | 1.0 |
| **Overall** | **1.6** | 3.0 | 1.0 |

Table 9. **Drift Analysis on Joints (Elbow/Knee).** Placing markers directly on joints results in significantly higher deviation (**10.3 mm**) compared to the mid-segment placement (1.6 mm). This justifies our strategic decision to avoid joint locations. (Unit: mm).

| Action | Avg. Dev. | Max Dev. | Min Dev. |
|---|---|---|---|
| Jumping | 9.2 | 12.0 | 8.0 |
| Punching | 10.5 | 14.0 | 9.0 |
| Kicking | 11.1 | 13.5 | 8.5 |
| **Overall** | **10.3** | 14.0 | 8.0 |

### C.1. Mechanical Stability of Wearable Markers

A primary concern with wearable MoCap is the "drift" of markers relative to the skin. To address this, we integrated the FlashCap sensing unit into a custom 3D-printed transparent housing and secured it using elastic straps with high-friction backing, adopting the mounting protocol of **Xsens** [10].

**Stability Validation.** We conducted a physical stress test where 4 volunteers performed intense actions (Jumping, Punching) for 1 minute. By physically marking the strap's initial position and measuring absolute displacement post-exercise, we found the average drift to be minimal ($\leq$ 1.6 mm, see Tab. 8). Crucially, since most *FlashMotion* sequences are short ($<$ 1 minute), this measurement serves as a conservative upper bound, implying actual drift in the dataset is negligible.

**Placement Strategy.** We intentionally place markers on the "mid-segment" of limbs rather than directly on joints. As compared in Tab. 9, placing markers on joints (elbows/knees) results in significantly higher deviation ($\sim$10.3 mm) and restricts user movement. Our mid-segment placement ensures both data accuracy and user comfort.

### C.2. Robustness to Lighting and Occlusion

We stress-tested the system under adverse conditions to prove its "in-the-wild" capability.

**Lighting Conditions.** To evaluate robustness against variable illumination, we conducted experiments outdoors at three specific times: Noon, Afternoon, and Evening. As shown in Tab. 10, the primary challenge arises at Noon, where recall rates of markers dip to 93%–98%. This is attributed to intense sunlight acting as strong ambient interference; the high luminosity causes the LED emissions to blend into the background, making them difficult to isolate. In less strong ambient lighting (Afternoon and Evening), the system consistently maintains superior reliability ($> 98\%$).

*Abrupt light changes* (switching lights on/off, see Tab. 11), the recall of markers drop is negligible (0.23%), demonstrating the inherent advantage of differential event sensing.

**Occlusion Resilience.** Tab. 12 demonstrates that even when 3 LEDs are simultaneously occluded, the remaining markers are tracked with 99.83% precision. The system only begins to degrade gracefully when >50% (9 LEDs) are occluded, yet still maintains acceptable recall.

### C.3. Signal-to-Noise Analysis

To verify that our active markers stand out against background noise, we analyzed the event density. **Methodology.** For each action sequence, we sampled 10 1000-ms segments and computed the event distribution within a $5 \times 5$ pixel region centered on each LED centroid.

**Results.** As detailed in Tab. 15, events triggered by our LEDs constitute **92.61%** of all events within this localized region. This high Signal-to-Noise Ratio (SNR) confirms that the specific frequency modulation of our LEDs effectively cuts through environmental noise, ensuring robust tracking.

### C.4. Clustering Algorithm Benchmarking

The choice of clustering algorithm is pivotal for real-time performance. We benchmarked DBSCAN against HDBSCAN [4], STDBSCAN [2], and deep learning approaches like DMoN [9].

**Results.** As shown in Tab. 13, DBSCAN achieves the best balance of speed (3.086 iterations/s) and Precision (99.92%). While deep methods (DMoN) perform well, they are computationally heavier.

**Parameter Tuning.** We performed a hyperparameter sweep (Tab. 14) and identified $min\_samples = 15, eps = 1$ as the optimal operating point for our specific LED cluster density.

## D. FlashMotion Dataset Details

This section provides comprehensive details on the dataset composition, the optimization pipeline for 3D ground truth, and the rigorous quality assurance protocols employed during annotation.

### D.1. Dataset Composition and Action Categories

The *FlashMotion* dataset is designed to capture high-speed dynamics often missed by conventional MoCap. As listed in Tab. 16, we curated 11 major action categories encompassing

Table 10. **Robustness to Illumination.** Recall rates of markers for six distinct motions under varying illumination conditions. While intense sunlight at Noon slightly affects detection, the system remains robust with recall rates consistently exceeding 93%.

| Condition | Running | Punching | Kicking | Jumping | Rotating Hands | Crossing Hands |
|---|---|---|---|---|---|---|
| Noon | 0.9834±0.0008 | 0.9323±0.0029 | 0.9414±0.0012 | 0.9738±0.0010 | 0.9522±0.0015 | 0.9402±0.0015 |
| Afternoon | 0.9919±0.0006 | 0.9865±0.0006 | 0.9821±0.0007 | 0.9900±0.0005 | 0.9813±0.0007 | 0.9884±0.0006 |
| Evening | 0.9976±0.0000 | 0.9916±0.0094 | 0.9977±0.0003 | 0.9960±0.0002 | 0.9999±0.0004 | 0.9943±0.0004 |

Table 11. **Robustness to Abrupt Lighting Changes.** Comparison of the recall rate of markers under steady incandescent lighting versus dynamic switching conditions (5s intervals). The layout is aligned with Tab. 10 for consistency.

| Condition | Running | Punching | Kicking | Jumping | Rotating Hands | Crossing Hands |
|---|---|---|---|---|---|---|
| Steady Light | 0.9998±0.0000 | 0.9712±0.0009 | 0.9902±0.0001 | 0.9671±0.0007 | 0.9627±0.0001 | 0.9632±0.0003 |
| Abrupt Change | 0.9993±0.0003 | 0.9680±0.0013 | 0.9884±0.0004 | 0.9635±0.0010 | 0.9602±0.0004 | 0.9611±0.0006 |

Table 12. **Occlusion Resilience Analysis.** Performance metrics under varying degrees of LED occlusion. Note that *Recall (Vis.)* measures the detection rate of the remaining non-occluded markers, which remains robust (>99.6%) even when 9 LEDs are blocked.

| # Occluded | Recall (All) | Recall (Vis.) | Precision |
|---|---|---|---|
| 1 LED | 0.9402±0.0087 | 0.9989±0.0012 | 0.9998±0.0002 |
| 2 LEDs | 0.8793±0.0104 | 0.9965±0.0015 | 0.9999±0.0001 |
| 3 LEDs | 0.8203±0.0129 | 0.9961±0.0016 | 0.9983±0.0008 |
| 9 LEDs | 0.4704±0.0117 | 0.9991±0.0009 | 0.9997±0.0003 |

Table 13. **Clustering Benchmark.** DBSCAN offers the best trade-off between speed and precision. (FPS: Frames Per Second).

| Method | Speed (FPS) ↑ | Prec. ↑ | Recall ↑ |
|---|---|---|---|
| **DBSCAN (Ours)** | **3.086** | **0.9992** | 0.9627 |
| HDBSCAN [4] | 2.623 | 0.0631 | 0.7632 |
| STDBSCAN [2] | 0.078 | 0.6569 | 0.9701 |
| Spectral [6] | 0.483 | 0.0517 | 0.4110 |
| DMoN [9] | 0.9753 | 0.9250 | **0.9753** |

Table 14. **DBSCAN Hyperparameter Sweep.** Performance metrics (**Precision / Recall** of LEDs) under varying *eps* and *min_samples*. The configuration (*eps*=1, *min_samples*=15) yields the optimal balance.

| eps | min_samples | | |
|---|---|---|---|
| | 10 | 15 | 20 |
| **1** | 0.9437 / 0.9696 | **1.0000 / 0.9899** | 0.9641 / 0.9839 |
| **2** | 0.8988 / 0.9694 | 0.8664 / 0.9668 | 0.8916 / 0.9806 |
| **4** | 0.7698 / 0.9329 | 0.7870 / 0.9533 | 0.7258 / 0.9366 |

19 detailed motion types. These range from periodic motions (e.g., Walking, Running) to highly transient, explosive actions (e.g., Punching, Jumping, Javelin Throwing, Fencing), ensuring a diverse testbed for high-temporal-resolution algorithms.

Table 15. **Signal-to-Noise Analysis.** Event statistics within a $5 \times 5$ pixel region around marker centroids. The high Signal Ratio (≈92.6%) confirms that LED-triggered events dominate the data stream. (M = Million events).

| Action | Total Events (M) | LED Signal (M) | Signal Ratio ↑ |
|---|---|---|---|
| Walk | $1.78 \pm 0.04$ | $1.64 \pm 0.03$ | $0.922 \pm 0.004$ |
| Kick | $1.83 \pm 0.04$ | $1.70 \pm 0.03$ | $0.929 \pm 0.004$ |
| Run | $1.75 \pm 0.04$ | $1.61 \pm 0.04$ | $0.921 \pm 0.005$ |
| Punch | $1.78 \pm 0.04$ | $1.64 \pm 0.03$ | $0.924 \pm 0.004$ |
| Jump | $1.74 \pm 0.03$ | $1.60 \pm 0.03$ | $0.922 \pm 0.003$ |
| Cross Hands | $1.79 \pm 0.04$ | $1.65 \pm 0.04$ | $0.923 \pm 0.004$ |
| Rotate Hands | $1.77 \pm 0.04$ | $1.64 \pm 0.03$ | $0.927 \pm 0.003$ |
| Wave | $1.70 \pm 0.04$ | $1.58 \pm 0.03$ | $0.929 \pm 0.004$ |
| Swing Racket | $1.74 \pm 0.04$ | $1.63 \pm 0.03$ | $\mathbf{0.938 \pm 0.004}$ |
| **Overall** | $\mathbf{1.76 \pm 0.04}$ | $\mathbf{1.63 \pm 0.03}$ | $\mathbf{0.926 \pm 0.004}$ |

## D.2. Optimization of 3D Ground Truth

While our primary contribution is the 1000 Hz 2D labels, we also provide high-quality 60 Hz 3D pose parameters (SMPL) as a complementary modality. To mitigate the drift inherent in raw IMU data, we employed the optimization pipeline from RELI11D [12]. This pipeline integrates IMU orientation with LiDAR point clouds by minimizing a composite loss function. Specifically, it utilizes a contact-aware loss $\mathcal{L}_{contact}$ to constrain vertex penetration by reconstructing the scene mesh from the LiDAR background. A smoothness loss $\mathcal{L}_{smooth}$ ensures the temporal consistency of joint rotations, while a geometry loss $\mathcal{L}_{geo}$ minimizes the 3D Chamfer distance between visible SMPL vertices and the human point cloud. This LiDAR-guided optimization effectively reduces IMU drift, yielding geometrically consistent 3D supervision.

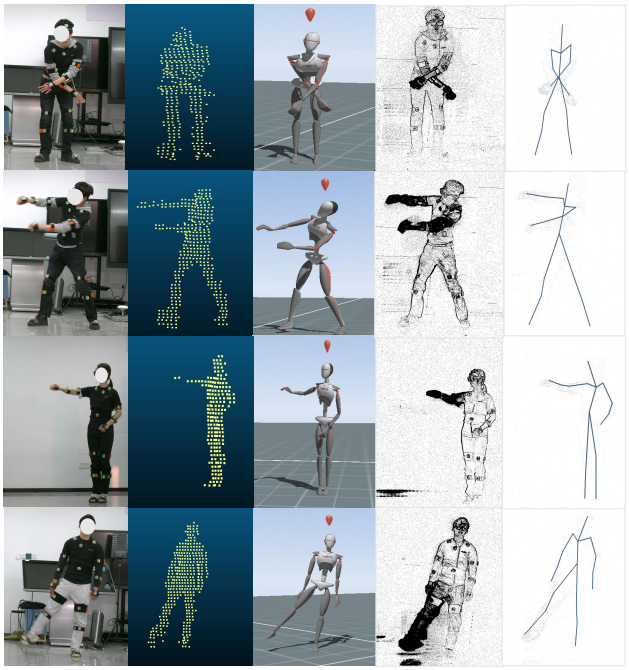


Figure 3. **Examples of FlashMotion.**

Table 16. **List of Action Categories.** The dataset comprises 11 major categories encompassing 19 detailed action types. Note that "Fencing" specifically captures the high-speed *Forward Lunge* motion.

| Major Category | Detailed Action Types (IDs) |
|---|---|
| **Walking** | 1. Walking |
| **Running** | 2. Sprinting, 3. High Knees |
| **Kicking** | 4. Snap Kick Left, 5. Snap Kick Right |
| **Punching** | 6. Punch Left, 7. Punch Right |
| **Jumping** | 8. Hop on Single Foot, 9. Star Jump, 10. Vertical Jump |
| **Swinging Racket** | 11. Forehand Smash |
| **Javelin** | 12. Run-up, 13. Throwing Action |
| **Waving Hands** | 14. Wave Left, 15. Wave Right, 16. Wave Both Hands |
| **Hand Interactions** | 17. Crossing Hands in Front of Body, 18. Rotating Hands in Front of the Body |
| **Fencing** | 19. Forward Lunge |

## D.3. Annotation Tool for Quality Assurance

To ensure label integrity, we developed a bespoke annotation tool (Fig. 4) acting as the final verification stage. The interface of the tool visualizes the output of our automated pipeline overlaid on RGB images. Human annotators utilize this tool to inspect complex scenarios—such as extreme occlusion during limb twisting—where spatial ambiguities might challenge the automated bipartite matching. The tool enables precise manual correction of 2D keypoints, ensuring the final labels meet high-quality standards.

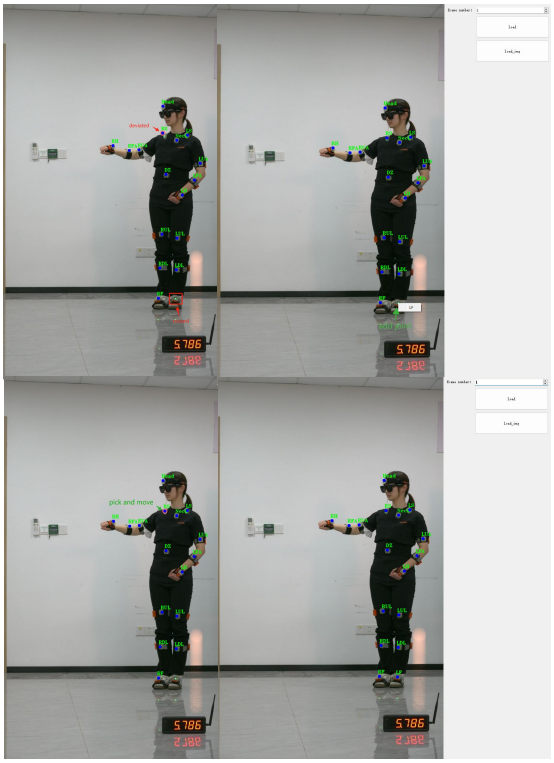

Figure 4. **Custom Annotation Interface.** The tool visualizes RGB images, enabling annotators to inspect and refine high-frequency labels with frame-by-frame precision.

## D.4. Quantitative Validation against Human Annotations

To rigorously quantify the reliability of our automated pipeline, we conducted a validation study against manual ground truth. **Setup.** We selected a diverse subset representing 10% of the dataset (24 sequences across 8 action categories). Human experts manually annotated these sequences frame-by-frame to serve as the reference Ground Truth. As presented in Tab. 17, our automated pipeline achieves an average Precision of 99.94% and a Recall of 99.11%. The high recall confirms the robustness of our LED frequency coding strategy. Missed detections typically occur when two markers become spatially indistinguishable (e.g., during tight limb crossing), which are subsequently flagged and corrected using the tool described in Sec. D.3.

Table 17. Quantitative evaluation against human annotated labels. Precision / Recall of LEDs for each action.

| Method | **Kicking** | **Crossing Hands** | **Jumping** | **Walking** | **Running** | **Punching** | **Waving** | **Swinging Racket** |
|---|---|---|---|---|---|---|---|---|
| **w/o $d_{ji}^{p}$** | 0.6970 / 0.9756 | 0.6400 / 0.9873 | 0.1650 / 0.9986 | 0.7320 / 1.0000 | 0.7337 / 0.9997 | 0.5643 / 0.9615 | 0.5598 / 0.9956 | 0.4963 / 0.9855 |
| **w/o Outlier Filter** | 0.9652 / 0.9569 | 0.8317 / 0.9850 | 0.8085 / 0.8480 | 0.8480 / 1.0000 | 0.8466 / 0.9996 | 0.8204 / 0.9513 | 0.8479 / 0.9952 | 0.8431 / 0.9845 |
| **w/o Tracking** | 0.9838 / 0.9816 | 0.8429 / 0.9821 | 0.7154 / 0.9692 | 0.8490 / 0.9997 | 0.8500 / 0.9998 | 0.8017 / 0.9424 | 0.8492 / 0.9924 | 0.8414 / 0.9818 |
| **FlashCap** | 0.9999 / 0.9899 | 0.9999 / 0.9865 | 1.0000 / 0.9981 | 0.9997 / 1.0000 | 0.9998 / 0.9998 | 0.9983 / 0.9712 | 0.9990 / 0.9972 | 0.9988 / 0.9857 |

# E. Experimental Settings and Additional Benchmarks

This section details the implementation specifics, training strategies for generalization, and additional benchmark evaluations that substantiate the efficiency and cost-effectiveness of our proposed method.

## E.1. Implementation Details

**Hardware & Setup.** Data annotation and evaluation were conducted on an Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz. For PMT and HPE tasks, we utilized an NVIDIA GeForce RTX 3090 GPU. All methods were implemented in PyTorch.

**Data Split.** To ensure rigorous evaluation, we split the dataset by subjects: data from 80% of the volunteers were used for training, and the remaining 20% were reserved for testing.

## E.2. ResPose Ablation Study

We validate the effectiveness of the key components in our ResPose architecture.

**SNN vs. ANN Encoder.** To justify the use of Spiking Neural Networks (SNN), we replaced the SNN-based event encoder with a standard CNN-based equivalent (ANN). As shown in Tab. 18, the SNN encoder achieves lower error (5.66 vs. 8.12). This performance gap arises because standard ANNs typically process integrated event frames, which inherently compresses temporal information and causes motion blur. In contrast, SNNs process the asynchronous event stream directly, effectively preserving the high-temporal-resolution micro-dynamics required for precise tracking. Moreover, SNNs are also more efficient than ANNs in our setting, as reflected by the runtime comparison in Tab. 22.

Table 18. Ablation: SNN vs. ANN Encoder. The SNN-based design yields higher precision by avoiding the temporal information loss inherent in frame-based ANN processing.

| Model Variant | MPJPE ↓ | PCK@0.5 ↑ |
|---|---|---|
| ANN-based Encoder | 8.12 | 0.96 |
| **SNN-based Encoder (Ours)** | **5.66** | **0.99** |

Table 19. **System Comparison.** We compare cost, bandwidth, and capabilities. Ours is the **only solution** enabling **automatic 1kHz GT acquisition** with low bandwidth across diverse illuminations.

| System | Cost (USD) | Illumination | Bandwidth | FPS | Annotation |
|---|---|---|---|---|---|
| High-Speed RGB | \$20k–\$700k | High Lux (> 2000) | 1–10 Gbps | 0.5k–2.5k | Manual (**Intensive**) |
| Optical MoCap (Vicon) | \$300k–\$800k | IR (Studio Only) | 1–5 Gbps | 120–330 | Auto |
| **LEDs + Event (Ours)** | **\$1k–\$6k** | **Robust (0 lux–Outdoor)** | **5–20 Mbps** | **1k** | **Auto** |
| *+ RGB Camera (opt.)* | \$0.1k–\$0.5k | Normal Lux | 50–200 Mbps | 20 | Auto |
| *+ LiDAR (opt.)* | \$0.6k–\$15k | All | 20–80 Mbps | 20 | Auto |
| *+ IMU (opt.)* | \$3k–\$10k | All | < 1 Mbps | 60 | Auto |

## E.3. Cost and Deployment Analysis

We clarify the cost structure by distinguishing between our acquisition system and the deployment system. The acquisition system is high-cost as it integrates LiDAR, Xsens IMUs, and master clocks; however, this setup is used solely for creating the dataset to ensure rigorous ground truth. In contrast, the deployment system for end-users is cost-effective, requiring only one standard RGB camera and one event camera, synchronized via a simple hardware trigger. This configuration costs a fraction (less than 10%) of professional high-speed cameras (e.g., NAC Memrecam, approx. \$45,000), fulfilling the goal of democratizing high-speed motion analysis.

To position FlashCap among existing acquisition systems, Tab. 19 compares cost, bandwidth, illumination robustness, and annotation scalability, showing that FlashCap provides a practical balance of affordability, deployment flexibility, and millisecond-level automatic annotation. Beyond professional scenarios such as the Olympic Games, FlashCap is also highly valuable for daily deployment in non-elite sports centers with limited budgets that cannot afford complex high-speed RGB setups. It is similarly practical for routine training in professional teams, where frequent, millisecond-level motion assessment is needed without the overhead of expensive laboratory-grade equipment.

Beyond professional scenarios such as the Olympic Games, FlashCap is also highly valuable for daily deployment in non-elite sports centers with limited budgets that cannot afford complex high-speed RGB setups. It is similarly practical for routine training in professional teams, where frequent, millisecond-level motion assessment is needed without the overhead of expensive laboratory-grade equipment.

## E.4. Additional Benchmarks and Efficiency Analysis

While the main paper comprehensively evaluates our proposed ResPose against multiple baselines, we provide further

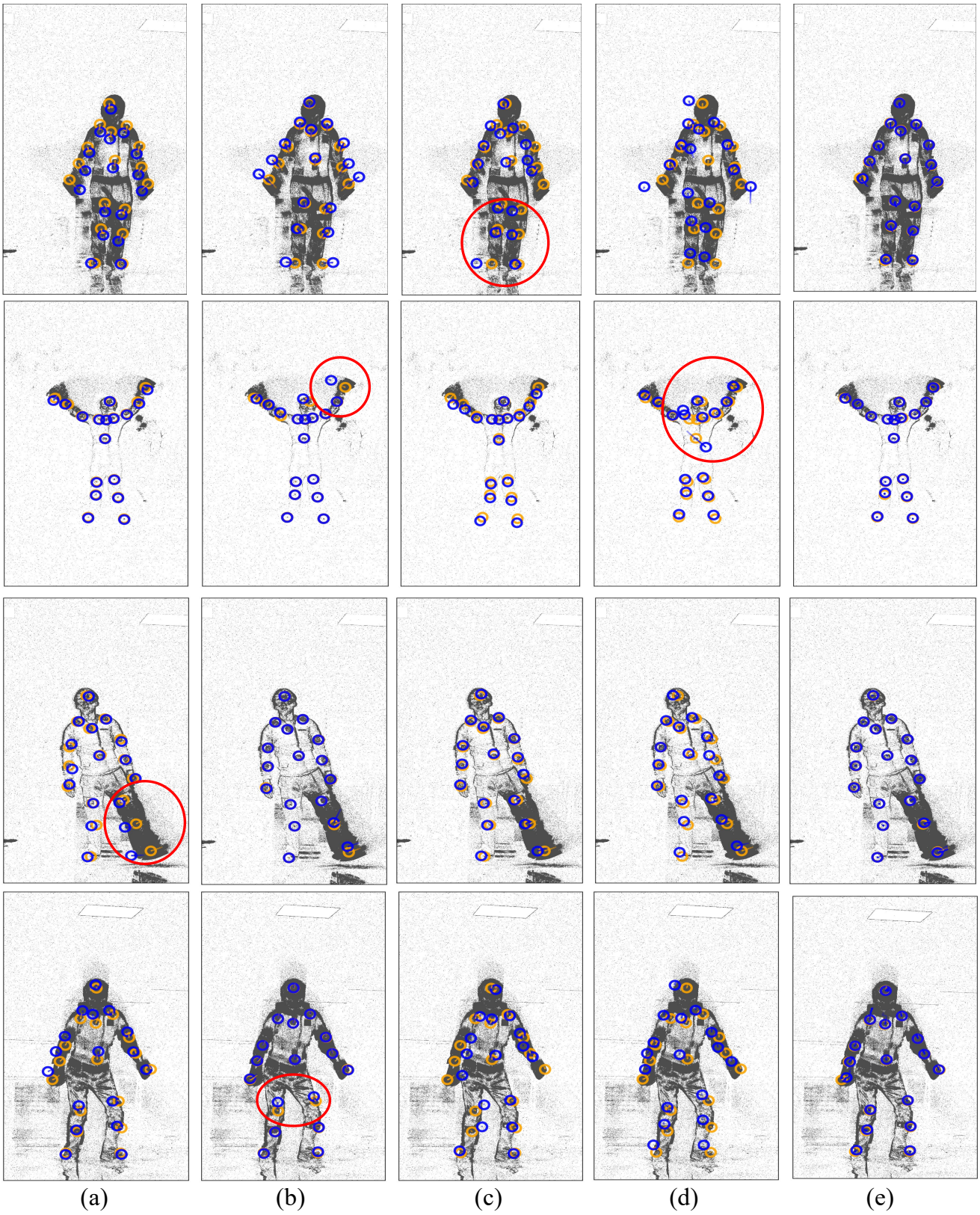


Figure 5. **Qualitative comparison of trajectories for the High Temporal Resolution HPE task over a 50ms interval. Orange indicates the Ground Truth (GT) and blue indicates the predicted pose. The columns correspond to: (a) ViTPose, (b) Hybrid ANN-SNN, (c) EventPointPose, (d) LEIR, and (e) ResPose (Ours).**

Table 20. The Mean Error of Estimated Time (PMT).Unit: $ms$.

| Method | Kicking | Punching | Jumping |
|---|---|---|---|
| ViTPose [11] | 48.5 | 62.3 | 31.4 |
| Hybrid ANN-SNN [1] | 85.2 | 54.1 | 66.7 |
| EventPointPose [5] | 98.3 | 118.7 | 80.4 |
| LEIR [12] | 112.4 | 135.8 | 78.2 |
| **ResPose (Ours)** | **7.2** | **4.8** | **6.5** |

Table 21. **Advanced Interpolation Strategies (HPE Task).** Comparison of linear vs. spline upsampling for applicable low-frame-rate baselines.

| Method | MPJPE↓ | PCK0.3↑ | PCK0.5↑ |
|---|---|---|---|
| ViTPose [11] (linear) | 10.06 | 0.96 | 0.98 |
| ViTPose [11] (spline) | 10.13 | 0.96 | 0.98 |
| GraphEnet [7] (linear) | 43.01 | 0.65 | 0.83 |
| GraphEnet [7] (spline) | 42.93 | 0.65 | 0.83 |
| EvSharp2Blur [8] (linear) | 8.78 | 0.95 | 0.96 |
| EvSharp2Blur [8] (spline) | 8.85 | 0.95 | 0.96 |
| **ResPose (Ours)** | **5.66** | **0.97** | **0.99** |

Table 22. **Runtime Comparison.** Inference efficiency comparison (Batch Size=32).

| Method | FPS↑ | Time (ms)↓ | Mem. (GB)↓ |
|---|---|---|---|
| ViTPose | 52.30 | 19.12 | 3.92 |
| Hybrid ANN-SNN | 243.33 | 4.11 | 7.04 |
| EventPointPose | 81.4* | 12.29* | – |
| LEIR | 88.29 | 12.11 | 3.14 |
| GraphEnet | 297.88 | 3.36 | 1.62 |
| EvSharp2Blur | 5.60 | 185.11 | 4.47 |
| ResPose (ANN Variant) | 156.82 | 6.37 | 3.42 |
| **ResPose (Ours)** | **324.00** | **3.08** | **1.60** |

* Reported by the original paper measured on Jetson Xavier NX, which is not directly comparable to our RTX 3090 hardware setup.

benchmarking details in this section to address the impact of upsampling strategies and runtime efficiency. For additional qualitative evidence, Fig. 5 visualizes the High-Temporal Resolution HPE trajectories over a 50ms interval across (a) ViTPose, (b) Hybrid ANN-SNN, (c) EventPointPose, (d) LEIR, and (e) ResPose (Ours).

**Advanced Interpolation Strategies (HPE Task):** In Sec 6.2 of the main text, we linearly upsampled the outputs of low-frame-rate baselines to 1000 Hz. To investigate whether advanced interpolation techniques could bridge the performance gap, we evaluate the High-Temporal Resolution HPE task using Spline interpolation for applicable baselines (e.g., ViTPose, GraphEnet, and EvSharp2Blur). As reported in Tab. 21, we observe a counter-intuitive phenomenon: utilizing a more advanced spline interpolation often yields *worse* results than simple linear interpolation . This occurs because low-frequency RGB methods inherently miss high-speed micro-dynamics and suffer from temporal uncertainties (e.g., motion blur). While spline interpolation attempts to fit a smooth continuous curve, it tends to over-fit to these sparse and noisy anchors, thereby exaggerating and amplifying the temporal errors (such as overshooting during rapid directional changes). In contrast, ResPose avoids this mathematical guessing by directly leveraging high-frequency event streams to capture actual micro-dynamics, consistently outperforming all interpolated variants.

**Runtime Efficiency Analysis:** A critical requirement for high-frequency motion capture is computational efficiency. We benchmarked the inference speed (Frames Per Second, FPS), latency (Time in ms), and GPU memory footprint (in GB) of all methods on a single NVIDIA RTX 3090 GPU (Batch Size = 32). As detailed in Tab. 22, ResPose achieves the best trade-off between precision and efficiency. Our SNN-Transformer hybrid architecture processes asynchronous events highly efficiently, operating at over 300 FPS while maintaining a lightweight memory footprint (approx. 1.60 GB), significantly outperforming heavy RGB-Event fusion frameworks.

**Additional PMT Results:** Furthermore, while the main paper evaluates four core methods for the Precise Motion Timing (PMT) task, we extend this evaluation to include EventPointPose [5]. As shown in Tab. 20, pure event-point-based methods struggle to provide accurate joint localizations at the exact crossing timestamps, yielding substantial millisecond-level timing errors.

# F. Discussion and Broader Impact

## F.1. Social Impact and Ethical Considerations

The *FlashMotion* dataset is designed to advance research in high-temporal-resolution human motion analysis, with potential applications in sports science, rehabilitation, and rapid-response robotics.

**Privacy Protection.** We strictly prioritized participant privacy. The released modalities—event streams, LiDAR point clouds, and IMU data—inherently obfuscate facial features and skin textures, minimizing the risk of personally identifiable information (PII) leakage.

**Ethical Compliance.** All participants provided written informed consent for their data to be used for research purposes. Furthermore, this study has obtained formal approval from the Institutional Review Board (IRB).

**Data Governance.** To prevent misuse, access to the dataset requires users to sign a strict usage agreement. We reserve the right to revoke access immediately if any violation of these terms is detected.

## F.2. Limitations and Future Work

Despite the advancements presented, our work has certain limitations that open avenues for future research.

**Modal Disparity (2D vs. 3D).** While we provide 2D labels at 1000 Hz, the 3D parameters (SMPL) are currently limited to 60 Hz due to the frequency limitation of IMU-based reconstruction. Consequently, current methodologies cannot yet fully leverage 3D geometric constraints for millisecond-level analysis. Future work will focus on "lifting" these high-frequency 2D cues to generate 1000 Hz 3D motion fields.

**Lack of Comparable Benchmarks.** We primarily evaluated ResPose on *FlashMotion* because existing event-based datasets (e.g., DHP19 [3]) typically provide ground truth at much lower frequencies ($\leq$ 100 Hz). This lack of comparable millisecond-accurate benchmarks limits the feasibility of cross-dataset evaluation for the specific task of High-Temporal Resolution HPE. We hope our dataset will fill this gap and serve as a standard for future algorithms.

## References


[1] Asude Aydin, Mathias Gehrig, Daniel Gehrig, and Davide Scaramuzza. A hybrid ann-snn architecture for low-power and low-latency visual perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5701–5711, 2024. 9

[2] Derya Birant and Alp Kut. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering*, 60(1):208–221, 2007. 4, 5

[3] Enrico Calabrese, Gemma Taverni, Christopher Awai Easthope, Sophie Skriabine, Federico Corradi, Luca Longinotti, Kynan Eng, and Tobi Delbruck. Dhp19: Dynamic vision sensor 3d human pose dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019. 10

[4] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013. 4, 5

[5] Jiaan Chen, Hao Shi, Yaozu Ye, Kailun Yang, Lei Sun, and Kaiwei Wang. Efficient human pose estimation via 3d event point cloud. In *3DV*, 2022. 9

[6] Anil Damle, Victor Minden, and Lexing Ying. Simple, direct and efficient multi-way spectral clustering. *Information and Inference: A Journal of the IMA*, 8(1):181–203, 2019. 5

[7] Gaurvi Goyal, Pham Cong Thuong, Arren Glover, Masayoshi Mizuno, and Chiara Bartolozzi. Graphenet: Event-driven human pose estimation with a graph neural network. In *ICCVW*, pages 4665–4674, 2025. 9

[8] Youngho Kim, Hoonhee Cho, and Kuk-Jin Yoon. From sharp to blur: Unsupervised domain adaptation for 2d human pose estimation under extreme motion blur using event cameras. In *ICCV*, pages 9406–9417, 2025. 9

[9] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 24(127):1–21, 2023. 4, 5

[10] XSENS. Xsens Technologies B.V. https://www.xsens.com/, 2025. 4

[11] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. In *Advances in Neural Information Processing Systems*, 2022. 9

[12] Ming Yan, Yan Zhang, Shuqiang Cai, Shuqi Fan, Xincheng Lin, Yudi Dai, Siqi Shen, Chenglu Wen, Lan Xu, Yuexin Ma, and Cheng Wang. Reli11d: A comprehensive multimodal human motion dataset and method. In *CVPR*, 2024. 5, 9