EventSPD: Sparse Query-Point Depth from Event Cameras

Architecture Document (based on research plan v12)

Yincheng Zhou
Formatted by Claude Code

February 2026

1 Motivation

Every existing event-camera depth method [1, 10] produces a dense $H \times W$ depth map. Yet many downstream tasks—SLAM keypoint tracking, grasping, obstacle avoidance—need depth at only a sparse set of pixels ($K \ll HW$). Dense decoding wastes both latency and power on the majority of pixels that are never used.

We propose **EventSPD**, which predicts depth only at user-specified query pixels. Inference decomposes as:

$$T(K) = \underbrace{T_{\text{precompute}}}_{\text{shared, } \sim 6 \text{ ms}} + \underbrace{\beta \cdot K}_{\sim 5\,\mu\text{s / query}} \tag{1}$$

At K=256 on an RTX 4090, this gives ∼7.8 ms total (estimated from MACs calculation, 2.9× faster than the dense baseline combining F^3 [7]—a predictive event-camera feature encoder—and DAv2 [26]—a dense monocular depth decoder—at ∼23 ms).

No published work targets sparse query-point depth from events. The closest RGB work is InfiniDepth [12] (LIIF-style [5] queries), which lacks attention routing and deformable sampling.

2 Background and Related Work

RGB depth foundations. MiDaS [21] introduced mixed-dataset zero-shot transfer. DPT [20] added ViT backbones with token reassembly. Depth Anything V2 [26] (NeurIPS 2024) scales this further with synthetic pre-training. ZoeDepth [2], Metric3D v2 [11], Marigold [14], and UniDepth [19] represent further advances. All decode densely.

Event-camera depth. E2Depth [10] uses recurrent ConvLSTM [22] for dense event depth. Depth AnyEvent [1] (ICCV 2025) distills DAv2 into an event encoder with a recurrent variant. DERD-Net (NeurIPS 2025) uses a lightweight GRU. F^3 [7] provides a predictive event representation at 120 Hz (HD). All produce dense output.

Query-based decoders. DETR [3] and Deformable DETR [29] decode object queries via cross-attention into encoder features. SAM [16] extends this to promptable segmentation. Perceiver IO [13] generalizes to arbitrary outputs. Our decoder follows the same principle: pre-computed features as static KV context, with each query point cross-attending through multiple layers.

3 Architecture

Overview

EventSPD decomposes inference into two phases:

Phase	Description
A. Shared Precompute	Runs *once per event window*. Encodes raw events into a multi-scale feature pyramid (4 levels), applies temporal recurrence, and pre-computes KV projections shared across all queries.
B. Per-Query Decoder	Runs *independently for each of K queries* in parallel. Five stages progressively refine a query embedding: local feature gathering → local cross-attention → global cross-attention with routing → deformable multi-scale sampling → fused cross-attention → depth prediction.

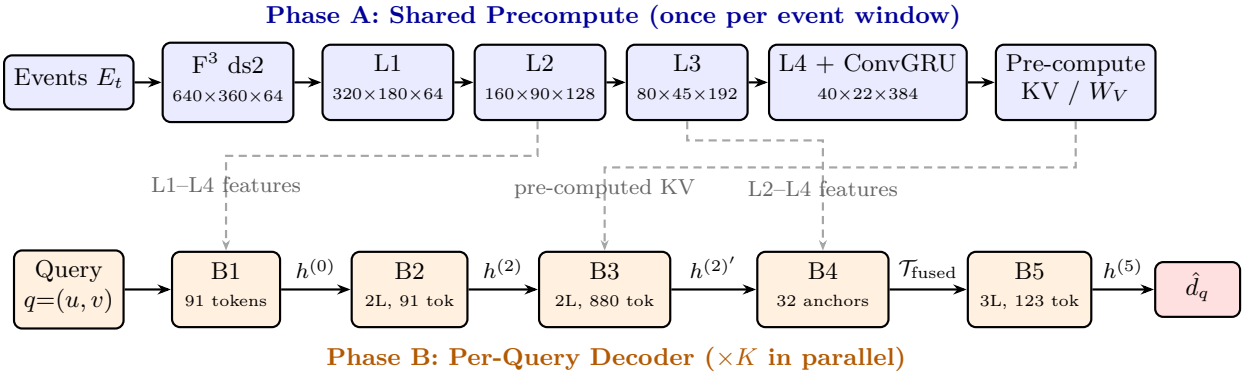

Figure 1: EventSPD architecture overview. Phase A (top) runs once per event window, producing a 4-level feature pyramid and pre-computed KV projections. Phase B (bottom) runs independently for each of K query pixels: five stages progressively refine a query embedding into a depth prediction. Dashed arrows show feature feeds from Phase A into Phase B stages.

Design principles and references. The architecture draws on several established ideas, combined in a novel way for sparse event-camera depth:

Component	What we use	Reference
Event encoding	F^3 ds2 (predictive features, 120 Hz)	[7]
Local backbone (L1–L2)	Large-kernel depthwise conv + GRN	[8, 25]
Mid-range backbone (L3)	Shifted window self-attention	[18]
Temporal state (L4)	Depthwise-separable ConvGRU	[1]
Per-layer KV projections	Fresh KV per decoder layer	[3, 16]
Deformable sampling (B4)	Multi-scale offsets + grid-sample	[29]
Static KV context (B5)	Uncompressed features as KV across layers	[13]
Feature distillation	Cosine similarity to frozen DAv2	[1]
Query-point paradigm	Decode only at requested pixels	[5, 15]

End-to-end data flow. The full pipeline proceeds as follows:

Step	Operation	Output
A1	F^3 ds2: raw events → feature tensor	$F_t \in \mathbb{R}^{640\times360\times64}$
A2	Wide pyramid backbone (ConvNeXt → Swin → SelfAttn → ConvGRU)	L1–L4 feature maps
A3	Pre-compute KV projections for B3; W_V for B4	Shared KV tables
A4	Global calibration from mean-pooled L4	Scale s_t, shift b_t
B1	Gather multi-scale tokens around query q	$\mathcal{T}_{\text{uni}} \in \mathbb{R}^{91\times d}$, $h^{(0)}$
B2	Local cross-attn (2L) over 91 tokens	$h^{(2)}$ (local-aware)
B3	Global cross-attn (2L) over 880 L4 tokens + routing	$h^{(2)'}$ + 32 anchors
B4	Deformable multi-scale read at 32 anchors	$\mathcal{T}_{\text{fused}} \in \mathbb{R}^{123\times d}$
B5	Fused cross-attn (3L) over 123 tokens + depth MLP	$\hat{d}_q$

Key numbers.

Quantity	Value
Core dimension d	192
Feature pyramid	4 levels: 64 / 128 / 192 / 384 channels at strides 4 / 8 / 16 / 32
Cross-attention layers	7 total (2 in B2 + 2 in B3 + 3 in B5); 6 heads, d_h=32
Lookups per query	2,396 (92 local + 2,304 deformable)
MACs per query	∼51.1M (∼4.9 μs wall-clock)
Total parameters	∼11.1M (7.7M backbone + 3.4M decoder)
Precompute latency	∼6.15 ms (F^3 4.5 ms + backbone 1.55 ms + 0.10 ms)

Phase A: Shared Precompute

A1. Event Encoding (F^3 ds2)

F^3 [7] encodes a fixed-duration *event window* E_t (all events accumulated within one time step at 120 Hz) into a dense feature tensor. We use the **ds2** ("down-sample 2×") configuration, which applies stride-2 in the first stage with 64 channels (vs. ds1's 32 channels at full 1280×720):

$$F_t = \mathcal{F}^{\text{ds2}}(E_t) \;\in\; \mathbb{R}^{640\times360\times64} \qquad (\sim 4.5 \text{ ms}) \tag{2}$$

ds2 halves spatial resolution and memory traffic relative to ds1. The resulting 59 MB feature map fits the RTX 4090 L2 cache (72 MB). ds2 requires training first.

A2. Wide Pyramid Backbone

Four-level features at strides $[4, 8, 16, 32]$ and channels $[64, 128, 192, 384]$:

Level	Blocks	Output	Params
Stem	Conv k=3, s=2 + LN	320×180×64	37K
L1	2× ConvNeXt V2, k=7 [25]	320×180×64	74K
L2	3× ConvNeXt V2, k=13 [8]	160×90×128	435K
L3	4× Swin [18], w=8	80×45×192	1,778K
L4	2× Full self-attn	40×22×384	3,542K
ConvGRU	3 gates, DW-sep k=3	40×22×384	886K
Total backbone			**7,179K**

Design rationale. L1–L2 use ConvNeXt with large kernels for local depth features at high resolution. L3 uses windowed attention for mid-range context. L4 uses full self-attention (880 tokens, trivially cheap) for global scene understanding. The ConvGRU at L4 integrates temporal state across event windows [1, 22] (standard ConvGRU formulation; we show only the state update for brevity):

$$G_t^{(4)} = (1-z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{3}$$

where z_t is the update gate, $\tilde{h}_t$ the candidate (both computed via depthwise-separable 3×3 convolutions), and h_0=0. Placing recurrence only at L4 minimizes cost (∼0.8G MACs) while maximizing impact (global receptive field).

Total backbone cost: ∼25.8G MACs, ∼1.55 ms.

A3. Pre-Computed Projections

Pre-computed once per frame and shared across all queries:

Target	Projection	Shape	Params
B3 KV (ℓ=1, 2) [3, 16]	$W_K^{(\ell)}, W_V^{(\ell)}$ per layer	384→192	296K
B4 anchor (W_g)	L4 feature → conditioning	384→192	74K
B4 value (W_V)	Per-level pre-proj [29]	L2: 128→192, L3: id, L4: 384→192	136K
Total A3			**506K**

A4. Calibration

Global scale and shift for converting the decoder's raw output into metric depth, derived from mean-pooled L4:

$$\bar{G}_t^{(4)} = \frac{1}{HW} \sum_i G_t^{(4)}[i] \ \in \ \mathbb{R}^{384} \tag{4}$$

$$s_t = \text{softplus}\big(W_s \, \bar{G}_t^{(4)} + b_s\big), \qquad b_t = W_b \, \bar{G}_t^{(4)} + b_b \tag{5}$$

where $W_s, W_b \in \mathbb{R}^{1\times 384}$ are linear heads (∼0.8K params each). These are used in B5's depth prediction (Eq. 25).

Phase B: Per-Query Decoder ($d = 192$)

Notation conventions. Layer indices ℓ are local to each stage (B2 and B3 each have ℓ=1, 2; B5 has ℓ=1, 2, 3). In B3, $h^{(2)}$ denotes in-place updates starting from B2's output; the final result is written $h^{(2)'}$ to distinguish it. All Fourier positional encodings are written $\phi_n(\cdot)$ where n is the number of frequencies (output dimensionality = $4n$). RPE denotes learned relative position embeddings.

B1. Token Construction

Gathers multi-scale features around query $q = (u, v)$:

Scale	Tok.	Grid / Source	Projection	Embed.
L1 (stride 4)	32	24 grid (5×5 excl. center) + 8 adaptive	MLP: $h_\delta = \mathrm{GELU}(W_{\mathrm{loc}}[f_\delta; \phi_4(\delta)])$	e_{L1}
L2 (stride 8)	25	5×5 grid	W_{v2}: 128→192	e_{L2} + RPE
L3 (stride 16)	25	5×5 grid	identity (192=d)	e_{L3} + RPE
L4 (stride 32)	9	3×3 grid	W_{v4}: 384→192	e_{L4} + RPE
Total	**91**			

The combined token set:

$$\mathcal{T}_{\mathrm{uni}} = [\underbrace{t^{(1)}_{1:32}}_{L1}; \underbrace{t^{(2)}_{1:25}}_{L2}; \underbrace{t^{(3)}_{1:25}}_{L3}; \underbrace{t^{(4)}_{1:9}}_{L4}] \in \mathbb{R}^{91\times 192} \tag{6}$$

Query seed: $h^{(0)} = W_q[f_q^{(1)}; \phi_8(q)] \in \mathbb{R}^{192}$, where $f_q^{(1)} \in \mathbb{R}^{64}$ is the bilinear-sampled L1 center feature and $\phi_8(q) \in \mathbb{R}^{32}$ is a Fourier encoding of the normalized query pixel coordinate $(u/W, v/H)$ with 8 frequencies.

B1 cost: 142K params, 92 bilinear lookups, ∼2M MACs.

B2. Local Multi-Scale Cross-Attention

Two-layer cross-attention decoder ($\ell = 1, 2$) with $h^{(0)}$ as initial query and $\mathcal{T}_{\mathrm{uni}}$ (91 tokens) as context. 6 heads, d_h=32, FFN 192→768→192.

Per-layer KV projections (DETR/SAM/Perceiver IO convention [3, 13, 16]):

$$K^{(\ell)} = W_K^{(\ell)}\ \mathrm{LN}_{\mathrm{kv}}(\mathcal{T}_{\mathrm{uni}}), \quad V^{(\ell)} = W_V^{(\ell)}\ \mathrm{LN}_{\mathrm{kv}}(\mathcal{T}_{\mathrm{uni}}) \in \mathbb{R}^{91\times d} \tag{7}$$

where $\mathrm{LN}_{\mathrm{kv}}$ is shared across B2 layers (normalizes heterogeneous token scales once); $W_K^{(\ell)}, W_V^{(\ell)} \in \mathbb{R}^{d\times d}$ are per-layer.

Pre-LN decoder (each layer = cross-attention + FFN with residuals):

$$h^{(\ell)} \leftarrow h^{(\ell-1)} + \mathrm{MHCrossAttn}^{(\ell)}\big(Q{=}\,\mathrm{LN}_q^{(\ell)}(h^{(\ell-1)}),\ K{=}K^{(\ell)},\ V{=}V^{(\ell)}\big) \tag{8}$$

$$h^{(\ell)} \leftarrow h^{(\ell)} + \mathrm{FFN}^{(\ell)}\big(\mathrm{LN}_{\mathrm{ff}}^{(\ell)}(h^{(\ell)})\big) \tag{9}$$

Output: $h^{(2)} \in \mathbb{R}^d$—local-aware query representation encoding fine L1 texture, mid-level L2/L3 structure, and coarse L4 context from the query neighborhood. B2 is the costliest stage because the 91 tokens are per-query (KV projection runs per query), unlike B3's pre-computed KV.

B2 cost: 891K params (2× $W_Q/W_K/W_V/W_O$ ∼296K + 2× FFN ∼592K + LNs ∼3K), ∼14.2M MACs.

B3. Global L4 Cross-Attention + Routing

Two-layer cross-attention ($\ell = 1, 2$) into the full L4 feature map (880 tokens). KV pairs are *pre-computed in Phase A* ($K_t^{(\ell)}, V_t^{(\ell)} \in \mathbb{R}^{880\times d}$ from A3, per-layer projections $W_K^{(\ell)}, W_V^{(\ell)} \in \mathbb{R}^{d\times 384}$). Per-query cost is only Q projection + attention + FFN—no KV recomputation.

Pre-LN decoder (same structure as B2; $h^{(2)}$ from B2 is updated in-place across B3's two layers):

$$h^{(2)} \leftarrow h^{(2)} + \mathrm{MHCrossAttn}^{(\ell)}\big(Q{=}\,\mathrm{LN}_q^{(\ell)}(h^{(2)}),\ K{=}K_t^{(\ell)},\ V{=}V_t^{(\ell)}\big) \tag{10}$$

$$h^{(2)} \leftarrow h^{(2)} + \mathrm{FFN}^{(\ell)}\big(\mathrm{LN}^{(\ell)}(h^{(2)})\big) \tag{11}$$

6 heads, d_h=32, FFN 192→768→192. After 2 layers the result is denoted $h^{(2)'} \in \mathbb{R}^d$—enriched with global scene context.

Attention-based routing (parameter-free). Average B3's *second-layer* attention weights across heads, select top-R:

$$\bar{\alpha}_q = \frac{1}{H}\sum_{h=1}^{H} \alpha_{q,h} \ \in\ \mathbb{R}^{880}, \qquad R_q = \text{Top-R}(\bar{\alpha}_q,\ R{=}32) \qquad (3.6\%\ \text{selection}) \tag{12}$$

Each $r \in R_q$ maps to pixel coordinate $\mathbf{p}_r$ via L4 grid geometry (40×22, stride 32). Gradient flow through the discrete selection uses a straight-through estimator.

Output: $h^{(2)'}$ (globally enriched) + 32 anchor positions for B4.

B3 cost: 740K params, ∼1.4M MACs per query. Cheap because KV is pre-computed and shared across all queries.

B4. Deformable Multi-Scale Read + Token Fusion

For each anchor $r \in R_q$ (32 from B3), predict sampling offsets and importance weights, then read from the multi-scale pyramid via bilinear grid-sample [29]:

Hyperparameter	**Value**
Heads (H)	6
Levels (L)	3 (L2, L3, L4)
Samples / head / level (M)	4
Anchors (R)	32 (from B3 routing)
Total lookups	$32 \times 6 \times 3 \times 4 = 2{,}304$

Conditioning. Each anchor is conditioned on the query state, anchor content, and spatial offset:

$$\Delta\mathbf{p}_r = \mathbf{p}_r - q \qquad (\text{query-to-anchor offset in pixels}) \tag{13}$$

$$u_r = \mathrm{LN}\big(\mathrm{GELU}(W_u\,[h^{(2)'};\ g_r;\ \phi_8(\Delta\mathbf{p}_r)] + b_u)\big) \ \in\ \mathbb{R}^d \tag{14}$$

where $g_r = W_g\, G_t^{(4)}[\mathbf{p}_r^{(4)}] \in \mathbb{R}^d$ is the anchor's L4 feature (pre-projected 384→192 in A3), and $\phi_8(\Delta\mathbf{p}_r) \in \mathbb{R}^{32}$ is a Fourier encoding (8 frequencies) of the normalized offset $[\Delta u/W,\ \Delta v/H]$. Input dimension: $d + d + 32 = 416$.

Offset and weight prediction. Two shared linear heads from the conditioning:

$$\Delta p_{r,h,\ell,m} = W^{\Delta}\, u_r + b^{\Delta} \ \in\ \mathbb{R}^{H\times L\times M\times 2} \quad (\text{offsets, } d{\to}144) \tag{15}$$

$$\beta_{r,h,\ell,m} = W^{a}\, u_r + b^{a} \ \in\ \mathbb{R}^{H\times L\times M} \quad (\text{weights, } d{\to}72) \tag{16}$$

Offsets are unbounded (no tanh), following Deformable DETR [29]. Zero-initialized with 0.1× LR.

Sampling with per-level normalization. Each offset is normalized by the spatial extent S_ℓ of level ℓ (making offsets scale-invariant):

$$p_{\text{sample}} = \mathbf{p}_r^{(\ell)} + \frac{\Delta p_{r,h,\ell,m}}{S_\ell} \tag{17}$$

$$f_{r,h,\ell,m} = \text{GridSample}\big(\hat{F}_t^{(\ell)},\ \text{Normalize}(p_{\text{sample}})\big) \tag{18}$$

where $\mathbf{p}_r^{(\ell)} = \mathbf{p}_r / s_\ell$ maps the anchor to level-ℓ native coordinates, $\hat{F}_t^{(\ell)}$ are the pre-projected features (already at d=192 from A3's W_V), and $S_\ell \in \{160, 80, 40\}$ for L2–L4.

Multi-head per-anchor aggregation. Per-head softmax over $L{\times}M$=12 samples, weighted sum, then concat + output projection:

$$a_{r,h,\ell,m} = \frac{\exp(\beta_{r,h,\ell,m})}{\sum_{\ell',m'} \exp(\beta_{r,h,\ell',m'})}, \qquad \tilde{h}_{r,h} = \sum_{\ell,m} a_{r,h,\ell,m}\, v_{r,h,\ell,m} \tag{19}$$

$$h_r = W_O\,[\tilde{h}_{r,1};\ \ldots;\ \tilde{h}_{r,H}] + b_O\ \in\ \mathbb{R}^d \tag{20}$$

where $v_{r,h,\ell,m} \in \mathbb{R}^{d/H}$ is the head-partitioned feature from $f_{r,h,\ell,m}$.

Token fusion. The 32 deformable tokens are appended with a type embedding $e_{\text{deform}} \in \mathbb{R}^d$:

$$\mathcal{T}_{\text{fused}} = [\,\mathcal{T}_{\text{uni}};\ h_{r_1}{+}e_{\text{deform}};\ \ldots;\ h_{r_{32}}{+}e_{\text{deform}}\,]\ \in\ \mathbb{R}^{123\times d} \tag{21}$$

Budget: $32 \times 72 =$ **2,304** deformable lookups (all remote).

B4 cost: 159K params (conditioning ∼80K + offsets ∼28K + weights ∼14K + W_O ∼37K), ∼5.1M MACs. Dominates **wall-clock** (51%) due to scattered memory reads despite only 10% of compute.

B5. Fused Cross-Attention + Depth Head

Three-layer cross-attention decoder ($\ell = 1, 2, 3$, local to B5) with $h^{(2)\prime}$ as initial query and $\mathcal{T}_{\text{fused}}$ (123 tokens) as KV. Same architecture as B2 (6 heads, d_h=32, FFN 192→768→192). The 91 local tokens remain as *uncompressed* reference in KV—following DETR/SAM's convention of static context across decoder layers [3, 16].

Per-layer KV projections (independent from B2):

$$K^{(\ell)} = W_K^{(\ell)}\ \text{LN}_{\text{kv2}}(\mathcal{T}_{\text{fused}}), \quad V^{(\ell)} = W_V^{(\ell)}\ \text{LN}_{\text{kv2}}(\mathcal{T}_{\text{fused}})\ \in\ \mathbb{R}^{123\times d} \tag{22}$$

Pre-LN decoder (same structure as B2):

$$h^{(\ell)} \leftarrow h^{(\ell-1)} + \text{MHCrossAttn}^{(\ell)}\big(Q{=}\,\text{LN}_q^{(\ell)}(h^{(\ell-1)}),\ K{=}K^{(\ell)},\ V{=}V^{(\ell)}\big) \tag{23}$$

$$h^{(\ell)} \leftarrow h^{(\ell)} + \text{FFN}^{(\ell)}\big(\text{LN}_{\text{ff}}^{(\ell)}(h^{(\ell)})\big) \tag{24}$$

Output after 3 layers: $h^{(5)} \coloneqq h_{\text{B5}}^{(3)}$ (we write $h^{(5)}$ to reflect the total decoder depth: 2 B2 + 3 B5 = 5 cross-attention layers).

Depth prediction:

$$z_q = W_{z2} \cdot \text{GELU}(W_{z1}\, h^{(5)} + b_{z1}) + b_{z2} \qquad (\text{MLP } 192{\to}384{\to}1) \tag{25}$$

$$\hat{d}_q = \underbrace{\frac{1}{\text{softplus}(s_t\, z_q + b_t) + \varepsilon}}_{\hat{\rho}_q\ =\ \text{predicted inverse depth}} \tag{26}$$

where s_t, b_t are the global calibration parameters from A4.

B5 cost: 1,411K params (3× $W_Q/W_K/W_V/W_O$ ∼444K + 3× FFN ∼888K + depth MLP ∼74K + LNs ∼5K), ∼28.4M MACs.

Summary

$$h^{(0)} \xrightarrow[\text{2L, 91 tok}]{\text{B2}} h^{(2)} \xrightarrow[\text{2L, 880 tok}]{\text{B3}} h^{(2)\prime} \xrightarrow[\text{3L, 123 tok}]{\text{B5}} h^{(5)} \to \hat{d}_q \tag{27}$$

Stage	Params	Lookups	MACs/query
B1: Token construction	142K	92	2.0M
B2: Local cross-attn (2L, 91 tok)	891K	—	14.2M
B3: Global cross-attn (2L, 880 tok)	740K	—	1.4M
B4: Deformable read (32 anchors)	159K	2,304	5.1M
B5: Fused cross-attn (3L, 123 tok)	1,411K	—	28.4M
Decoder total	**3,343K**	**2,396**	**51.1M**

Total: ∼**11.1M params** (7,688K backbone + 3,343K decoder). 7 cross-attention layers total (2+2+3). ∼44 nonlinear stages per query (25 backbone + 19 decoder)—comparable to DAv2-S's ∼24 ViT sub-layers, ensuring sufficient model capacity at 2.3× fewer parameters.

4 Training

4.1 Loss Functions

Four core losses, all active during training:

$$\mathcal{L} = L_{\text{point}} + \lambda_{\text{si}}\, L_{\text{silog}} + L_{\text{dense}} + L_{\text{feat}} \tag{28}$$

with $\lambda_{\text{si}} = 0.5$. Training-only components (L_{dense} heads, L_{feat} connectors) are discarded at inference.

Point loss (data fit). Huber loss on predicted vs. ground-truth inverse depth at each query $q \in Q_v$ (the set of queries with valid GT). The predicted inverse depth $\hat{\rho}_q$ is defined in B5's depth head (Eq. 25): $\hat{\rho}_q = \text{softplus}(s_t z_q + b_t) + \varepsilon = 1/\hat{d}_q$.

$$L_{\text{point}} = \frac{1}{|Q_v|} \sum_{q \in Q_v} \text{Huber}\big(\hat{\rho}_q - \rho_q^*\big) \tag{29}$$

Scale-invariant log loss (structural). Enforces consistent relative depth structure independent of global scale:

$$L_{\text{silog}} = \sqrt{\frac{1}{|Q_v|} \sum_{q \in Q_v} \delta_q^2 \;-\; \lambda_{\text{var}} \Big(\frac{1}{|Q_v|} \sum_{q \in Q_v} \delta_q \Big)^2}, \qquad \delta_q = \log \hat{d}_q - \log d_q^* \tag{30}$$

with $\lambda_{\text{var}} = 0.5$.

Dense backbone auxiliary (training only). Dense inverse-depth prediction at L2 and L3 resolutions via 1×1 conv heads, providing 18,000 gradient sources (14,400 at L2 + 3,600 at L3) vs. K=256 sparse queries—56× more spatial coverage [6, 27]:

$$L_{\text{dense}} = \sum_{\ell \in \{2,3\}} \frac{\lambda_{\text{d}\ell}}{|P_v^{(\ell)}|} \sum_{p \in P_v^{(\ell)}} \text{Huber}\big(\hat{\rho}_{\text{dense}}^{(\ell)}(p) - \rho^*(p)\big) \tag{31}$$

where $P_v^{(\ell)}$ is the set of pixels at level ℓ's resolution with valid GT. $\lambda_{\text{d2}} = 0.5$, $\lambda_{\text{d3}} = 0.25$. A warm-down schedule reduces the overall L_{dense} weight from 1.0 → 0.25 over training (dense dominates early while sparse routing is untrained, then fades as the query decoder matures).

Feature distillation from DAv2 (training only). Cosine similarity between our backbone features and frozen DAv2 ViT-S [26] intermediate features, with sparsity-aware weighting [1]:

$$L_{\text{feat}} = \sum_{\ell \in \{3,4\}} \frac{\lambda_{\text{f}\ell}}{|P^{(\ell)}|} \sum_{p \in P^{(\ell)}} w_p \left(1 - \frac{\hat{F}^{(\ell)}_{\text{event}}(p) \cdot \hat{F}^{(\ell)}_{\text{DAv2}}(p)}{\|\hat{F}^{(\ell)}_{\text{event}}(p)\| \, \|\hat{F}^{(\ell)}_{\text{DAv2}}(p)\|} \right) \tag{32}$$

$\hat{F}^{(\ell)}_{\text{event}}$ are our features projected through a connector MLP: $\hat{F}^{(\ell)}_{\text{event}} = W^{(\ell)}_{c2} \cdot \text{GELU}(W^{(\ell)}_{c1} \cdot F^{(\ell)}_{\text{event}})$.

Level	Our feature	DAv2 teacher	Connector MLP	Params
L3	80×45×192	ViT-S layer 6 (384-d)	192→384→384	148K
L4	40×22×384	ViT-S layer 12 (384-d)	384→384→384	296K
Total connector (training only, discarded at inference)				**444K**

Sparsity-aware weight: $w_p = \min(n_p/\bar{n},\ 2.0)$, where n_p is the local event count and $\bar{n}$ the spatial mean.

Loss hyperparameters summary.

Loss	Weight	Scope	Inference
L_{point}	1.0	Sparse queries with GT	—
L_{silog}	λ_{si}=0.5	All queries	—
L_{dense} (L2)	λ_{d2}=0.5	Dense at L2 (160×90)	Discarded
L_{dense} (L3)	λ_{d3}=0.25	Dense at L3 (80×45)	Discarded
L_{feat} (L3)	λ_{f3}=0.1	Cosine vs. DAv2 layer 6	Discarded
L_{feat} (L4)	λ_{f4}=0.1	Cosine vs. DAv2 layer 12	Discarded

4.2 Training Data

Dataset	GT Type	Resolution	Density	Metric
M3ED [4]	Real LiDAR (VLP-16)	1280×720	Sparse (∼5–10%)	Yes
DSEC [9]	Stereo + LiDAR	640×480	Semi-dense (∼30–50%)	Yes
MVSEC [28]	LiDAR + MoCap	346×260	Sparse	Yes
TartanAir v2 [24]	Synthetic (Unreal)	640×640	Dense (100%)	Yes
M3ED pseudo	DAv2 pseudo labels	1280×720	Dense (100%)	No (relative)

Real GT (LiDAR/stereo) is primary—LiDAR sparsity is not a problem since queries are sampled at GT-valid locations. DAv2 pseudo labels provide supplementary dense coverage (used with scale-invariant losses only).

4.3 Query Sampling

Multinomial sampling with category priority—each draw first selects a category, then samples a pixel from that category:

Weight	Category
40%	LiDAR-valid pixels (real GT, highest quality)
20%	DAv2 pseudo-labeled pixels without LiDAR (dense coverage)
15%	Event-dense regions (high-signal areas)
15%	High depth-gradient regions (boundary quality)
10%	Hard-example regions from spatial loss map [17, 23]

Hard-example mining. A spatial loss map $\mathcal{M}(u, v)$ at stride 16 (80×45 grid) is maintained via EMA:

$$\mathcal{M}_{t+1}(u, v) = 0.99 \cdot \mathcal{M}_t(u, v) + 0.01 \cdot \bar{L}_{\text{local}}(u, v) \tag{33}$$

where $\bar{L}_{\text{local}}$ is the mean loss of queries falling in each grid cell. The hard-example category samples cells with probability $\propto \mathcal{M}(u, v)$, focusing training on regions where the model struggles [15].

Train-large-K, infer-small-K. Train with $K_{\text{train}} = 2{,}048$ (8× the inference budget of $K_{\text{infer}} = 256$). The decoder has no inter-query coupling, so K is a free parameter. Larger K provides more loss terms per batch, better spatial coverage for the backbone, and more diverse routing patterns for B3.

4.4 Training Schedule

	Stage 1: Relative depth (∼15–20 epochs)	**Stage 2: Metric fine-tuning** (∼10 epochs)
L_{point}	LiDAR-valid queries	Real GT queries
L_{silog}	All queries (λ_{si}=0.5); pseudo-label queries use L_{silog} only	Real GT queries (λ_{si}=0.5)
L_{dense}	Active; warm-down 1.0→0.25	Active; weight reduced to 0.1
L_{feat}	Active (frozen DAv2 on aligned RGB)	Disabled (real data may lack RGB)
Datasets	M3ED, DSEC, TartanAir v2 (LiDAR + pseudo + RGB)	M3ED, DSEC, MVSEC (real GT only, no pseudo)
Backbone	Frozen (pre-trained F^3)	Optionally unfreeze at 0.1× LR
ConvGRU	Trains from scratch; BPTT 4 windows; h_0=0	Continues training; BPTT 4 windows
K_{train}	2,048	2,048
Evaluation	—	MVSEC outdoor_day1/day2

4.5 Regularization

Technique	**Setting**
Attention dropout	$p = 0.1$ on all 7 cross-attn layers (B2+B3+B5)
Weight decay	0.01
Gradient clipping	max norm 1.0
Precision	Mixed (bf16)

5 Projected Performance

K	**Decoder**	**Total**	**Hz**	**vs. Dense**
1	0.3 ms	6.5 ms	154	3.5×
64	0.6 ms	6.8 ms	147	3.4×
256	1.6 ms	7.8 ms	128	**2.9**×
1,024	5.4 ms	11.6 ms	86	2.0×

Dense baseline: F^3 ds1 + DAv2 ≈ 23 ms. Crossover at $K \approx 3{,}310$. Precompute (∼6.15 ms) dominates; F^3 encoder (∼4.5 ms) is 73% of precompute.

Per-query bottleneck is **memory bandwidth**, not compute. Arithmetic intensity is ∼14.6 MACs/byte, well below the FP16 roofline (∼165). B4's scattered reads dominate wall-clock; compute-only changes (layer depth, FFN width) have negligible latency impact.

	DAv2-S	EventSPD	Ratio
Total params	∼25M	∼11.1M	2.3× smaller
Decoder params	∼3M	∼3.4M	comparable
Dense pixels	268K (518^2)	2,396 / query	112× fewer
Latency (K=256)	∼23 ms	∼7.8 ms	2.9× faster
Temporal state	none	ConvGRU	unique

References

[1] Luca Bartolomei, Enrico Mannocci, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. Depth AnyEvent: A cross-modal distillation paradigm for event-based monocular depth estimation. In *International Conference on Computer Vision (ICCV)*, 2025.

[2] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. ZoeDepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020.

[4] Kenneth Chaney, Fernando Cladera, Ziyun Wang, Anthony Bisulco, M. Ani Hsieh, Christopher Korpela, Vijay Kumar, Camillo Jose Taylor, and Kostas Daniilidis. M3ED: Multi-robot, multi-sensor, multi-environment event dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023.

[5] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. Oral.

[6] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Ross Girshick. Masked-attention mask transformer for universal image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] Richeek Das, Kostas Daniilidis, and Pratik Chaudhari. Fast feature fields (F^3): A predictive representation of events. *arXiv preprint arXiv:2509.25146*, 2025.

[8] Xiaohan Ding, Yiyuan Zhang, Yixiao Ge, Sijie Zhao, Lin Song, Xiangyu Yue, and Ying Shan. UniRepLKNet: A universal perception large-kernel ConvNet for audio, video, point cloud, time-series and image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[9] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A stereo event camera dataset for driving scenarios. In *IEEE Robotics and Automation Letters (RA-L)*, 2021.

[10] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events. In *International Conference on 3D Vision (3DV)*, 2020.

[11] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Kaixuan Wang, Hao Chen, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3D v2: A versatile monocular geometric foundation

model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(12):10579–10596, 2024.

[12] InfiniDepth Authors. InfiniDepth: Continuous depth estimation at arbitrary resolution. *arXiv preprint arXiv:2601.03252*, 2026.

[13] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations (ICLR)*, 2022.

[14] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Mez, Patrick Dauber, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[15] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *International Conference on Computer Vision (ICCV)*, 2023.

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision (ICCV)*, 2017.

[18] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *International Conference on Computer Vision (ICCV)*, 2021.

[19] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[20] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *International Conference on Computer Vision (ICCV)*, 2021.

[21] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(3):1623–1637, 2022.

[22] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[23] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[24] 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 (IROS)*, 2020.

[25] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[26] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[27] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[28] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multi vehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters (RA-L)*, 3(3):2032–2039, 2018.

[29] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021. Oral.