RILA: A Radar-Native Structured Interface from Sparse mmWave Point Clouds to Large Language Models

Shenglei Li*
Waseda University
shenglei_li@aoni.waseda.jp

Tomoji Kishi
Waseda University
kishi@waseda.jp

Abstract

Millimeter-wave radar is promising for privacy-preserving human sensing, yet current radar-to-language pipelines typically map sparse point clouds into either global clip features or discrete token codes that are only weakly matched to the structure expected by large language models (LLMs). We present **RILA**, a **r**adar-native structured **i**nterface layer for **l**arge-language-model **a**daptation, which converts sparse mmWave radar point clouds into *LLM-readable event-aware tokens*. RILA combines kinematic phase-space tokenization, flow-aware dual-order serialization, dual-scan selective state-space encoding, and an event-aware token abstraction that exposes both global clip context and temporally localized motion units to the language model. The same interface supports three language tasks through a shared decoder: clip summary generation, ordered event description, and temporal question answering. To stabilize the interface before instruction tuning, we introduce a structured interface-language alignment objective over clip and event tokens. We evaluate RILA using physics-aware synthetic pretraining and real-world mmWave adaptation under controlled radar-native baselines and matched decoder settings. This formulation reframes radar-to-LLM modeling as an interface design problem and studies how radar-native structure, rather than only generic cross-modal projection, determines whether sparse mmWave point clouds can be effectively understood by LLMs. Project page: https://github.com/Frozenshot/RILA.

1 Introduction

Millimeter-wave (mmWave) radar is an attractive sensing modality for privacy-sensitive human understanding because it remains informative in low-light, visually degraded, and partially occluded environments while revealing much less identity-specific appearance than RGB sensing [23, 21, 30, 33, 15, 39]. Over the past decade, radar-based human sensing has progressed from micro-Doppler activity recognition and people tracking [11, 10, 41] to point-cloud activity modeling [9], skeletal pose estimation [28], human mesh recovery [37, 3], multimodal body benchmarks [1], and scene-flow reasoning on mmWave point clouds [5]. However, most existing pipelines still terminate in structured labels, poses, or meshes rather than open-ended language.

This limitation matters because many practical deployments require more than recognition. In assisted living, rehabilitation, and ambient intelligence, the desired output is often a semantic and temporally organized account of what happened, e.g., that a subject first approached a chair, then turned, then sat down. Such outputs are naturally linguistic. At the same time, sparse mmWave point clouds are unordered, discontinuous, and event-structured, which makes direct radar-to-language modeling substantially harder than standard video captioning or clip-level activity recognition.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In this paper, we argue that radar-to-language modeling is fundamentally an interface design problem rather than only an encoder-design problem. Generic point-cloud SSMs can serialize geometry, and video-token methods can compress dense visual streams, but sparse radar-to-language modeling needs a different interface: detections must retain kinematics and expose temporal event units before reaching the LLM. Existing radar-language pipelines typically compress radar into global clip features or discrete token codes that are convenient for language modeling but may discard motion continuity and transition structure. We instead construct a radar-native structured interface that converts sparse mmWave point clouds into *LLM-readable event-aware tokens*. RILA preserves radar-native kinematics, abstracts local motion into event tokens, and reuses the same compact prefix for clip summarization, localized event description, and temporal question answering (Fig. 1).

Contributions. We make three contributions: (i) we formulate sparse mmWave radar-to-language modeling as radar-native interface design rather than generic encoder projection; (ii) we propose RILA, which couples 4D phase-space tokenization, flow-aware dual-order serialization, selective state-space encoding, and event-aware abstraction into LLM-readable prefix tokens; and (iii) we evaluate one shared interface across clip summary, ordered event description, and temporal QA under physics-aware pretraining, real-world adaptation, and controlled radar-native comparisons.

Related Work and Positioning Radar-based human sensing has evolved from micro-Doppler recognition and tracking [11, 10, 41, 14, 40], to point-cloud activity modeling, pose/mesh recovery, multimodal body benchmarks, and scene-flow estimation on mmWave point clouds [9, 28, 37, 3, 1, 5]. Recent dual-mmWave radar systems have also explored sequential clustering for sparse and noisy detections [16] and collaborative sensing for smart spaces [17]. These works improve radar interpretation, but remain centered on classification, regression, tracking, or geometry reconstruction rather than language-level understanding. In parallel, MotionGPT, PointLLM, LiDAR-LLM, and SensorLLM show that non-visual inputs can be interfaced with LLMs through encoder-projector-alignment pipelines [8, 36, 38, 18]. Within RF sensing, mmCLIP studies signal–text alignment for zero-shot HAR, while RadarLLM is the closest prior radar-language system and uses a discrete radar tokenizer plus a radar-aware language model for mmWave point-cloud-to-text generation [2, 13]. These works establish feasibility, but also show that the representation consumed by the LLM is a modality-specific design choice.

Two nearby method families clarify what RILA does not claim. Point-cloud sequence models such as P4Transformer, PointMamba, and Point Cloud Mamba emphasize serialization for 3D perception [6, 7, 4, 19, 40]. RILA instead uses serialization as part of a radar-to-language interface: the geometry order is defined over radar phase space (x, y, z, v), while the flow order preserves short-horizon motion continuity for event-token construction. Video event-token and temporal-localization methods compress or structure dense visual embeddings for video LLMs [35, 24]. Sparse mmWave radar lacks such dense visual semantics, so event abstraction must be induced from discontinuous detections with Doppler, sparsity, and reflection artifacts.

Our positioning is therefore specific. We do not claim a new radar-to-language task, nor do we claim that Mamba, Hilbert ordering, or event tokens are individually new. The contribution is their radar-native coupling: sparse radar kinematics are serialized in complementary geometry and motion orders, compressed into global and event-aware tokens, and reused by one LLM backend across summary, localized event description, and temporal QA.

2 Method

RILA organizes radar-language modeling around a two-stage interface. A *radar-native interface layer* first converts sparse mmWave point clouds into structured interface tokens, and a language model then consumes these tokens under task-specific prompts. We refer to this interface as **RILA** (**R**adar **I**nterface **L**ayer for large-language-model **A**daptation).

Given a radar point-cloud sequence

$$\mathcal{P}_{1:T} = \{P_1, \ldots, P_T\},$$

our goal is to construct an interface sequence

$$\mathcal{I}(\mathcal{P}_{1:T}) = [\bar{u}^{\mathrm{clip}}, \bar{u}_1^{\mathrm{evt}}, \ldots, \bar{u}_M^{\mathrm{evt}}],$$

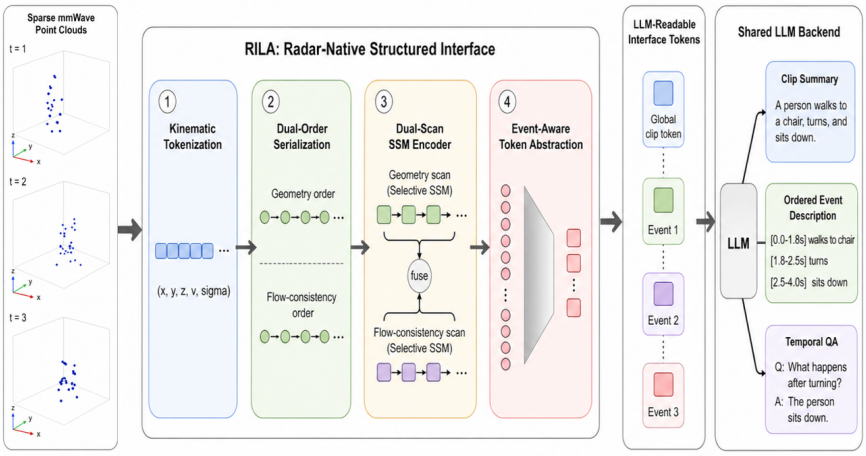

Figure 1: Overview of RILA. Sparse mmWave point clouds are organized into global and event-aware prefix tokens for one shared LLM. Illustrated event spans are schematic; inference spans are predicted from radar.

where $\bar{u}^{\text{clip}}$ is a global clip token and $\bar{u}_m^{\text{evt}}$ are event-aware tokens projected into the embedding space of the LLM. The same interface is then consumed by a shared language model under different prompts for three tasks: clip summary, ordered event description, and temporal question answering.

RILA consists of four components: kinematic phase-space tokenization, flow-aware dual-order serialization, dual-scan selective state-space encoding, and event-aware token abstraction. These stages map a sparse radar sequence to a compact set of global and event-aware interface tokens.

2.1 Kinematic Phase-Space Tokenization

At frame t, the radar produces a sparse set of detections

$$P_t = \{o_{t,i}\}_{i=1}^{N_t}, \qquad o_{t,i} = [x_{t,i}, y_{t,i}, z_{t,i}, v_{t,i}, \sigma_{t,i}]^\top,$$

where (x, y, z) denote 3D position, v denotes radial velocity, and σ denotes reflected signal intensity. Because N_t varies with sparsity, clutter, and multipath, we impose a fixed per-frame token budget by keeping the top-K detections after standard radar preprocessing.

We normalize only the kinematic coordinates (x, y, z, v) using the hardware limits of the sensor:

$$\tilde{p}_{t,i} = \text{clamp}\left(\frac{1}{2} + \frac{1}{2}\begin{bmatrix} x_{t,i}/R_x \\ y_{t,i}/R_y \\ z_{t,i}/R_z \\ v_{t,i}/V_{\max} \end{bmatrix}, 0, 1\right), \tag{1}$$

where R_x, R_y, R_z are the maximum unambiguous spatial ranges and $V_{\max}$ is the maximum unambiguous Doppler velocity. We deliberately exclude σ from the ordering space and keep it only as an attached feature. This avoids intensity-driven topological jumps, where spatially distant points would otherwise become adjacent merely because they share similar reflectivity.

To inject motion cues without relying on full long-horizon tracking, we estimate short-horizon motion descriptors between consecutive frames. For each point $o_{t,i}$, we find a locally consistent predecessor in frame $t-1$ by minimizing

$$j^\star(t,i) = \arg\min_{j \in \mathcal{N}(t,i)} \left(\|\tilde{p}_{t,i}^{xyz} - \tilde{p}_{t-1,j}^{xyz}\|_2 + \beta_v |\tilde{v}_{t,i} - \tilde{v}_{t-1,j}| \right), \tag{2}$$

where $\mathcal{N}(t,i)$ is a local candidate set. If the minimum cost exceeds a matching threshold τ_{match}, the point is treated as unmatched. We then define

$$\delta p_{t,i} = \tilde{p}_{t,i}^{xyz} - \tilde{p}_{t-1,j^\star}^{xyz}, \qquad \delta v_{t,i} = \tilde{v}_{t,i} - \tilde{v}_{t-1,j^\star}. \tag{3}$$

These descriptors are not interpreted as full-body trajectories. Instead, they provide local motion continuity cues for serialization and input-dependent state transitions.

Each radar token is represented as

$$f_{t,i} = [\tilde{p}_{t,i},\ \sigma_{t,i},\ \delta p_{t,i},\ \delta v_{t,i}], \tag{4}$$

with zero-filled motion descriptors for unmatched points.

2.2 Flow-Aware Dual-Order Serialization

Sparse radar point clouds are unordered and dynamically discontinuous. A single serialization is often insufficient: geometry-preserving orders retain local neighborhoods but may break evolving motion fragments, while motion-driven orders alone may lose stable spatial structure. We therefore construct two complementary orders.

Geometry order. For each frame, we quantize the scaled Hilbert-order coordinates $\tilde{p}^G_{t,i}(\lambda_v) \in [0,1]^4$ with resolution k and compute a 4D Hilbert index

$$u^G_{t,i} = H\big(\lfloor (2^k - 1)\tilde{p}^G_{t,i}(\lambda_v) \rfloor\big), \tag{5}$$

where $H(\cdot)$ is the Hilbert mapping [29]. We then sort the frame points by $u^G_{t,i}$ and concatenate frames chronologically:

$$S^G = \mathrm{Concat}\ \big(\mathrm{SortByHilbert}(P_1), \ldots, \mathrm{SortByHilbert}(P_T)\big). \tag{6}$$

This order preserves local proximity under the calibrated 4D radar phase-space scale while retaining the natural temporal order of the clip.

Motion-consistency order. We next construct a motion-consistency order. Using the local associations from Eq. (2), we link points into short tracklets across nearby frames. Each tracklet is grown greedily under the same spatial-velocity consistency rule, without enforcing long-term identity preservation. We then serialize tracklets rather than raw frames:

$$S^F = \mathrm{SerializeTracklets}(\mathcal{T}), \tag{7}$$

where tracklets are ordered by their first appearance time and the Hilbert index of their first token. This order makes locally coherent motion fragments more contiguous in 1D and is especially useful around transitions such as turning, reaching, sitting, or changing direction.

The two orders encode different inductive biases. S^G is stable and locality-preserving, while S^F is transition-sensitive and more directly aligned with evolving motion fragments.

2.3 Dual-Scan Selective State-Space Encoding

We encode S^G and S^F using two selective state-space branches with untied parameters. For each branch $b \in \{G, F\}$, let x^b_n denote the n-th token after linear projection and positional augmentation.

The state update is

$$h^b_n = \bar{A}^b_n h^b_{n-1} + \bar{B}^b_n x^b_n, \qquad y^b_n = C^b_n h^b_n + D^b x^b_n, \tag{8}$$

where $\bar{A}^b_n = \exp(\Delta^b_n A^b)$ is the discretized state transition and $\bar{B}^b_n$ denotes the discretized input map. The stability properties are discussed in Appendix F.2. To keep the transition stable, we parameterize

$$A^b = -\operatorname{diag}(\exp(a^b)). \tag{9}$$

The input-dependent step size is defined as

$$\Delta^b_n = \mathrm{softplus}\big(w^{b\top}_\Delta x^b_n + \gamma_p \|\delta p_n\|_2 + \gamma_v |\delta v_n|\big) + \varepsilon_\Delta, \tag{10}$$

where $\gamma_p, \gamma_v \geq 0$ are learnable. Larger local kinematic changes increase attenuation of stale states near motion discontinuities.

Because the two branches operate on different permutations, we map their outputs back to a shared frame-major layout before fusion. Let π^G and π^F denote the two serialization permutations. The canonicalized outputs are

$$\tilde{Y}^G = (\pi^G)^{-1}(Y^G), \qquad \tilde{Y}^F = (\pi^F)^{-1}(Y^F).$$

Here Y^G and Y^F denote the branch outputs in their respective serialized orders. We fuse them token-wise using gated linear fusion:

$$g_n = \sigma\big(W_g[\tilde{y}_n^G \| \tilde{y}_n^F]\big), \tag{11}$$

$$u_n = g_n \odot W_G \tilde{y}_n^G + (1 - g_n) \odot W_F \tilde{y}_n^F. \tag{12}$$

The fused token sequence $U = \{u_n\}$ is then reshaped back into frame-wise token groups. This design differs from simply scanning one order in two directions: the two branches encode two semantically different orders and therefore expose complementary structure to the interface.

2.4 Event-Aware Token Abstraction

The output of the encoder is still a dense radar stream. To make it LLM-readable, RILA converts the stream into a compact set of structured interface tokens.

We first pool the K tokens within each frame:

$$r_t = \mathrm{Pool}\left(\{u_{t,i}\}_{i=1}^{K}\right), \tag{13}$$

where $\mathrm{Pool}(\cdot)$ is implemented as attention pooling.

A lightweight boundary head predicts event starts and ends:

$$[p_t^{\mathrm{start}},\, p_t^{\mathrm{end}}] = \sigma(\mathrm{MLP}_{\mathrm{bnd}}(r_t)). \tag{14}$$

A monotonic decoder then converts these scores into a non-overlapping sequence of event spans

$$\hat{\mathcal{S}} = \{(\hat{b}_m, \hat{e}_m)\}_{m=1}^{\hat{M}}.$$

When event annotations are available during training, they supervise this boundary head; otherwise the event spans are treated as latent proposals.

For each span (b_m, e_m), we compute a segment representation

$$z_m = \mathrm{SegPool}\left(\{r_t\}_{t=b_m}^{e_m}\right), \tag{15}$$

and a global clip representation

$$z^{\mathrm{clip}} = \mathrm{Pool}\left(\{r_t\}_{t=1}^{T}\right). \tag{16}$$

To make segment tokens explicitly event-aware, we append temporal metadata:

$$\eta_m = \left[\, \mathrm{PE}\!\left(\frac{b_m}{T}\right) \,\middle\|\, \mathrm{PE}\!\left(\frac{e_m}{T}\right) \,\middle\|\, \mathrm{PE}\!\left(\frac{e_m - b_m}{T}\right) \right], \tag{17}$$

and form

$$u_m^{\mathrm{evt}} = W_{\mathrm{evt}}[z_m \| \eta_m], \qquad u^{\mathrm{clip}} = W_{\mathrm{clip}} z^{\mathrm{clip}}. \tag{18}$$

Finally, both clip and event tokens are projected into the embedding dimension of the LLM:

$$\bar{u}^{\mathrm{clip}} = W_{\mathrm{llm}} u^{\mathrm{clip}}, \qquad \bar{u}_m^{\mathrm{evt}} = W_{\mathrm{llm}} u_m^{\mathrm{evt}}. \tag{19}$$

The resulting sequence

$$\mathcal{I}(\mathcal{P}_{1:T}) = [\bar{u}^{\mathrm{clip}}, \bar{u}_1^{\mathrm{evt}}, \ldots, \bar{u}_M^{\mathrm{evt}}]$$

is the core output of the interface.

2.5 Interface-Language Alignment and Training

To stabilize the interface before full instruction tuning, we align the clip token and event-aware tokens to text embeddings and optimize the shared LLM under task-specific prompts.

Let $\phi(\cdot)$ be a frozen text encoder and y^{clip} the clip-summary text target. We define

$$\hat{u}^{\mathrm{clip}} = \mathrm{Norm}(W_c u^{\mathrm{clip}}), \qquad \hat{e}^{\mathrm{clip}} = \mathrm{Norm}(W_t \phi(y^{\mathrm{clip}})), \tag{20}$$

and optimize the clip-level contrastive loss

$$\mathcal{L}_{\mathrm{clip-align}} = -\log \frac{\exp((\hat{u}^{\mathrm{clip}})^\top \hat{e}^{\mathrm{clip}}/\tau)}{\sum_j \exp((\hat{u}^{\mathrm{clip}})^\top \hat{e}_j^{\mathrm{clip}}/\tau)}. \tag{21}$$

Here $\tau > 0$ denotes the contrastive temperature, with negatives drawn from the current batch. When event descriptions are available, we similarly align each event-aware token with its localized text target y_m for event m:

$$\hat{u}_m^{\mathrm{evt}} = \mathrm{Norm}(W_e u_m^{\mathrm{evt}}), \qquad \hat{e}_m^{\mathrm{evt}} = \mathrm{Norm}(W_t \phi(y_m)), \tag{22}$$

$$\mathcal{L}_{\mathrm{evt-align}} = -\frac{1}{M} \sum_{m=1}^{M} \log \frac{\exp((\hat{u}_m^{\mathrm{evt}})^\top \hat{e}_m^{\mathrm{evt}}/\tau)}{\sum_j \exp((\hat{u}_m^{\mathrm{evt}})^\top \hat{e}_j^{\mathrm{evt}}/\tau)}. \tag{23}$$

We refer to this event-level interface alignment as **SGSA** in the experiments. Its role is deliberately narrow: it does not define a new semantic dynamical system; instead, it makes the structured interface tokens semantically anchored and more directly consumable by the language model.

The aligned interface is then consumed by a shared LLM under task-specific prompts for clip summary generation, ordered event description, and temporal question answering. The overall objective combines task losses and interface-alignment losses:

$$\mathcal{L} = \mathcal{L}_{\mathrm{sum}} + \lambda_{\mathrm{evt}}\mathcal{L}_{\mathrm{evt}} + \lambda_{\mathrm{qa}}\mathcal{L}_{\mathrm{qa}} + \lambda_{\mathrm{clip}}\mathcal{L}_{\mathrm{clip-align}} + \lambda_{\mathrm{evt-align}}\mathcal{L}_{\mathrm{evt-align}} + \lambda_{\mathrm{bnd}}\mathcal{L}_{\mathrm{bnd}}, \tag{24}$$

where event-related terms are applied whenever localized annotations are available. Training follows two stages: we first learn the radar interface on physics-aware synthetic radar-language data, and then adapt the interface and the language model on real-world radar data using the same interface definition and task family.

The dual-scan encoder is linear in the serialized sequence length, and the final interface compresses the radar stream into a small number of clip and event tokens before language decoding.

3 Experiments

Our experiments evaluate RILA as a radar-native structured interface rather than a single captioning head. We therefore test the same interface on three language tasks—clip summary generation, ordered event description, and temporal QA—under controlled radar-native comparisons. The key questions are whether the interface improves real-world radar understanding after synthetic pretraining and which structural components are essential.

3.1 Datasets and Interface Supervision

We pretrain RILA on a physics-aware synthetic radar-language corpus built from articulated human motion and aligned motion-language resources [22, 25, 26]. The synthetic corpus provides both clip-level summaries and temporally ordered event annotations, and is used to learn the tokenizer, the dual-order encoder, the event proposal head, and the interface-language alignment layers. We then adapt and evaluate the model on a real-world benchmark collected with a TI IWR6843ISK radar and a DCA1000EVM capture board [32, 31]. The benchmark contains 500 minutes of data across four indoor environments; synchronized RGB videos are used only for annotation and are withheld at inference time. All clips are paired with clip-level summaries, while a dense core subset additionally provides ordered non-overlapping event spans and the derived temporal-QA benchmark. We use subject-disjoint train/validation/test splits and also define an unseen-room subset. Detailed dataset statistics, annotation breakdowns, and robustness subsets are provided in the appendix.

3.2 Evaluation Tasks and Metrics

We evaluate the shared interface on three tasks. For clip summary generation, the model outputs a single textual summary and we report ROUGE-L and CIDEr [20, 34]. For ordered event description, the model predicts event spans and localized descriptions; we report Event-F1@0.5 and segment-level

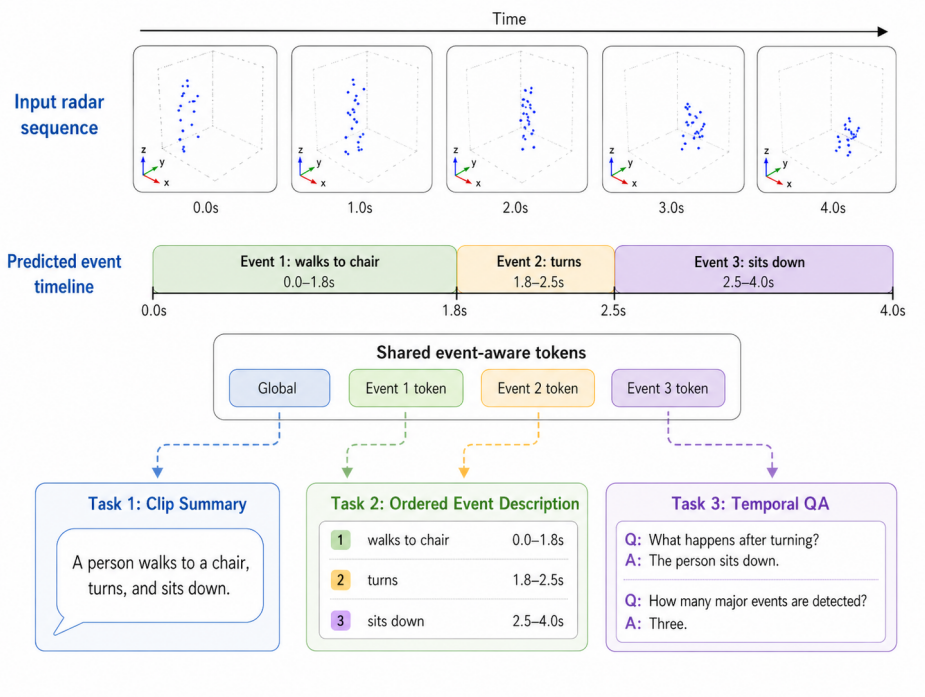

Figure 2: One radar sequence, three language tasks. The same event-aware interface supports clip-level summary generation, ordered event description, and temporal question answering from a shared set of global and event tokens.

CIDEr after Hungarian matching based on temporal overlap [12, 34]. For temporal QA, we build a benchmark from the event annotations and report exact-match accuracy (QA-Acc). This metric set is intentionally compact but task-aligned: summary metrics test global semantic compression, event metrics test whether the interface exposes localized temporal structure, and QA accuracy tests whether the same token layer supports structured querying. Boundary-sensitive evaluation and per-category temporal-QA results are reported in Appendix C.

3.3 Baselines and Comparison Protocol

We restrict all main-table claims to controlled radar-native baselines, because the contribution lies in the proposed interface rather than differences in upstream sensing pipelines or external modality bridges.

We compare RILA against five baselines: Spectrogram+LSTM operating on time–frequency radar representations; PointNet++ + Decoder using a frame-wise point-set encoder with the same downstream language decoder [27]; PointMamba + Decoder using a point-cloud Mamba encoder under the same decoder [19]; PointMamba + EventPool, which attaches RILA's frame pooling, boundary head, event-token projection, SGSA objective, prompts, and LLM backend to PointMamba; and a Discrete Radar Interface baseline inspired by prior radar-to-language tokenization strategies. All controlled baselines use the same radar input source, identical Top-K selection, the same hardware-consistent normalization, the same decoder family, and the same task prompts. Unless otherwise stated, all methods are trained with the same two-stage schedule consisting of synthetic interface pretraining followed by real-world adaptation. Implementation settings, task prompt templates, and runtime measurements are provided in the appendix.

The controlled main-table comparisons use Qwen3.5, with a frozen `all-mpnet-base-v2` encoder for alignment. We use $K = 128$, Hilbert resolution $k = 8$, Doppler scale $\lambda_v = 1$, a validation-fixed boundary threshold of 0.5, and a sequence-dependent event count capped at $M_{\max} = 8$. Further implementation details are given in Appendix B.

Table 1: Main real-world comparison under matched radar inputs, Top-K budget, normalization, decoder family, and task prompts. *PointMamba + EventPool* attaches RILA's event-pooling, SGSA, prompt, and LLM stack to a PointMamba encoder.

Method	Summary R-L ↑	Summary CIDEr ↑	Event-F1 @0.5 ↑	Seg. CIDEr ↑	QA Acc. ↑
Spectrogram+LSTM	23.1	2.6	17.9	2.0	38.2
PointNet++ + Decoder	27.3	4.4	23.5	3.7	43.3
PointMamba + Decoder	29.7	6.9	33.9	3.9	48.8
PointMamba + EventPool	30.3	6.9	35.6	4.1	50.6
Discrete Radar Interface	32.2	7.2	41.2	4.3	52.7
RILA (ours)	**34.3**	**8.1**	**44.8**	**5.1**	**61.7**

Table 2: Core ablations of the proposed interface on the real-world dense test split. Each row removes or replaces one major design choice relative to the full model. The table is designed to distinguish gains from (i) complementary serialization orders, (ii) explicit event-aware token abstraction, and (iii) structured interface-language alignment.

Configuration	Summary R-L ↑	Summary CIDEr ↑	Event-F1 @0.5 ↑	Seg. CIDEr ↑	QA Acc. ↑
Full RILA	**34.3**	**8.1**	**44.8**	**5.1**	**61.7**
w/o flow-consistency order	33.1	6.7	33.9	3.9	53.4
Bidirectional same-order scan	33.4	7.3	38.4	4.2	57.0
w/o event-aware tokens	32.9	6.9	30.0	3.7	48.6
w/o SGSA	33.4	7.1	37.7	4.4	55.9
Discrete Radar Interface	32.2	7.2	41.2	4.3	52.7

3.4 Main Results

Figure 2 illustrates a representative example of how the same interface supports clip summary, ordered event description, and temporal QA from one radar sequence. Table 1 reports the main quantitative comparison on the real-world test split. We intentionally center the main table on real-world evaluation, because the main claim of the paper is not that the interface fits synthetic radar better, but that it provides a stronger and more LLM-readable representation for real radar understanding under controlled radar-native comparisons.

The results support three observations. First, RILA achieves the strongest performance on all five metrics. Relative to the strongest controlled baseline, i.e., the Discrete Radar Interface, RILA improves summary ROUGE-L from 32.2 to 34.3 and summary CIDEr from 7.2 to 8.1, showing that event-aware radar structuring does not come at the expense of global semantic compression. On the overlapping clip-summary task, our independent reimplementation of RadarLLM [13] obtains ROUGE-L/CIDEr of $30.5 \pm 0.4/6.1 \pm 0.3$ over three training seeds, compared with 34.3/8.1 for RILA's fixed checkpoint. A shared-Qwen3.5 interface comparison further supports this trend (Appendix E, Table 21). Second, the gains are larger on event-level and structured-querying metrics than on summary metrics. In particular, RILA improves Event-F1@0.5 from 41.2 to 44.8, Seg-CIDEr from 4.3 to 5.1, and QA-Acc from 52.7 to 61.7 over the discrete baseline. This pattern suggests that the proposed interface is not merely improving generic decoding fluency; its main effect is to expose a more structured representation of localized motion to the language model. Third, the same interface also remains clearly stronger than continuous radar encoders without explicit interface design. Compared with PointMamba + Decoder, RILA improves summary ROUGE-L by 4.6 points, Event-F1@0.5 by 10.9 points, Seg-CIDEr by 1.2 points, and QA-Acc by 12.9 points.

Taken together, these results support the central thesis of the paper: localized radar-language performance depends on the coupling between radar-native encoding and event-aware abstraction. The qualitative example in Fig. 2 is consistent with this trend: the same shared token layer supports both compact global summaries and temporally localized outputs from one radar sequence.

Table 3: Interface comparison with and without synthetic radar-language pretraining under the matched real-world protocol.

Interface	Training	E-F1@0.5 ↑	S-CIDEr ↑	QA ↑
RILA	Synthetic + real	44.8	5.1	61.7
Discrete Interface	Synthetic + real	41.2	4.3	52.7
RILA	Real-only	35.4	4.0	55.3
Discrete Interface	Real-only	32.4	3.5	48.4

3.5 Core Ablations

Table 2 isolates the contribution of the main design choices. The variant *w/o flow-consistency order* keeps only the Hilbert geometry order, *bidirectional same-order scan* replaces the complementary dual-order design with forward/backward scans over a single order, *w/o event-aware tokens* removes the explicit event-token abstraction, *w/o SGSA* removes the structured interface-language alignment objective, and *Discrete Radar Interface* replaces the proposed continuous structured interface with a discrete token interface.

Three findings stand out. First, removing the flow-consistency order causes the largest encoder-side degradation among the continuous-interface variants. Summary CIDEr drops from 8.1 to 6.7, Event-F1@0.5 drops from 44.8 to 33.9, Seg-CIDEr drops from 5.1 to 3.9, and QA-Acc drops from 61.7 to 53.4. This confirms that geometry-preserving serialization alone is insufficient; the second, motion-consistency order is a major source of temporal structure for the interface.

Second, bidirectional scanning over the same order recovers part of the loss but remains consistently below the full model. Compared with Full RILA, the bidirectional same-order variant reaches 33.4/7.3 on summary ROUGE-L/CIDEr, 38.4 on Event-F1@0.5, 4.2 on Seg-CIDEr, and 57.0 on QA-Acc. This result is important because it shows that the gain does not come merely from scanning one sequence in two directions. Instead, the stronger performance of the full model comes from using two semantically different orders that encode complementary radar structure.

Third, removing the event-aware token abstraction produces the largest degradation on localized and query-based tasks. The *w/o event-aware tokens* variant drops to 30.0 Event-F1@0.5, 3.7 Seg-CIDEr, and 48.6 QA-Acc, while its summary metrics remain at 32.9 ROUGE-L and 6.9 CIDEr. This confirms that event-aware tokens are necessary for localized radar-language access. However, Table 1 shows that they are not sufficient by themselves: PointMamba + EventPool uses the same event-pooling, SGSA, prompts, and LLM stack, yet remains below Full RILA on Event-F1@0.5 and QA-Acc. Finally, removing SGSA causes a smaller but still consistent degradation, especially on Event-F1@0.5 and QA-Acc (37.7 and 55.9, respectively), while summary metrics remain relatively closer to the full model. This indicates that SGSA mainly improves the semantic organization of interface tokens rather than only generic text fluency.

Overall, the ablation results support the intended division of roles in RILA: the dual-order encoder preserves complementary radar structure, the event-aware token layer makes that structure accessible to the language model, and SGSA aligns these interface tokens with localized semantics. The localized-task gains therefore should be attributed to the coupling between radar-native encoding and event-aware abstraction, not to event tokens alone.

Synthetic Pretraining versus Interface Design. To isolate the contribution of interface design from synthetic pretraining, Table 3 compares both interfaces with and without synthetic pretraining; matched training details are provided in Appendix C.4. Without synthetic pretraining, RILA achieves 35.4/4.0/55.3 Event-F1@0.5/Seg-CIDEr/QA-Acc, compared with 32.4/3.5/48.4 for Discrete. Both interfaces benefit from synthetic pretraining, but RILA retains a 3.0/0.5/6.9 advantage under real-only training. Thus, synthetic pretraining alone does not explain RILA's interface advantage.

3.6 Discussion

The empirical results suggest that the main contribution of RILA lies in *what* the language model receives from radar rather than only in *how* the encoder is parameterized. If the gains mainly came from a stronger decoder or generic supervision, one would expect the improvements to concentrate

on clip-level summary metrics. Instead, the largest gains appear on ordered event description and temporal QA, where the LLM must access localized temporal structure. This pattern matches the intended role of RILA: preserving radar-native kinematics, exposing event-aware structure, and allowing one language backend to reuse the same interface across tasks.

This also clarifies the distinction between RILA and nearby component families. Point-cloud SSMs such as PointMamba and Point Cloud Mamba study how to serialize unordered geometry for point-cloud perception. RILA uses serialization for a different target: constructing an LLM-facing radar interface. The Hilbert order preserves sparse phase-space locality in (x, y, z, v), while the flow-consistency order preserves short-horizon motion continuity for event-token construction. Similarly, video event-token methods usually compress dense visual embeddings, whereas RILA must induce event structure from sparse and discontinuous radar detections with Doppler and reflection artifacts. Thus, the novelty is not that Mamba, Hilbert ordering, or event tokens are individually new, but that they are coupled into a radar-native interface for language reasoning.

A second implication is that the interface layer should be viewed as a reusable representation rather than a task-specific captioning component. The same tokenization and encoding stages support clip-level summarization, localized event description, and structured temporal querying without changing the underlying radar representation. In this sense, RILA is not only a stronger radar encoder, but a step toward making sparse mmWave radar compatible with shared language interfaces.

Boundary Sensitivity and Radar Robustness. We further evaluate boundary localization and robustness under controlled radar-input perturbations. With frozen checkpoints, RILA obtains 42.4 Event-F1 and 59.1 QA under 20% point dropout, compared with 44.8/61.7 on clean data. Under +50% injected clutter, performance decreases to 36.9/52.4, revealing a more challenging failure regime. Boundary-threshold and oracle-span diagnostics quantify localization sensitivity and remaining headroom (Appendix C.2); additional corruption and fast-motion comparisons are reported in Appendix D.1.

On the same-device unseen-room subset, RILA achieves 43.1/4.6/60.1 Event-F1/Seg-CIDEr/QA, versus 39.3/4.2/53.7 for the Discrete Interface. The room, participants, and sessions are held out, sensor placement differs, and no target-room tuning is used. This evaluates a joint participant/session/room/placement shift under fixed hardware (Appendix C.6).

3.7 Limitations

The current study has several limitations. First, the empirical scope is restricted to single-person radar-language understanding under one sensing setup. This is a meaningful first step because even the single-person setting requires the interface to preserve localized temporal structure across summary generation, ordered event description, and temporal QA. However, the current results should not be interpreted as solving multi-person interaction understanding, cross-device transfer, or all radar deployment conditions. Second, the temporal QA benchmark is derived from event annotations and therefore evaluates structured access to event-aware interface tokens rather than unrestricted open-domain reasoning. This is appropriate for validating the interface, but the QA results should be interpreted as evidence of temporal structure access, not as a claim of general-purpose reasoning from radar. Finally, although mmWave radar is more privacy-preserving than RGB sensing, language outputs can still reveal sensitive behavioral information; deployment should therefore require task-specific safeguards.

4 Conclusion

We introduced **RILA**, a radar-native structured interface that converts sparse mmWave point clouds into LLM-readable global and event-aware tokens. Across clip summary, ordered event description, and temporal QA, the results support the central claim that the representation handed to the LLM is a key design object: preserving radar kinematics and event structure makes sparse radar observations more descriptive, temporally localized, and queryable through a shared language backend. Future work will extend this interface toward multi-person activity understanding, broader sensing conditions, and richer real-world evaluation.

Acknowledgments and Disclosure of Funding

This work was supported by internal research funding from Waseda University.

References

[1] S. An, Y. Li, and U. Y. Ogras. mri: Multi-modal 3d human pose estimation dataset using mmwave, rgb-d, and inertial sensors. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:27414–27426, 2022.

[2] Qiming Cao, Hongfei Xue, Tianci Liu, Xingchen Wang, Haoyu Wang, Xincheng Zhang, and Lu Su. mmclip: Boosting mmwave-based zero-shot har via signal-text alignment. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, pages 184–197, 2024.

[3] A. Chen, X. Wang, S. Zhu, Y. Li, J. Chen, and Q. Ye. mmbody benchmark: 3d body reconstruction dataset and analysis for millimeter wave radar. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3501–3510, 2022.

[4] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

[5] F. Ding, Z. Luo, P. Zhao, and C. X. Lu. milliflow: Scene flow estimation on mmwave radar point cloud for human motion sensing. In *European Conference on Computer Vision (ECCV)*, pages 202–221. Springer, 2024.

[6] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14204–14213, 2021.

[7] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[8] B. Jiang, X. Chen, W. Liu, J. Yu, G. Yu, and T. Chen. Motiongpt: Human motion as a foreign language. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 20067–20079, 2023.

[9] L. Kang, Z. Li, X. Zhao, Z. Zhao, and T. Braun. St-pct: Spatial–temporal point cloud transformer for sensing activity based on mmwave. *IEEE Internet of Things Journal*, 11(6):10979–10991, 2023.

[10] Y. Kim and T. Moon. Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 13(1):8–12, 2016.

[11] Youngwook Kim and Hao Ling. Human activity classification based on micro-doppler signatures using a support vector machine. *IEEE Transactions on Geoscience and Remote Sensing*, 47(5):1328–1337, 2009.

[12] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[13] Zengyuan Lai, Jiarui Yang, Songpengcheng Xia, Lizhou Lin, Lan Sun, Renwen Wang, Jianran Liu, Qi Wu, and Ling Pei. Radarllm: Empowering large language models to understand human motion from millimeter-wave point cloud sequence. *arXiv preprint arXiv:2504.09862*, 2025.

[14] Shenglei Li and Reiko Hishiyama. Counting and tracking people to avoid from crowded in a restaurant using mmwave radar. *IEICE TRANSACTIONS on Information and Systems*, 106(6):1142–1154, 2023.

[15] Shenglei Li and Reiko Hishiyama. An indoor people counting and tracking system using mmwave sensor and sub-sensors. *IFAC-PapersOnLine*, 56(2):7096–7101, 2023.

[16] Shenglei Li and Tomoji Kishi. A dual mmwave radar system using density-based sequential clustering. In *Pacific Rim International Conference on Artificial Intelligence*, pages 281–289. Springer, 2025.

[17] Shenglei Li, Tengfei Shao, Haoran Luo, and Tomoji Kishi. DMTC: A collaborative dual mmWave radar system for smart spaces. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 19622–19626, 2026.

[18] Zechen Li, Shohreh Deldari, Linyao Chen, Hao Xue, and Flora D. Salim. Sensorllm: Aligning large language models with motion sensors for human activity recognition. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 354–379, 2025.

[19] Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *Advances in neural information processing systems*, 37:32653–32677, 2024.

[20] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[21] Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, Bing Wang, Changhao Chen, John A Stankovic, Niki Trigoni, and Andrew Markham. See through smoke: robust indoor mapping with low-cost mmwave radar. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 14–27, 2020.

[22] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[23] Jeffrey A Nanzer. A review of microwave wireless techniques for human presence detection and classification. *IEEE Transactions on Microwave Theory and Techniques*, 65(5):1780–1794, 2017.

[24] Zongshang Pang, Mayu Otani, and Yuta Nakashima. Measure twice, cut once: Grasping video structures and event semantics with llms for video temporal localization. *arXiv preprint arXiv:2503.09027*, 2025.

[25] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.

[26] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. Babel: Bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 722–731, 2021.

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[28] Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns. *IEEE Sensors Journal*, 20(17):10032–10044, 2020.

[29] John Skilling. Programming the hilbert curve. In *AIP Conference Proceedings*, volume 707, pages 381–387. American Institute of Physics, 2004.

[30] Yangfan Sun, Renlong Hang, Zhu Li, Mouqing Jin, and Kelvin Xu. Privacy-preserving fall detection with deep learning on mmwave radar signal. In *2019 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2019.

[31] Texas Instruments. Dca1000evm: Real-time data-capture adapter for radar sensing evaluation modules. `https://www.ti.com/tool/DCA1000EVM`, 2025. Accessed: 2026-4-01.

[32] Texas Instruments. Iwr6843isk: Iwr6843 evaluation module. `https://www.ti.com/tool/IWR6843ISK`, 2025. Accessed: 2026-4-01.

[33] Bram van Berlo, Amany Elkelany, Tanir Özçelebi, and Nirvana Meratnia. Millimeter wave sensing: A review of application pipelines and building blocks. *IEEE Sensors Journal*, 21(9):10332–10368, 2021.

[34] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015.

[35] Mengyue Wang, Shuo Chen, Kristian Kersting, Volker Tresp, and Yunpu Ma. Metok: Multistage event-based token compression for efficient long video understanding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18881–18895, 2025.

[36] R. Xu, X. Wang, T. Wang, Y. Chen, J. Pang, and D. Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision (ECCV)*, pages 131–147. Springer, 2024.

[37] H. Xue, Y. Ju, C. Miao, Y. Wang, S. Wang, A. Zhang, and L. Su. mmmesh: Towards 3d real-time dynamic human mesh construction using millimeter-wave. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 269–282, 2021.

[38] Sen Yang, Jiaming Liu, Ray Zhang, Mingjie Pan, Zhuozhuo Guo, Xiaoqi Li, Zehui Chen, Peng Gao, Hongsheng Li, Yandong Guo, et al. Lidar-llm: Exploring the potential of large language models for 3d lidar understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9247–9255, 2025.

[39] Jawad Yousaf, Alyssa Palmer, Maitha Shams, Zayna Wasma, Montasir Qasymeh, Taimur Hassan, Deepak Arora, and Mohammed Ghazal. mmwave radar-based noninvasive vehicle occupancy detection with insights from vision, rf, and physical sensors: A review. *IEEE Sensors Journal*, 25:18643–18661, 2025.

[40] Tao Zhang, Haobo Yuan, Lu Qi, Jiangning Zhang, Qianyu Zhou, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Point cloud mamba: Point cloud learning via state space model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 10121–10130, 2025.

[41] Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. mid: Tracking and identifying people with millimeter wave radar. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 33–40. IEEE, 2019.

Appendix

A Benchmark, Supervision, and Annotation Protocol

This appendix documents the data resources and annotation procedures used to train and evaluate RILA. The study relies on two complementary sources of supervision: a large synthetic radar-language corpus used to learn the interface under structured supervision, and a real-world mmWave benchmark used to evaluate transfer to measured data. The real benchmark further includes a densely annotated core subset that supports event-level evaluation and temporal question answering. Together, these components define the empirical scope of the paper.

Synthetic interface-pretraining corpus. The synthetic corpus is generated with the physics-aware radar simulation pipeline described in the main paper. Starting from articulated human motion resources and temporally aligned motion semantics from AMASS, SMPL-X, and BABEL [22, 25, 26], we generate sparse radar point-cloud sequences together with two levels of language supervision: a clip-level summary and an ordered sequence of non-overlapping event spans. The current corpus contains 13,308 synthetic radar-language sequences with a total duration of approximately 45 hours. We use a 70/15/15 unseen-subject split for synthetic pretraining, validation, and held-out synthetic evaluation. These external assets are credited in the bibliography and are used in accordance with their respective licenses and terms of use.

Real-world benchmark. The real-world benchmark is collected using a TI IWR6843ISK mmWave radar together with a DCA1000EVM capture board [32, 31]. It spans 500 minutes across four indoor rooms and is organized as a subject-disjoint train/validation/test benchmark. Synchronized RGB videos are used only for annotation and are withheld at inference time. The full benchmark provides clip-level summaries, while a dense core subset additionally provides temporally ordered event spans and derived temporal-QA supervision. Because the recordings involve privacy-sensitive human activity data, the real benchmark is non-public at submission time. The real-world recordings were collected under a fixed activity protocol with informed participant involvement, including author self-participation, and no crowdsourcing platform or external paid labor was used.

Roles of the two data sources. The synthetic corpus and the real benchmark serve different purposes. The synthetic corpus supplies large-scale structured supervision for learning the tokenizer, dual-order encoder, event proposal head, and interface-language alignment layers. The real benchmark is then used to assess whether the resulting interface remains effective on measured radar data. The dense core subset should therefore be viewed as an event-aware evaluation layer within the real benchmark rather than as a separate standalone dataset.

The real benchmark uses two annotation layers. All clips are paired with clip-level summaries, while the dense core subset additionally receives temporally ordered event spans with short event descriptions. Temporal QA is not collected as an independent annotation task; instead, it is derived directly from the event annotations. This choice keeps the evaluation aligned with the central interface claim of the paper, namely whether the same token layer can support both global description and structured temporal access.

Event annotation policy. Each dense-core clip is segmented into a small number of monotonic, non-overlapping semantic phases. Annotators are instructed to place boundaries at semantic transitions rather than minor kinematic fluctuations, and very short fragments are merged into the nearest semantic parent event. Event descriptions are written as short verb-centered clauses such as "walks toward the chair," "turns," or "sits down." This form is sufficiently localized for event-level evaluation while remaining simple enough for consistent annotation.

Temporal-QA construction. The temporal-QA benchmark is deterministically derived from the event annotations rather than collected separately. We consider four families of questions: order, counting, transition, and coarse boundary / duration queries. The answers are generated from the event graph and temporal metadata associated with each dense-core clip. In this way, temporal QA evaluates whether the interface exposes reusable event structure rather than unrestricted open-domain reasoning.

Agreement and auditability. A subset of the dense core clips is independently re-labeled by multiple annotators and finalized through adjudication. At the current stage, the appendix emphasizes the annotation protocol and audit structure rather than detailed agreement statistics, because the

Table 4: Summary of the datasets and supervision layers used in this work.

Subset	Scale	Supervision	Primary role
Synthetic interface-pretraining corpus	13,308 sequences ($\approx$ 45 h)	Clip summaries + ordered event spans	Interface pretraining under dense structured supervision
Real benchmark (full)	500 min across 4 rooms	Clip-level summaries	Real-world adaptation and summary evaluation
Real dense core subset	Dense held-out subset of the full benchmark	Clip summaries + ordered event spans + temporal QA	Event-aware evaluation and temporal-QA analysis

Table 5: Annotation protocol and QA construction. The table specifies what is annotated, how quality is controlled, and how each supervision component is used in training or evaluation.

Component	Annotation unit	Quality-control rule	Audit signal	Downstream use
Clip summary	One short sentence per clip	Annotators describe the dominant activity sequence rather than frame-level detail	Spot-checked semantic consistency	Clip summary generation
Event spans	Monotonic, non-overlapping temporal intervals	Boundaries are placed at semantic transitions; very short fragments are merged	Adjudicated boundary consistency	Ordered event description; boundary-sensitive evaluation
Event text	One short clause per event	Descriptions remain local and concise	Event-tag / event-text consistency review	Localized language supervision
Temporal QA: order	Template-derived QA pair	Built directly from event order	Deterministic answer generation	Tests temporal ordering access
Temporal QA: counting	Template-derived QA pair	Built from the number of major events	Deterministic answer generation	Tests structured access to event count
Temporal QA: transition	Template-derived QA pair	Built from neighboring event relations	Deterministic answer generation	Tests whether event-aware tokens preserve local transitions
Temporal QA: boundary / duration	Template-derived QA pair	Built from event starts, ends, and coarse durations	Deterministic answer generation	Tests whether the interface exposes usable boundary structure
Agreement audit subset	Independently relabeled dense-core clips	Multi-annotator labeling followed by adjudication	Final labels used as reference	Annotation reliability and auditability

immediate goal is to make clear how the benchmark is constructed and how the supervision should be interpreted. The same annotation design is used consistently across ordered event description and temporal QA.

Taken together, these supervision layers allow the same interface to be evaluated from three complementary perspectives: global summarization, localized event grounding, and structured temporal access. This is important for the overall story of the paper, because RILA is not intended as a single captioning head, but as a reusable structured interface between sparse mmWave radar and a shared language model.

B Interface Realization and Reproducibility Details

This section records the implementation choices needed to interpret RILA as a controlled radar-native interface. We report the sensing setup, radar representation, token budget, serialization settings, language-backend protocol, training schedule, and baseline-matching rules. The goal is to make clear which factors are held fixed across comparisons without turning the appendix into a release note for a non-public benchmark.

Core implementation configuration. RILA operates on sparse mmWave point-cloud sequences collected or simulated under the same representation assumptions used throughout the paper. The real benchmark is collected with a TI IWR6843ISK sensor and a DCA1000EVM capture board. The interface consumes sparse radar point clouds rather than heatmap-like radar tensors. All controlled point-cloud baselines use the same Top-K radar point budget, hardware-consistent normalization, task prompts, and downstream language-backend protocol.

Table 6: Core implementation details for RILA and the controlled radar-native comparison protocol.

Item	Value	Notes
Radar hardware	TI IWR6843ISK + DCA1000EVM	Used for real-world benchmark collection.
Real benchmark scale	500 min across 4 rooms	Full real benchmark used for adaptation and evaluation.
Annotation stream	Synchronized RGB only for annotation	RGB is withheld at inference time.
Radar representation	Sparse mmWave point clouds	The interface is defined over point-cloud sequences rather than heatmap inputs.
Radar point budget	Top-$K = 128$	Identical across controlled point-cloud baselines.
Hilbert resolution	$k = 8$	Used for 4D geometry-order serialization.
Doppler scale	$\lambda_v = 1$	Used for normalized Doppler in the Hilbert ordering.
Interface prefix	$1 + M$ tokens	One global clip token plus M event-aware tokens.
Language backend	Qwen3.5 (main) Llama-4 (limited replication)	Main results and ablations use the same Qwen3.5 checkpoint.
Prompt protocol	Matched across controlled comparisons	The same radar-prefix format, task prompts, and decoding protocol are used for each controlled comparison.
Radar prefix handling	Learned continuous prefix embeddings	Interface tokens are projected into the language-model embedding space.
Alignment text side	Frozen MPNet (mean pooling)	Used for clip- and event-level interface-language alignment.
Projection layer	Learned linear projection layer(s)	Maps radar interface tokens into the LLM embedding space.
Normalization policy	Hardware-consistent kinematic normalization	Shared across controlled point-cloud comparisons.
Controlled-baseline fairness	Same radar input, decoder protocol, prompts, normalization, and Top-K	Keeps the comparison focused on the interface rather than backend changes.
Optimizer	AdamW	Used for reported training runs.
Learning-rate schedule	Cosine schedule	Shared schedule family across the two-stage training setup.
Stage 1	Synthetic interface pretraining, 300 epochs	Learns interface components on the synthetic radar-language corpus.
Stage 2	Real-world adaptation, 8 epochs	Adapts the interface and language backend protocol on the real benchmark.
Real-world batch size	4	Used for real-world adaptation.
Model selection	Best validation checkpoint	The validation-best checkpoint is used for reported results.

SGSA uses the frozen `sentence-transformers/all-mpnet-base-v2` text encoder with mean pooling, while learned projection layers map interface tokens into the language-model embedding space. All main results and ablations use the same Qwen3.5 checkpoint under matched adaptation and decoding settings. Llama-4 is used only for limited cross-backbone replication. Table 6 summarizes the implementation card used in the present submission. The key comparison principle is that decoder-side settings and radar input budgets are matched across controlled baselines, so the reported differences primarily reflect the radar interface rather than changes in decoder family, prompt design, or point budget.

Prompt templates. All controlled methods use the same task templates. The radar prefix is represented by learned continuous embeddings and is denoted as `<RADAR_PREFIX>`.

Complementary dual-order serialization. A central part of the interface is the use of two different serializations before sequence encoding. The geometry order preserves local neighborhoods in normalized kinematic space through a 4D Hilbert traversal, while the flow-consistency order groups locally coherent motion fragments across frames. Figure 3 provides a schematic view of these two paths.

The purpose of the dual-order design is not redundancy. The two branches expose different forms of radar structure to the interface before fusion. This is also why the structural ablations in Appendix C are informative: the comparison between the full model, the single-order variants, and the bidirectional

Table 7: Task prompt templates used by the shared language backend.

Task	Prompt template
Clip summary	`<RADAR_PREFIX>` `Describe the human activity in one concise sentence.`

Ordered events	`<RADAR_PREFIX>`
	`List the events in chronological order. For each event, output:`
	`[start, end]: description.`

Temporal QA	`<RADAR_PREFIX>`
	`Question: {question}`
	`Answer:`

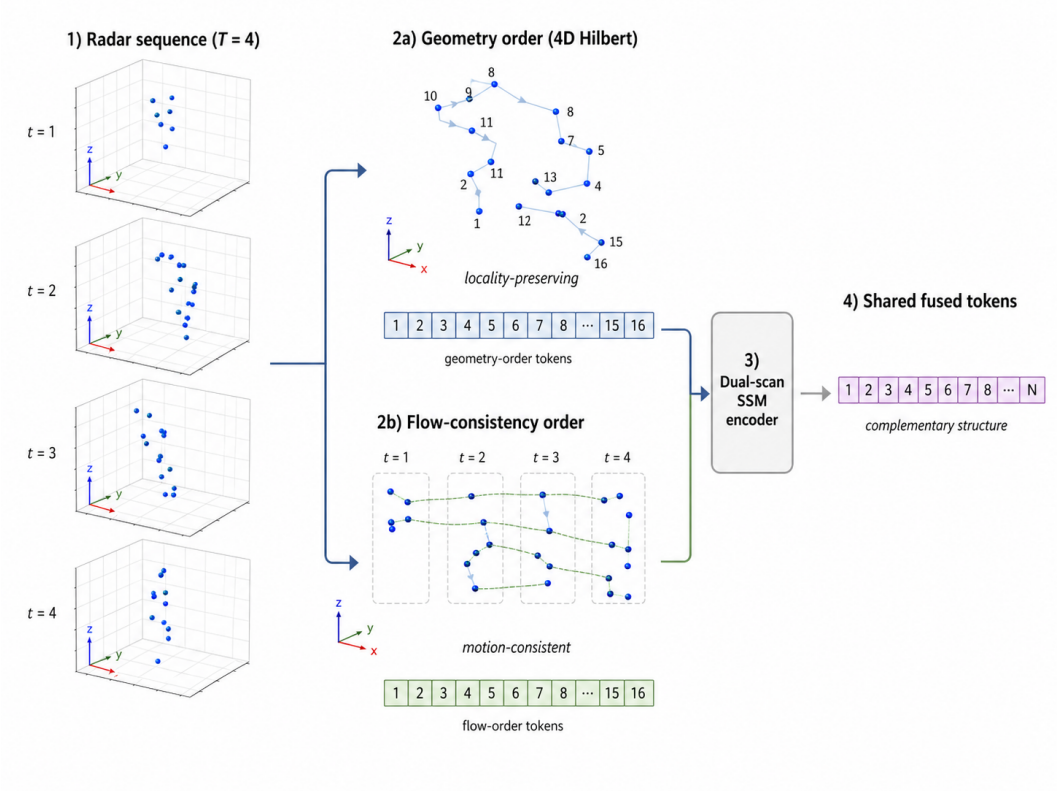

Figure 3: Complementary dual-order serialization. The geometry order preserves local kinematic neighborhoods through 4D Hilbert traversal, while the flow-consistency order groups locally coherent motion fragments across frames. The two orders provide complementary structure before dual-scan fusion.

same-order scan helps separate the value of complementary serializations from the value of simply scanning one sequence in multiple directions.

Event-aware token abstraction. The output of the dual-scan encoder is still a dense radar stream. What turns RILA into an interface layer is the final abstraction step, which pools frame-level representations into one global clip token and a small number of event-aware tokens. Figure 4 illustrates this process.

This abstraction step explains why the same interface can support summary generation, ordered event description, and temporal QA with a shared language backbone. The clip token provides global context, while the event-aware tokens expose localized temporal structure. In that sense, the interface is not only a stronger radar encoder followed by a projector. It is a staged transformation from sparse radar observations to a compact token layer designed for language reuse.

Interpretation of the implementation choices. The implementation details in this section should therefore be read together rather than in isolation. The sparse point-cloud input, the dual-order

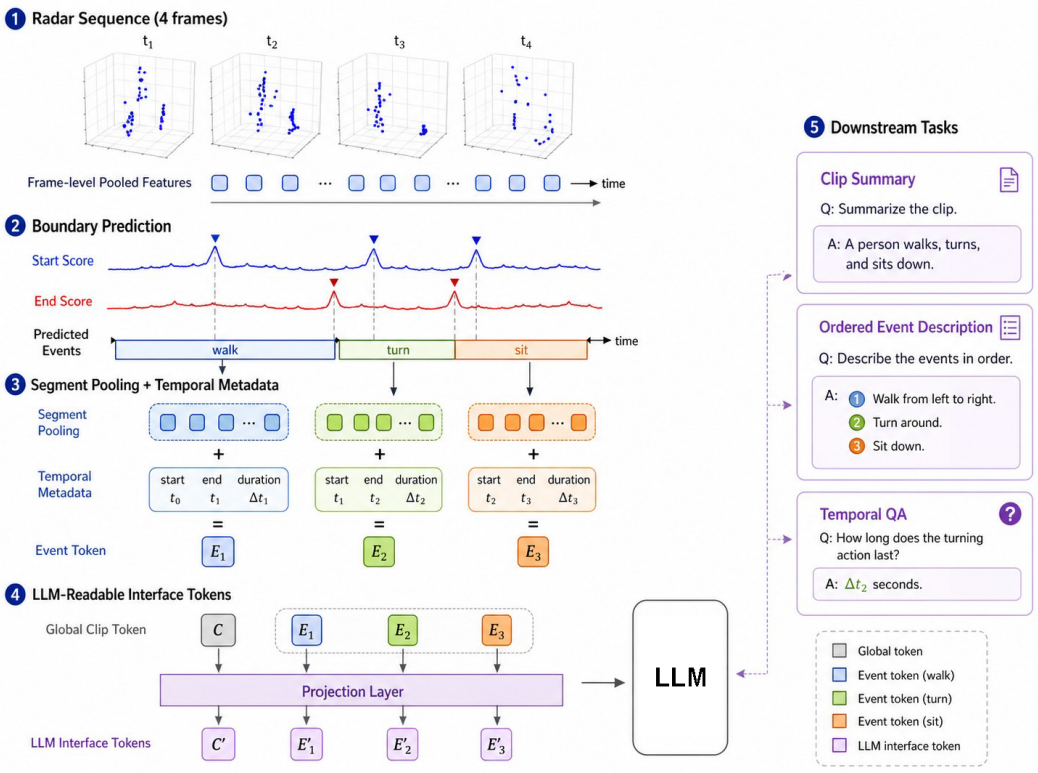

Figure 4: Event-aware token abstraction. Frame-level radar features are segmented into event spans, pooled with temporal metadata, and projected into LLM-readable interface tokens consisting of one global clip token and multiple event tokens. These structured tokens are then shared across downstream language tasks. Illustrated event spans are schematic; inference spans are predicted from radar.

serialization, the dual-scan state-space encoder, the event-aware token abstraction, and the shared decoder all serve the same goal: to make sparse mmWave radar sequences accessible to the language model through a token structure that is both compact and semantically organized. The controlled-baseline protocol in Table 6 is important precisely because it holds the decoder-side setting fixed, allowing the empirical comparisons to be interpreted as comparisons between interfaces rather than between unrelated multimodal pipelines.

C Extended Quantitative Evaluation

The main text establishes the central empirical claim of the paper through a compact set of interface-level comparisons. This appendix expands that analysis in a form that is easier to audit. We proceed in two steps. We first restate the real-world comparison and clarify how localized errors should be interpreted. We then extend the ablation analysis so that the contribution of each part of RILA can be examined more directly, including the structure of the interface itself, the role of alignment and temporal supervision, the sensitivity to serialization and radar point budgets, and the extent to which event-aware gains remain visible under room and task shift.

C.1 Full Real-World Metrics

Table 8 presents the full real-world comparison in the five primary metrics used throughout the paper. This view makes it possible to verify whether RILA improves one narrow output mode or whether it consistently strengthens the interface across summary generation, localized event description, and temporal QA.

Table 8: Full real-world metrics on the controlled radar-native benchmark.

Method	Summary R-L ↑	Summary CIDEr ↑	Event-F1@0.5 ↑	Seg-CIDEr ↑	QA-Acc ↑
Spectrogram+LSTM	23.1	2.6	17.9	2.0	38.2
PointNet++ + Decoder	27.3	4.4	23.5	3.7	43.3
PointMamba + Decoder	29.7	6.9	33.9	3.9	48.8
PointMamba + EventPool	30.3	6.9	35.6	4.1	50.6
Discrete Radar Interface	32.2	7.2	41.2	4.3	52.7
RILA (ours)	**34.3**	**8.1**	**44.8**	**5.1**	**61.7**

The comparison shows a consistent pattern. RILA ranks first on all five metrics. Relative to the strongest controlled baseline, namely the Discrete Radar Interface, RILA improves summary ROUGE-L from 32.2 to 34.3 and summary CIDEr from 7.2 to 8.1. The gains are larger on the structured tasks: Event-F1@0.5 rises from 41.2 to 44.8, Seg-CIDEr from 4.3 to 5.1, and QA-Acc from 52.7 to 61.7. The comparison with PointMamba + Decoder shows the same trend. While PointMamba is already much stronger than PointNet++ on the real benchmark, the gap between PointMamba and RILA is still substantial on event-aware description and temporal QA. This suggests that the main advantage of RILA lies not only in stronger sequence modeling, but in how the radar stream is organized before it reaches the language model.

Overall, Table 8 supports the same interpretation as the main paper: the strongest gains of RILA appear when the output requires localized temporal structure rather than only a single global description. This is consistent with the claim that the interface layer, rather than only the decoder, is the main object of improvement.

C.2 Boundary-Sensitive Evaluation and Failure Taxonomy

The event-aware evaluation in the main paper is intentionally compact. To interpret those results more precisely, Table 9 records the boundary-sensitive metrics and the failure taxonomy used in the appendix. The first six rows define the quantitative terms used to audit temporal structure, while the remaining rows describe the most important qualitative failure modes.

This distinction matters because event-level errors do not all have the same meaning. A model may preserve a plausible global summary while failing to place event boundaries accurately. Likewise, an event description may remain locally plausible while appearing in the wrong temporal order. In the appendix, we therefore use the terms in Table 9 as the common vocabulary for interpreting additional results and case-based analysis.

Boundary-Threshold Sensitivity. To assess localization sensitivity, we vary only the boundary decision threshold while keeping the trained checkpoint and other decoding settings fixed.

Across thresholds 0.40–0.60, all three metrics vary by at most 0.8 points, indicating a relatively stable region around the validation-selected threshold of 0.50.

Oracle-Span Diagnostic. Replacing radar-predicted event spans with annotation spans while freezing all other components increases Event-F1@0.5/Seg-CIDEr/QA from 44.8/5.1/61.7 to 49.5/5.8/64.1. This diagnostic quantifies the remaining localization headroom. The oracle spans are not used in the main evaluation protocol.

C.3 Structural Interface Ablations

Table 11 isolates the main structural components of the interface. Removing the flow-consistency order causes the largest degradation among the continuous-interface variants. Summary CIDEr drops from 8.1 to 6.7, Event-F1@0.5 drops from 44.8 to 33.9, and QA-Acc drops from 61.7 to 53.4. This indicates that geometry-preserving Hilbert traversal alone is not sufficient for event-aware radar-language modeling; the second order contributes substantial temporal organization.

The bidirectional same-order scan recovers part of this loss but remains below the full model. This separates the value of complementary serializations from the value of simply scanning one sequence twice. The event-aware token ablation shows a different role: without explicit event tokens, summary

Table 9: Boundary-sensitive metrics and failure taxonomy used to audit the event-aware interface.

Type	Name	Definition / counting rule	Diagnostic meaning
Metric	Boundary-P	Fraction of predicted boundaries that fall within a fixed tolerance of a reference boundary (e.g., ± 0.3s)	High precision indicates few spurious event cuts and low over-segmentation.
Metric	Boundary-R	Fraction of reference boundaries recovered within the same tolerance window	High recall indicates low under-segmentation and fewer missed event transitions.
Metric	Boundary-F1	Harmonic mean of Boundary-P and Boundary-R	Compact boundary-quality score.
Metric	Segment mIoU	Mean temporal IoU after matching predicted and reference event spans	Measures overlap quality of localized intervals.
Metric	Event-F1@0.3	Event correctness under a coarse temporal-overlap threshold	Captures rough recovery of event structure.
Metric	Event-F1@0.5	Event correctness under a stricter temporal-overlap threshold	Main-paper event metric; emphasizes more precise grounding.
Failure	Over-segmentation	Too many short predicted events for one reference event	Usually lowers Boundary-P and fragments event descriptions.
Failure	Under-segmentation	Multiple reference events merged into one predicted event	Usually lowers Boundary-R and Event-F1.
Failure	Boundary drift	Correct event count but shifted starts or ends	Often hurts segment mIoU and stricter Event-F1 more than summary quality.
Failure	Order inversion	Localized descriptions are plausible but appear in the wrong temporal order	Often leaves summary metrics relatively intact while hurting transition and order QA.
Failure	QA inconsistency	Summary or event descriptions look plausible but contradict temporal QA answers	Indicates that the shared interface tokens are not being reused consistently across tasks.

Table 10: Boundary-threshold sensitivity on the real-world test split using a fixed checkpoint.

Threshold	Boundary-F1	Event-F1@0.5	QA
0.30	68.9	42.9	60.2
0.40	71.6	44.1	61.2
0.50	72.4	44.8	61.7
0.60	71.6	44.4	61.0
0.70	69.1	42.8	59.7

quality decreases moderately, whereas Event-F1@0.5, Seg-CIDEr, and QA-Acc decline much more strongly. This demonstrates that event-aware tokens are necessary for localized access, but not sufficient by themselves. The PointMamba + EventPool control further shows that attaching the same event-pooling stack to a generic point-cloud SSM improves over PointMamba + Decoder but remains below Full RILA on localized metrics.

The additional rows confirm that the remaining structural components are also non-trivial. Removing temporal metadata causes a visible but smaller drop than removing event-aware tokens entirely, which suggests that the token abstraction itself matters more than temporal metadata alone. Removing gated fusion similarly weakens performance but does not collapse it, indicating that fusion is beneficial but not the sole source of structure. Finally, the flow-consistency-only variant remains substantially below the full model, which supports the view that geometry and motion continuity are complementary rather than interchangeable.

Selective Step-Size Parameterization. We compare the full selective step with two matched controls. The content-only variant sets $\gamma_p = \gamma_v = 0$, while the constant-learned variant replaces the input-dependent step with a learned positive constant. Both controls retain the same serialization branches, fusion, event abstraction, and losses, with matched initialization policy, training budget, and validation criteria.

The full parameterization improves Event-F1, overall QA, and transition QA over both controls. These results support the contribution of explicit radar-kinematic modulation beyond content-dependent step selection alone.

Table 11: Structural interface ablations on the real-world dense test split.

Configuration	R-L ↑	CIDEr ↑	E-F1@0.5 ↑	S-CIDEr ↑	QA ↑
Full RILA	**34.3**	**8.1**	**44.8**	**5.1**	**61.7**
w/o flow-consistency order	33.1	6.7	33.9	3.9	53.4
Bidirectional same-order scan	33.4	7.3	38.4	4.2	57.0
w/o event-aware tokens	32.9	6.9	30.0	3.7	48.6
w/o SGSA	33.4	7.1	37.7	4.4	55.9
PointMamba + EventPool	30.3	6.9	35.6	4.1	50.6
Discrete Radar Interface	32.2	7.2	41.2	4.3	52.7
w/o temporal metadata	33.7	7.7	41.0	4.7	59.1
w/o gated fusion	33.4	7.4	38.9	4.3	57.8
flow-consistency order only	32.8	6.8	32.3	3.9	53.5

Table 12: Ablation of selective step-size parameterization. Transition QA reports accuracy on transition questions.

Configuration	CIDEr	E-F1@0.5	S-CIDEr	QA	Transition QA
Full RILA	8.1	44.8	5.1	61.7	68.9
Content-only	7.7	42.4	4.9	59.3	66.1
Constant learned	7.6	42.1	4.8	59.4	65.9

C.4 Alignment and Temporal-Supervision Ablations

Table 13 examines how different forms of alignment and temporal supervision shape the interface. Unlike the structural ablations above, which test the architecture of the interface itself, this table asks how the interface is organized during training and which supervision signals are most important for preserving localized temporal semantics.

The measured rows already show a clear hierarchy. Removing SGSA causes a consistent drop across all structured tasks: Event-F1@0.5 decreases from 44.8 to 37.7, Seg-CIDEr decreases from 5.1 to 4.4, and QA-Acc decreases from 61.7 to 55.9, while Boundary-F1 also drops from 72.4 to 69.3. This indicates that SGSA contributes mainly to the semantic organization of localized interface tokens rather than only to global text quality.

The clip-level and event-level alignment terms play different roles. Removing clip-level alignment leads to a relatively modest drop in CIDEr, from 8.1 to 7.6, while the event-sensitive metrics remain comparatively closer to the full model than in the other ablations. In contrast, removing event-level alignment leaves CIDEr at 7.9 but reduces Boundary-F1 to 67.9, Event-F1@0.5 to 41.1, and QA-Acc to 58.9. This difference suggests that clip-level alignment mainly stabilizes global semantics, whereas event-level alignment is more important for preserving localized temporal structure.

Explicit boundary supervision is also important even when the rest of the interface remains unchanged. Without the boundary head, Boundary-F1 falls from 72.4 to 63.0, with corresponding drops in Event-F1@0.5, Seg-CIDEr, and QA-Acc. The final two rows clarify the role of supervision scale. Summary-only training preserves a partially usable summary head but substantially weakens boundary quality, event localization, and QA, while removing synthetic pretraining causes a broad degradation across all structured metrics. Together, these results indicate that the final interface is not formed by a single loss term or a single training stage, but by the interaction of structured alignment, explicit temporal supervision, and synthetic interface pretraining.

Matched Real-Only Interface Control. To disentangle the effects of synthetic pretraining and interface design, we compare RILA and the Discrete Radar Interface with and without synthetic pretraining on the same subject-disjoint real-world test split.

The real-only experiments omit synthetic radar-language pretraining and use matched radar inputs, Top-K budget, normalization, language backend, prompts, decoding settings, and maximum real-data update budget. Checkpoints are selected using the same pre-specified validation criterion, without test-based selection. Neither selected real-only checkpoint corresponds to the final training step.

Table 13: Alignment and temporal-supervision ablations. Boundary-F1 is included to separate temporal-structure failures from purely language-side degradation.

Configuration	CIDEr ↑	B-F1 ↑	E-F1@0.5 ↑	S-CIDEr ↑	QA ↑
Full RILA	**8.1**	**72.4**	**44.8**	**5.1**	**61.7**
w/o SGSA	7.1	69.3	37.7	4.4	55.9
w/o clip-level alignment	7.6	71.2	42.5	4.8	59.2
w/o event-level alignment	7.9	67.9	41.1	4.8	58.9
w/o boundary head	7.4	63.0	39.2	4.6	57.7
summary-only training	6.9	49.7	31.8	3.7	50.6
no synthetic pretraining	7.2	59.4	35.4	4.0	55.3

Table 14: Sensitivity to Hilbert resolution and radar token budget.

Factor	Setting	CIDEr ↑	E-F1@0.5 ↑	S-CIDEr ↑	QA ↑
Hilbert resolution					
k	6	7.3	38.7	4.1	56.3
k	7	7.8	43.4	4.8	59.7
k	8	8.1	44.8	5.1	61.7
k	9	8.0	44.1	4.9	60.6
Radar token budget					
Top-K	64	6.9	38.5	4.3	56.9
Top-K	96	7.8	42.7	4.8	59.4
Top-K	128	8.1	44.8	5.1	61.7
Top-K	160	7.9	44.4	5.1	61.0

Synthetic pretraining improves both interfaces, while RILA remains ahead of Discrete under real-only training by 3.0 Event-F1, 0.5 Seg-CIDEr, and 6.9 QA points. These fixed-checkpoint differences are interpreted descriptively rather than as formal significance tests.

C.5 Sensitivity to Serialization Resolution and Token Budget

Table 14 studies two measured design choices: the Hilbert resolution k and the per-frame radar point budget Top-K. For the Hilbert resolution, performance improves from $k = 6$ to $k = 8$ and then flattens or slightly decreases at $k = 9$, suggesting that the chosen operating point balances local kinematic preservation and over-discretization. A similar pattern appears in the Top-K study: increasing the point budget from 64 to 128 yields consistent gains, especially on Event-F1@0.5 and QA-Acc, while increasing further to 160 produces only minor changes.

The 4D Hilbert order also depends on the relative scaling between Cartesian position and radial velocity. In the main experiments we set the Doppler scale to $\lambda_v = 1$, which treats normalized Doppler as comparable to each normalized spatial coordinate after ROI calibration. This choice is simple and stable in the current setup, but it is not a universal physical constant. Incorrect ROI bounds or Doppler scaling can create artificial adjacency or discontinuity in the 1D Hilbert order. We therefore treat adaptive velocity scaling as a limitation and leave broader cross-device scaling analysis to future work.

C.6 Generalization Attribution under Room and Task Shift

Table 15 asks a more specific question than the main paper: are the gains of RILA confined to in-domain generation quality, or do they persist when the interface must preserve event structure under shift? The first block already provides a strong signal. On the overall real-world test split, removing event-aware tokens causes Event-F1@0.5 to drop from 44.8 to 30.0, Seg-CIDEr to drop from 5.1 to 3.7, and QA-Acc to drop from 61.7 to 48.6. The gap between Full RILA and the Discrete Radar Interface is smaller but still substantial, especially on QA.

The unseen-room subset was collected in a separate room excluded from training and validation, with disjoint participants and sessions. The radar model and acquisition configuration remained unchanged, while sensor placement differed. All checkpoints and evaluation settings were fixed without target-room adaptation or tuning. The unseen-room results show that this pattern is not confined to the in-domain setting. Full RILA remains clearly ahead of both the discrete baseline and the version without event-aware tokens, with the gap remaining especially visible on QA. The order-only and transition-only QA slices strengthen the same conclusion. On these more structured

Table 15: Generalization attribution under room and task shift.

Slice	Variant	E-F1@0.5 ↑	S-CIDEr ↑	QA ↑
Overall real test	Full RILA	44.8	5.1	61.7
Overall real test	Discrete Radar Interface	41.2	4.3	52.7
Overall real test	w/o event-aware tokens	30.0	3.7	48.6
Unseen-room	Full RILA	43.1	4.6	60.1
Unseen-room	Discrete Radar Interface	39.3	4.2	53.7
Unseen-room	w/o event-aware tokens	29.8	3.5	44.5
Order-only QA	Full RILA	–	–	68.4
Order-only QA	Discrete Radar Interface	–	–	61.2
Order-only QA	w/o event-aware tokens	–	–	57.1
Transition-only QA	Full RILA	–	–	68.9
Transition-only QA	Discrete Radar Interface	–	–	54.2
Transition-only QA	w/o event-aware tokens	–	–	49.9

Table 16: Temporal QA accuracy (%) under controlled input interventions using a fixed checkpoint.

Input condition	Order	Count	Transition	Duration
Full RILA	68.4	65.2	68.9	45.6
Mismatched radar prefix	33.1	35.1	27.9	24.6
Permuted event-token order	38.5	60.7	42.1	49.9

subsets, the event-aware token layer continues to offer a clear advantage. This is consistent with the broader claim of the paper: the event-aware interface is not merely improving lexical fluency or global clip semantics, but preserving reusable temporal structure that remains useful under room shift and structured temporal querying.

C.7 Temporal QA Input Interventions

We examine how temporal QA depends on clip-matched radar information and predicted event-token ordering using the same frozen RILA checkpoint. Both intervention mappings were fixed before scoring, without using model outputs or correctness.

For the mismatched-prefix condition, each clip receives the complete radar prefix of another test clip under a one-to-one mapping with no self-matches, while questions and reference answers remain unchanged. For the order intervention, only the order of predicted event tokens is permuted; the global token, event-token contents and count, and question are fixed.

Mismatching the radar prefix substantially reduces accuracy across all four question types, indicating dependence on clip-matched radar information. Permuting event-token order mainly affects order and transition questions, while count accuracy changes less. Duration accuracy increases slightly, so no ordering-related conclusion is drawn for duration. These diagnostics concern temporal relations within the predefined event ontology.

D Stress-Relevant Evidence and Interface Compactness

This section reports controlled radar-input robustness diagnostics and summarizes the compactness of the interface exposed to the language model. The robustness results complement the structural ablations and parameter sensitivity analyses in Appendix C.3 and Appendix C.5, while the compactness audit characterizes the decoder-side representation.

D.1 Controlled Radar-Input Perturbations

We evaluate the frozen clean-data checkpoints under controlled point-cloud perturbations without corruption-specific retraining or adaptation. The same perturbed inputs are used for RILA and the Discrete Radar Interface.

Point dropout and clutter injection are applied before Top-K selection. Clutter returns are sampled from source-room no-human recordings, and sector masking removes one contiguous azimuth sector spanning 25% of the horizontal field of view. Stochastic perturbation results are averaged over five

Table 17: Robustness under controlled radar-input perturbations and on an unperturbed fast-motion subset. B-F1 denotes Boundary-F1 and E-F1 denotes Event-F1@0.5.

Condition	RILA B-F1	RILA E-F1	Discrete E-F1	RILA QA	Discrete QA
Clean	72.4	44.8	41.2	61.7	52.7
20% point dropout	69.9	42.4	39.6	59.1	50.0
40% point dropout	64.3	39.6	36.0	55.9	45.4
+25% injected clutter	67.5	41.4	38.8	57.1	49.1
+50% injected clutter	60.9	36.9	35.8	52.4	47.1
25% sector masking	63.4	39.5	36.1	56.7	48.2
Fast-motion quartile	65.2	40.4	35.6	57.5	48.7

Table 18: Representation-level compactness audit for the proposed interface.

Quantity	RILA value	Unit	Interpretation
Raw frame token budget	Top-$K = 128$	detections / frame	Fixes the point-cloud input budget used by all controlled point-cloud comparisons.
Input sequence size	$T \times K$	detections / clip	Number of sparse radar detections processed before interface compression.
Encoder scan count	2	branches	Geometry order and flow-consistency order are processed as complementary serializations.
Asymptotic encoder complexity	$O(2TK)$	scaling law	Captures the linear recurrent cost of the dual-scan interface encoder.
Global interface token count	1	token / clip	Provides a clip-level summary anchor for the shared language decoder.
Event interface token count	M	tokens / clip	Provides compact localized access to event-level temporal structure.
LLM prefix length	$1 + M$	tokens / clip	Decoder-side sequence length contributed by the radar interface.
Decoder class	Shared decoder-only causal LLM	protocol	Keeps summary generation, ordered event description, and temporal QA on the same language backend.
Alignment text side	Frozen text encoder	protocol	Keeps the semantic target space fixed during structured interface-language alignment.

fixed perturbation-generation seeds, with matched inputs across methods. These are not independent training runs.

The fast-motion subset comprises the top quartile of unperturbed test clips under a predefined radar-only motion score based on normalized Doppler, local displacement, and Doppler change. On this subset, removing the flow-consistency order reduces Event-F1/QA from 40.4/57.5 to 30.9/49.7.

RILA retains an advantage over the Discrete Radar Interface across the evaluated conditions, while severe clutter causes substantial degradation. These tests are controlled proxies for radar-input degradation rather than measured-multipath, cross-device, or deployment-level evaluations.

D.2 Interface Compactness Audit

Table 18 records the representation-level quantities that define the compactness of the proposed interface. The raw radar stream contains $T \times K$ frame-level detections after Top-K selection, whereas the language model receives one global clip token and M event-aware tokens. This reduction from frame-level detections to a short structured prefix is part of the interface design: the LLM is not asked to decode the entire sparse radar stream directly, but instead consumes compact global and localized tokens.

The predicted event count has a mean/median/p95/maximum of 3.6/3/6/8, with only 1.1% of clips reaching $M_{\max} = 8$. The mean radar-prefix length is 4.6 tokens, including the global token. On an NVIDIA A100 (80 GB), batch-1 interface/end-to-end latency is 31.8/358 ms for PointMamba+EventPool, 33.9/364 ms for the Discrete Interface, and 37.6/391 ms for RILA, under identical input preprocessing and decoding/output-length caps. RILA incurs approximately 9% end-to-end latency overhead relative to PointMamba+EventPool. Thus, interface compactness does not imply faster end-to-end execution.

Table 19: Positioning of RILA relative to nearby serialization and event-token methods. The table clarifies that RILA is evaluated as a radar-to-language interface rather than as a generic point-cloud SSM or a video-token compression module.

Method family	Primary design target	Difference from RILA
Point-cloud SSMs	Serialize unordered 3D geometry for point-cloud perception	RILA serializes radar phase-space and local motion for LLM-facing event tokens
Video event-token methods	Compress dense visual tokens or localize events in video	RILA induces events from sparse radar detections without dense RGB semantics
Discrete radar-language interfaces	Convert radar streams into discrete codes for language decoding	RILA keeps a structured continuous interface with global and event-aware tokens
RILA	Build a radar-native token layer for a shared LLM backend	Preserves kinematics, event structure, and task-sharable prefix tokens

E Benchmark Positioning, Modality Choice, and Ethical Scope

This section clarifies how the current benchmark and interface should be interpreted. The main paper focuses on the central interface claim and therefore avoids a long discussion of alternative radar representations, benchmark scope, and deployment-facing safeguards. These issues are nevertheless important for understanding what RILA is intended to solve and what remains outside the current evidence. In particular, this section discusses the choice of sparse point clouds as the interface substrate, the roles of the controlled Discrete Interface baseline and the independently reproduced RadarLLM comparison, and the privacy and release safeguards appropriate for a radar-language benchmark.

Relation to point-cloud SSMs and video event-token methods. Several nearby methods share surface-level components with RILA, but they optimize for different interface targets. Point-cloud SSMs such as PointMamba and Point Cloud Mamba study how to serialize unordered 3D geometry for efficient point-cloud perception. RILA instead uses serialization as part of a radar-to-language interface: the geometry order is defined over radar kinematic phase space, the flow-consistency order is tied to short-horizon motion continuity, and the final output is a compact LLM prefix rather than only a point-cloud feature sequence. Video event-token methods also differ in their substrate. They usually begin from dense visual tokens produced by RGB/video encoders, whereas RILA must infer event structure from sparse and discontinuous radar detections. For this reason, event-aware abstraction in RILA is coupled to radar-native kinematic tokenization and local motion consistency rather than treated as a generic token-pruning module.

Point clouds as the interface substrate. RILA is centered on sparse mmWave point clouds because the main claim of the paper is about converting compact radar detections into LLM-readable event-aware tokens. Point clouds make the kinematic content of radar observations explicit and allow the interface to operate directly on tuples of the form (x, y, z, v, σ). This is particularly convenient for the present design, because dual-order serialization, motion-consistency grouping, and event-aware token formation all rely on sparse kinematic structure rather than on dense image-like layouts. By contrast, range–Doppler or range–azimuth heatmaps remain useful radar representations, but they shift the design burden to a different front-end: dense spatial encoding, patch/token formation over continuous grids, and a less direct relationship between the representation and localized motion events.

At the same time, the point-cloud choice should not be interpreted as a claim that heatmap-like representations are unsuitable for language interfaces. Rather, the present paper makes a narrower design decision: it studies whether a radar-native structured interface can be built directly on sparse point clouds. In that sense, heatmap-based radar-language modeling is better viewed as an extension path than as a one-to-one substitute for the current interface.

How to interpret the discrete radar interface baseline. The Discrete Radar Interface baseline is included to provide a controlled comparison for a different interface philosophy under the same radar input source, normalization policy, point budget, task prompts, and language-backend protocol. It encodes the same Top-K radar stream into 256-dimensional frame-level embeddings and quantizes them using a learned 512-entry codebook with straight-through vector quantization ($\beta_{\text{commit}} = 0.25$). The resulting code embeddings are supplied to the shared decoder. Unlike RILA, this baseline does not expose a continuous dual-order event-aware prefix consisting of one global token and M event

Table 20: Radar representation choices as interface substrates. The table positions the current point-cloud interface against two broader families of radar representations without changing the scope of the present paper.

Representation	Strengths for language interfacing	Main limitations	Relation to the present paper
Sparse point clouds	Expose explicit kinematics, compact tokenization, and direct compatibility with event-aware token abstraction	Sensitive to sparsity, clutter, and imperfect point extraction	Primary representation studied by RILA
Heatmap-like radar tensors	Retain dense spatial / spectral structure and may be easier to integrate with vision-style backbones	Require a different front-end for patching, tokenization, and temporal grouping; less directly tied to sparse event tokens	Meaningful extension path, but outside the present interface design
Raw radar signal tensors	Preserve the most complete low-level sensing information	Highest preprocessing and learning burden; weak immediate compatibility with compact event-aware token formation	Out of scope for the present submission

Table 21: RadarLLM comparison on the overlapping clip-summary task. The first block reports an independent RadarLLM system reimplementation; the second compares radar interfaces under a shared Qwen3.5 backend. Values with $\pm$ are mean and standard deviation over three training seeds; unmarked results are fixed submitted checkpoints.

Method	ROUGE-L ↑	CIDEr ↑
Independent system comparison		
RadarLLM (independent reimplementation)	30.5 ± 0.4	6.1 ± 0.3
RILA (fixed checkpoint)	34.3	8.1
Shared-Qwen3.5 interface comparison		
RadarLLM tokenizer (independent reimplementation)	31.4 ± 0.3	6.5 ± 0.5
Discrete Radar Interface	32.2	7.2
RILA (fixed checkpoint)	34.3	8.1

tokens. Its role is not to claim exact architectural equivalence to any prior external system, but to test whether a discretized radar-token interface is sufficient under the same evaluation protocol.

More broadly, the current benchmark should be interpreted as an interface benchmark rather than as a universal radar-language benchmark. It evaluates whether sparse radar observations can be reorganized into a token layer that is simultaneously useful for clip-level summary generation, localized event description, and temporal QA. The benchmark is therefore narrower than a general multimodal reasoning benchmark. It is focused on single-person activity sequences, a fixed point-cloud sensing setup, and a shared language backend.

Common-Protocol RadarLLM Comparison. To evaluate the overlapping clip-summary task, we independently reimplemented the published RadarLLM method [13]. The evaluation uses the same subject-disjoint real-world split, reference summaries, and ROUGE-L/CIDEr evaluator as RILA. Implementation choices were fixed before test evaluation, and checkpoints were selected using validation data. Results for the reproduced RadarLLM system are reported as mean and standard deviation over three training seeds.

We additionally perform a shared-backend interface comparison using the same Qwen3.5 checkpoint for the reproduced RadarLLM tokenizer, the Discrete Radar Interface, and RILA. This control matches the data split, task prompts, optimization protocol, training budget, validation selection, decoding settings, and evaluator, allowing a more direct comparison of interface design choices.

The shared-Qwen3.5 comparison further isolates the interface choice from differences in the language backend. RILA obtains the highest observed clip-summary scores in both comparisons. Because reproduced variants use three training seeds while RILA and Discrete use fixed checkpoints, the differences are interpreted descriptively rather than as statistical significance claims.

Scope and non-claims. The present submission does not claim to introduce a fully general radar-to-language benchmark, nor does it claim that point clouds are the only valid substrate for radar-language

Table 22: Ethical scope and release safeguards. The table summarizes the main risks that remain relevant even for a radar-based benchmark and the safeguards adopted in the current submission.

Risk	Why it matters here	Current mitigation	Recommended future safeguard
Sensitive behavior inference	Language outputs can make private behaviors explicit even when the sensing modality is not visually identifying	The benchmark is used for research evaluation only, not deployment claims	Restrict release scope and require task-level usage agreements
RGB leakage	Synchronized RGB videos exist during annotation	RGB is used only for annotation and withheld at inference time	Exclude all synchronized RGB from any future benchmark release
Identity leakage through metadata	Even radar data can become sensitive if linked to subject identity or contextual metadata	Identity-bearing metadata is not part of the benchmark description	Remove personal identifiers and release only controlled benchmark metadata
Benchmark misuse	A benchmark can be reused for surveillance-style tasks beyond the intended research scope	The paper frames the benchmark as a structured interface benchmark, not a deployment-ready system	Use controlled-access release terms and explicit non-deployment licensing
Over-interpretation of QA outputs	Temporal QA may be mistaken for unrestricted reasoning rather than structured event access	The paper defines QA through event-derived structured queries	Release QA construction protocol together with scope limitations

modeling. It does not address multi-person interaction scenes, unrestricted open-domain reasoning, or cross-device generalization. Similarly, the temporal QA benchmark should be interpreted as structured access to event-aware tokens rather than as a test of open-ended world knowledge. These limits are important because they define what the current evidence supports: RILA is evaluated as a radar-native interface for sparse point-cloud sequences under a well-defined sensing and annotation protocol.

Privacy, sensitive inference, and release safeguards. Although mmWave radar is more privacy-preserving than RGB sensing, it does not eliminate privacy risks. Turning radar observations into natural-language descriptions can amplify sensitive inference by making behavior explicit. The current benchmark therefore follows three safeguards: synchronized RGB is used only for annotation and is withheld at inference time; the real benchmark remains non-public at submission time; and the present work does not claim deployment readiness for private or high-stakes environments. Any future release should exclude synchronized RGB, identity metadata, and personal information, and should adopt controlled-access conditions appropriate for behavior-sensitive sensing data.

Taken together, this section situates the present benchmark and interface more precisely. RILA should be understood as a point-cloud radar-language interface under a specific but practically meaningful sensing regime. It is not a claim that all radar-language modeling should follow the same representation, nor a claim that privacy-friendly sensing removes all ethical concerns. Rather, the present evidence supports a narrower conclusion: sparse mmWave point clouds can be reorganized into a structured token layer that is useful for multiple language tasks, provided that the benchmark scope and deployment boundaries are stated clearly.

F Technical Supplement and Scope of Formal Claims

The purpose of this section is to formalize the limited technical statements that the present design of RILA actually relies on. We do not use the appendix to replace missing experiments with stronger theoretical claims. Instead, we focus on four concrete questions. First, what kind of locality guarantee is provided by the 4D Hilbert-based geometry order used in the interface? Second, under the chosen parameterization, why does the dual-scan selective state-space encoder remain stable? Third, why does the structured interface-language alignment objective not admit trivial radar-side collapse when the text side is frozen? Fourth, in what precise sense does event-aware token abstraction compress the radar stream while preserving the structure needed for multiple downstream language tasks?

F.1 Discrete Locality of 4D Hilbert Serialization

The geometry-order branch serializes quantized kinematic coordinates through a discrete 4D Hilbert traversal. The property needed by RILA is a one-sided locality statement: nearby positions in Hilbert order have bounded distance in the calibrated 4D grid. We do not require or claim the converse.

Let

$$\mathcal{G}_k = \{0, \dots, 2^k - 1\}^4$$

be the 4D grid of order k, and let

$$h_k : \mathcal{G}_k \to \{0, \dots, 16^k - 1\}$$

denote the discrete 4D Hilbert traversal.

Proposition F.1. There exists a constant $C_H > 0$, independent of k, such that for any $u, v \in \mathcal{G}_k$,

$$\|u - v\|_2 \le C_H \big(|h_k(u) - h_k(v)| + 1\big)^{1/4}. \tag{25}$$

Proof. The proof follows the standard Hölder locality of the 4D Hilbert curve. Let $\psi : [0,1] \to [0,1]^4$ denote the corresponding continuous Hilbert curve. There is a dimension-dependent constant C_4 such that

$$\|\psi(s) - \psi(t)\|_2 \le C_4 |s - t|^{1/4} \qquad \text{for all } s, t \in [0,1]. \tag{26}$$

For a discrete cell u, define the normalized cell center

$$c(u) = \frac{u + \frac{1}{2}\mathbf{1}}{2^k}$$

and the midpoint parameter

$$s_u = \frac{h_k(u) + \frac{1}{2}}{16^k}.$$

The finite-order construction maps each Hilbert subinterval of length 16^{-k} to a grid cell of side length 2^{-k}. Thus the cell center differs from the continuous curve value at the midpoint parameter by at most $C_4' 2^{-k}$. For two cells u, v,

$$\|c(u) - c(v)\|_2 \le C_4 |s_u - s_v|^{1/4} + C_4' 2^{-k}.$$

Because

$$|s_u - s_v| = \frac{|h_k(u) - h_k(v)|}{16^k},$$

multiplying by 2^k yields

$$\|u - v\|_2 \le 2^k C_4 \left(\frac{|h_k(u) - h_k(v)|}{16^k} \right)^{1/4} + C_4' \le C_H \big(|h_k(u) - h_k(v)| + 1\big)^{1/4}.$$

This proves the claim. □

Scope. The proposition is one-sided. A small Hilbert-index gap implies a bounded grid distance, but nearby grid cells may still be separated in Hilbert order. This is sufficient for RILA because the geometry branch uses Hilbert traversal as a locality-preserving serialization, not as an exact inverse map between geometry and sequence order.

F.2 Stability of the Dual-Scan Selective State-Space Encoder

This subsection establishes token-wise contraction and finite-clip boundedness for the selective state-space branch. The analysis does not require a global input-independent contraction bound.

For branch b, the continuous-time transition matrix is parameterized as

$$A^b = -\operatorname{diag}(\exp(a^b)).$$

The selective scan uses

$$\bar{A}_n^b = \exp(\Delta_n^b A^b),$$

where $\Delta_n^b > 0$ for finite softplus inputs. The discretized input term is denoted by $\bar{B}_n^b x_n^b$. The exact form of $\bar{B}_n^b$ depends on the selective-scan discretization, but the stability argument below only requires the injected input term $\bar{B}_n^b x_n^b$ to be bounded over the finite clip.

Proposition F.2. Let

$$\alpha^b_{\min} = \min_i \exp(a^b_i) > 0.$$

For every token index n with finite $\Delta^b_n > 0$,

$$\|\bar{A}^b_n\|_2 \le \exp(-\Delta^b_n \alpha^b_{\min}) < 1. \tag{27}$$

Proof. Since A^b is diagonal with entries $-\exp(a^b_i)$,

$$\bar{A}^b_n = \operatorname{diag}\big(\exp(-\Delta^b_n \exp(a^b_1)), \ldots, \exp(-\Delta^b_n \exp(a^b_d))\big).$$

The spectral norm of a diagonal matrix equals the maximum absolute diagonal entry. Therefore,

$$\|\bar{A}^b_n\|_2 = \max_i \exp(-\Delta^b_n \exp(a^b_i)) \le \exp(-\Delta^b_n \alpha^b_{\min}).$$

For finite softplus inputs, $\Delta^b_n > 0$, so the right-hand side is strictly smaller than one. □

Corollary F.3. Consider a finite clip with token indices $1, \ldots, N$. Define the clip-specific minimum step size

$$\underline{\Delta}^b_N = \min_{1 \le n \le N} \Delta^b_n.$$

For finite softplus inputs, $\underline{\Delta}^b_N > 0$. Let

$$\rho^b_N = \exp(-\underline{\Delta}^b_N \alpha^b_{\min}) < 1.$$

If the injected input term is bounded over the clip,

$$\|\bar{B}^b_n x^b_n\|_2 \le U^b_N \qquad \text{for all } 1 \le n \le N,$$

then for every $n \le N$,

$$\|h^b_n\|_2 \le (\rho^b_N)^n \|h^b_0\|_2 + \frac{U^b_N (1 - (\rho^b_N)^n)}{1 - \rho^b_N}. \tag{28}$$

Proof. Expanding the recursion gives

$$h^b_n = \left(\prod_{m=1}^{n} \bar{A}^b_m \right) h^b_0 + \sum_{j=1}^{n} \left(\prod_{m=j+1}^{n} \bar{A}^b_m \right) \bar{B}^b_j x^b_j.$$

Using Proposition F.2 and the clip-specific bound

$$\|\bar{A}^b_m\|_2 \le \rho^b_N \qquad \text{for all } m \le N,$$

we obtain

$$\|h^b_n\|_2 \le (\rho^b_N)^n \|h^b_0\|_2 + U^b_N \sum_{j=1}^{n} (\rho^b_N)^{n-j}.$$

The geometric sum gives Eq. (28). □

Scope. This result establishes finite-clip boundedness under bounded input injection, rather than a global BIBO-stability claim. The statement is sufficient for the finite radar clips used in the paper: with finite softplus steps and bounded input injection, the branch recursion remains bounded over the clip. Larger local kinematic changes increase Δ^b_n, which strengthens attenuation of stale state contributions near motion discontinuities.

F.3 Non-Collapse of SGSA under a Frozen Text Side

The structured interface-language alignment objective aligns radar-side interface tokens with text embeddings while keeping the text encoder frozen. A natural concern is whether the radar-side embeddings could collapse to a constant vector. We show a simple but useful statement: under standard batch-diversity assumptions, the fully collapsed radar-side solution cannot be globally optimal.

Consider the event-level alignment loss on a batch of size B:

$$\mathcal{L}_{\text{evt}-\text{align}} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(z_i^\top e_i / \tau)}{\sum_{j=1}^{B} \exp(z_i^\top e_j / \tau)},$$

where $\|z_i\|_2 = \|e_i\|_2 = 1$, the e_i are frozen text embeddings, and the z_i are radar-side embeddings.

Proposition F.4. Suppose that at least two text embeddings in the batch differ, i.e. $e_i \neq e_j$ for some $i \neq j$. Then any fully collapsed radar-side solution $z_1 = \cdots = z_B = z$ satisfies

$$\mathcal{L}_{\mathrm{evt-align}} \geq \log B.$$

Moreover, if $\max_{i \neq j} e_i^\top e_j < 1$, then the non-collapsed assignment $z_i = e_i$ yields strictly smaller loss than any fully collapsed solution. Hence the collapsed family cannot be a global minimizer.

Proof. Under full collapse,

$$\mathcal{L}_{\mathrm{evt-align}} = -\frac{1}{B}\sum_{i=1}^{B}\frac{z^\top e_i}{\tau} + \log\sum_{j=1}^{B}\exp(z^\top e_j/\tau).$$

Define the batch mean

$$\mu = \frac{1}{B}\sum_{i=1}^{B} e_i.$$

Then

$$\mathcal{L}_{\mathrm{evt-align}} = -\frac{z^\top \mu}{\tau} + \log\sum_{j=1}^{B}\exp(z^\top e_j/\tau).$$

By Jensen's inequality,

$$\log\sum_{j=1}^{B}\exp(z^\top e_j/\tau) \geq \log\big(B\exp(z^\top \mu/\tau)\big) = \log B + \frac{z^\top \mu}{\tau}.$$

Substituting gives

$$\mathcal{L}_{\mathrm{evt-align}} \geq \log B.$$

Now consider the non-collapsed assignment $z_i = e_i$. The i-th term becomes

$$\ell_i = -\frac{1}{\tau} + \log\left(\exp(1/\tau) + \sum_{j\neq i}\exp(e_i^\top e_j/\tau)\right).$$

If every off-diagonal similarity is strictly smaller than 1, then each cross term is strictly smaller than $\exp(1/\tau)$, so

$$\ell_i < -\frac{1}{\tau} + \log(B\exp(1/\tau)) = \log B.$$

Averaging over i yields a total loss strictly below $\log B$, proving that the collapsed family is not globally optimal. □

Interpretation. This is intentionally a limited statement. It does not prove that optimization always converges to the best non-collapsed solution. It only shows that, under the actual design choices used in RILA, the contrastive structure of SGSA does not make full radar-side collapse a globally optimal outcome when the text targets in the batch are semantically diverse.

F.4 Token Compression and Multi-Task Reuse

The final interface maps a dense radar stream of TK frame-level tokens into one global clip token and M event-aware tokens:

$$\mathcal{I}(\mathcal{P}_{1:T}) = [\bar{u}^{\mathrm{clip}}, \bar{u}_1^{\mathrm{evt}}, \ldots, \bar{u}_M^{\mathrm{evt}}].$$

The point of this subsection is not to prove a deep information-theoretic optimality statement, but to formalize the exact sense in which the interface compresses the decoder-side representation.

Proposition F.5. If $M \ll TK$, then the interface reduces the number of tokens exposed to the language model from $O(TK)$ to $O(M+1)$ while retaining both global clip context and localized event structure.

Proof. Before event-aware abstraction, the radar stream contains TK frame-level tokens after Top-K selection. After frame pooling, event proposal, segment pooling, and token projection, the language model receives exactly one global clip token plus M event tokens. Therefore the decoder-side prefix length is

$$|\mathcal{I}(\mathcal{P}_{1:T})| = M + 1.$$

If $M \ll TK$, then this is a strict compression of the decoder prefix from $O(TK)$ to $O(M+1)$. The compression is structured rather than arbitrary: the clip token preserves global context, while the event-aware tokens preserve localized motion units. □

Interpretation. This proposition is simple but central to the paper's framing. RILA is not only an encoder; it is a token-economy layer. The compressed prefix is also what makes the same interface reusable across clip summary generation, ordered event description, and temporal QA. In other words, the usefulness of the interface is tied not only to representation quality but also to the fact that the resulting token layer remains compact and task-sharable.

F.5 Scope of the Formal Claims

The formal results above support four limited statements: the geometry-order branch has a discrete one-sided locality guarantee; each selective state-space branch is token-wise contractive and finite-clip bounded under bounded input injection; the SGSA objective does not admit full collapse as a global optimum under frozen and diverse text targets; and event-aware abstraction compresses the decoder prefix while preserving a two-level token structure. These claims are intentionally narrower than stronger statements about unrestricted reasoning, exact segmentation optimality, or cross-device generalization. The role of this section is to formalize what the current paper actually relies on, not to substitute theorem statements for empirical evidence.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction match the paper's scope, namely a radar-native structured interface for sparse mmWave point clouds evaluated on summary, ordered event description, and temporal QA.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated limitations subsection discussing the current single-person scope, the interpretation of temporal QA, the sensing setup, and privacy-related usage boundaries.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The assumptions and proofs for the formal results are provided in Appendix F, including Hilbert locality, branch stability, SGSA non-collapse, and token compression.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper and appendix describe the datasets, supervision protocol, model structure, training stages, evaluation tasks, and controlled-baseline settings needed to reproduce the main claims.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The current submission does not provide open access to the full benchmark and dataset at submission time, although the protocol and algorithmic structure are documented in the appendix.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The benchmark structure, supervision layers, model configuration, training schedule, and evaluation protocol are specified in the paper and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report mean and standard deviation over three training seeds for the RadarLLM reimplementations. The main RILA and controlled-baseline results use fixed checkpoints without multi-seed uncertainty estimates.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix D.2 reports inference hardware and batch-1 latency, but complete per-run training costs and total compute are not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is presented as a research benchmark and interface study, and the paper discusses privacy-sensitive inference and usage boundaries.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: The paper discusses both positive uses of privacy-preserving human sensing and negative risks such as sensitive behavior inference and misuse in private spaces.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: The paper states that synchronized RGB is used only for annotation and withheld at inference time, and Appendix E discusses safeguards for any future benchmark release or use.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The existing assets used in the paper are credited, and their sources and usage context are documented in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The submission introduces new components, but these assets are not released alongside the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The study did not use crowdsourcing or external participant recruitment. The real-world recordings involved only the authors and their adult family members as voluntary participants. All participants gave prior consent, and no minors were involved. The recordings captured participant movements for activity annotation and radar-language evaluation. No compensation was provided.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [No]

Justification: No formal IRB or equivalent institutional review was obtained for this study. The real-world recordings involved only the authors and their adult family members as voluntary participants. All participants gave prior consent, no minors were involved, and the recordings captured participant movements for activity annotation and radar-language evaluation. The paper describes the participant scope, privacy handling, and potential risks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The method uses an LLM as part of the model pipeline. Any LLM assistance in writing, editing, or language refinement did not affect the methodology, experiments, results, or conclusions.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.