2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

StreamVLN: Streaming Vision-and-Language Navigation via SlowFast Context Modeling

Meng Wei[*1,2], Chenyang Wan[*1,3], Xiqian Yu[*1], Tai Wang[*1,†], Xiaohan Mao[1,4] Chenming Zhu[1,2], Wenzhe Cai[1], Hanqing Wang[1], Yilun Chen[1],Xihui Liu[2,‡],Jiangmiao Pang[1,‡]

***Abstract*— Vision-and-Language Navigation (VLN) in real-world settings requires agents to process continuous visual streams and generate actions with low latency grounded in language instructions. While Video-based Large Language Models (Video-LLMs) have driven recent progress, current VLN methods based on Video-LLM often face trade-offs among fine-grained visual understanding, long-term context modeling and computational efficiency. We introduce StreamVLN, a streaming VLN framework that employs a hybrid slow-fast context modeling strategy to support multi-modal reasoning over interleaved vision, language and action inputs. The fast-streaming dialogue context facilitates responsive action generation through a sliding-window of multi-turn dialogues, while the slow-updating memory context compresses historical visual states using a 3D-aware token pruning strategy. With this slow-fast design, StreamVLN achieves real-time dialogues through KV cache reuse, supporting long video streams with bounded context size and inference cost. Experiments on VLN-CE benchmarks show state-of-the-art performance with low latency, ensuring robustness and efficiency in real-world deployment. The project page is: https://streamvln.github.io/.**

I. Introduction

Vision-and-Language Navigation (VLN) in continuous real-world environments is a critical task in embodied AI, where an agent must ground linguistic cues in visual observations and plan actionable trajectories. However, achieving robust VLN remains challenging due to the need for fine-grained multimodal alignment, long-term sequence reasoning, and generalization to unseen environments. Recent advances in Video-LLMs offer new capabilities for VLN systems. Several research efforts [1], [2], [3] have extended Video-LLMs to vision-language-action models (VLA) for navigation, which integrate visual observation encoding, language understanding, and action prediction in a unified end-to-end framework.

For real-world navigation, VLA models must process continuously incoming video streams, where maintaining long-term context and real-time responsiveness are both crucial. This poses challenges for Video-LLMs in managing linearly growing visual tokens. Some methods [3], [4] sample a fixed number of video frames, but the limited temporal resolution may fail to accurately predict low-level actions when fine-grained temporal changes are needed. Other methods [1], [2] compress vision tokens into sparse memory tokens via pooling or token merging, which helps control the token volume but sacrificing temporal and visual details. Furthermore, these methods typically require refreshing the LLM's dialogue context at every action step. This leads to significant redundant computation during both training and inference, hindering data scalability and real-world deployment. In this paper, we propose StreamVLN, a novel streaming vision-and-language navigation framework for low-latency action generation. We extend Video-LLM into an interleaved vision-language-action model, enabling continuous interaction with a video stream through multi-turn dialogue. To address the challenges of long-term context modeling and computational efficiency, StreamVLN introduces a hybrid strategy that combines a **fast-streaming dialogue context** and a **slow-updating memory context**. Specifically, it employs a sliding-window mechanism to cache the key/value states (KV) of tokens over a fixed number of dialogue turns for highly responsive action decoding. After each streaming dialogue ends, the visual context within the window is consolidated into memory at a slower pace, to provide long-term context for subsequent windows.

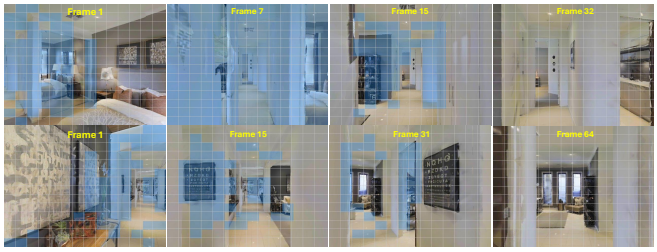

Fig. 1. Visualization of the proposed 3D spatial pruning strategy. The spatial-temporal redundancy in ego-centric VLN video data arises from fine-grained low-level action trajectories. While temporal sampling alleviates part of this redundancy, substantial patch-level redundancy remains at the spatial level under high input resolutions of Video-LLMs.

Moreover, current state-of-the-art Video-LLMs are trained on high-resolution visual inputs, inevitably generating a large number of visual tokens which cause substantial KV cache and decoding overhead. However, as shown in Figure 1, even after temporal sampling, egocentric VLN videos retain high spatial redundancy, where 3D spatial cues offer an effective and efficient means to remove such redundancy. Hence, we propose a **training-free** spatial pruning strategy guided by voxel-based 3D proximity. Compared to video token compression strategies for generic offline videos [5], which often rely on costly token feature similarity computations or operations on large attention matrices, our geometry-based spatial pruning has higher computation efficiency and sup-

[1]Shanghai AI Lab, [2]The University of Hong Kong, [3]Zhejiang University
[4]Shanghai Jiao Tong University
[*]Equal Contribution, [†]Project Lead, [‡]Corresponding Authors

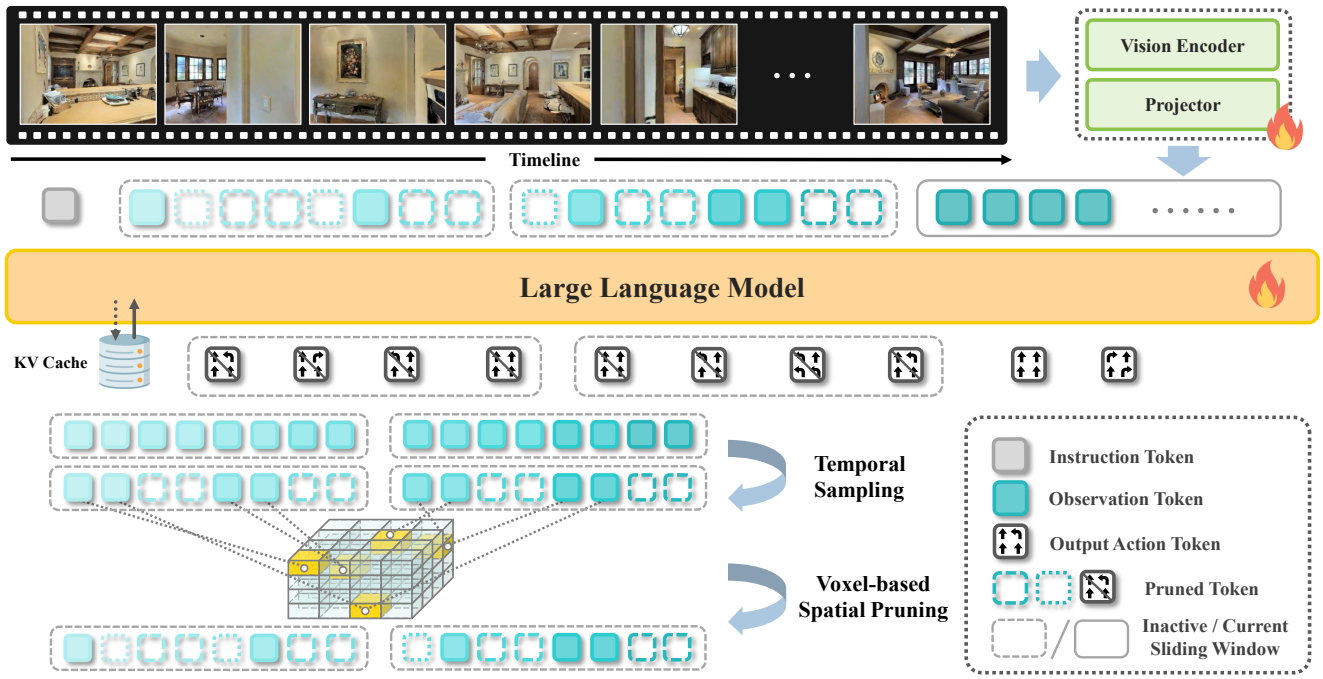

Fig. 2. **Framework of StreamVLN.** The input consists of a language instruction and a stream of RGB images. Each navigation episode is framed as a multi-turn dialogue, where the agent continually queries for the next actions. To support long-horizon reasoning while maintaining a manageable context size and low latency, we adopt a fixed-size sliding window to retain recent dialogue history. The context in inactive windows is updated by token pruning to to reduce memory overhead.

ports streaming video processing and KV cache compression.

In summary, StreamVLN offers an efficient and scalable solution to suit the continuous interaction requirements of real-time vision-and-language navigation. Its slow-fast context modeling design enables the model trained on short clips (e.g., 16 frames), to work effectively on long video streams, without incurring context length growth or compromising inference latency. Experiments on existing VLN-CE benchmarks shows that StreamVLN achieves superior performance while maintaining low latency.

II. Related Work

Vision-and-Language Navigation (VLN). This task requires an agent to follow language instructions while perceiving and acting in environments. Early progress mainly focused on discrete settings, where agents navigate by "teleporting" between predefined nodes of a discrete scene graph [6], [7], [8], [9]. This formulation emphasizes high-level decision-making but ignores the challenges of real-world navigation. More recent work [10], [11], [12], [13] has focused on continuous environments [14], where agents must perform low-level actions in realistic simulators. To address the increased complexity, some methods incorporate a waypoint predictor [13], [15] pretrained in simulators to propose candidate positions, which are then used to guide high-level navigation decisions. Although these approaches have achieved strong performance, the waypoint predictors typically rely heavily on the training scenes and exhibit limited generalization to unseen scenes. Therefore, more flexible and scalable navigation frameworks is needed to generalize better to long-horizon, real-world setting.

Navigation with Multi-Modal Large Language Models (MLLMs). Recent advancements in MLLMs have opened new possibilities for VLN by enabling agents to interpret and reason over natural language instructions in a more generalizable way. Some methods [16], [17], [18] directly use LLMs as planner in a training-free manner within a modular framework. But there's still a performance gap compared to task-specific models. Other lines of work [1], [2], [3], [4] further fine-tune Video-based LLMs [19], [20], [21] to better capture spatial-temporal information and generate low-level actions in an end-to-end manner, but often face challenges in balancing computational efficiency and long-horizon memory retention. StreamVLN aims to better accommodate streaming video input by introducing an efficient and scalable framework that supports action generation with coherent multi-turn reasoning with low-latency response and bounded memory usage.

III. Method

StreamVLN generates action outputs from continuous video input in an online, multi-turn dialogue manner. Built on LLaVA-Video [19], we extend it for interleaved vision, language, and action modeling. The overall framework of StreamVLN is shown in Figure 2. We briefly introduce the autoregressive generation in continuous multi-turn dialogues for a streaming VLN process (Section III-A). For both effective context modeling of long sequence and efficient computation for real-time interaction, StreamVLN has: (1) a

fast-streaming dialogue context with a sliding-window KV cache (Section III-B); and (2) a slow-updating memory via token pruning (Section III-C). Finally, we describe how we curate the navigation data and incorporate diverse multi-modal data for multi-task training (Section III-D).

A. Preliminary: Multi-Turn Autoregressive Generation

A multi-turn dialogue session for VLN consists of a sequence of interleaved observations and actions. In each dialogue $d_i = (o_i, a_i)$, the VLN model receives a new observation o_i and produces an action response a_i conditioned on both the current input and the dialogue history. The full input sequence at step i is constructed as: $o_1 a_1 o_2 a_2 ... o_{i-1} a_{i-1}$. In this streaming setting, new tokens from o_i are appended to the token stream continuously. The response a_i is generated token-by-token via autoregressive decoding. For each dialogue turn, Transformer-based LLMs first perform a **prefill phase** to encode input tokens, caching their key/value (KV) states in attention layers. These cached KV pairs are then used in the **decoding phase** to generate new tokens. If we don't use KV cache across turns, the model will repeat this prefilling process of all previous tokens for a new dialogue.

B. Fast-Streaming Dialogue Context

While multi-turn KV cache reuse can eliminate over 99% of prefilling time, it introduces substantial memory overhead. As the number of dialogues increases, the KV cache grows linearly (e.g., 2K tokens can consume around 5GB of memory), making long sessions impractical. In addition, existing Video-LLMs tend to exhibit degraded reasoning performance when processing overly long contexts.

To manage dialogue context, we adopt a sliding window KV cache over continuous dialogues, retaining a fixed number N of recent dialogues in an active window: $W_j = [o_{(i-N+1)} a_{(i-N+1)} ... o_i a_i]$ When the window reaches capacity, the key/value states are offloaded from the LLM, and the states of non-observation dialogue tokens, such as prompts and generated actions, are immediately discarded. For the new sliding window, the token states from past windows are processed into memory token states $\{\mathcal{M}_0, \ldots, \mathcal{M}_j\}$ (as detailed in Section III-C). Formally, for the latest observation o_i, the decoder generates a_i based on the cached token states and the current window's KV cache:

$$a_i^{W_{j+1}} = \text{Decoder}\big(o_i, \{\mathcal{M}_0, \ldots, \mathcal{M}_j\}, \{k_{(i-N+1)} v_{(i-N+1)}, \ldots, k_{(i-1)} v_{(i-1)}\}\big).$$

C. Slow-Updating Memory Context

We observe that most VLN trajectories, collected with fine-grained low-level actions, contain redundant observations. To mitigate this, we first adopt a fixed-number temporal sampling strategy following [3]. However, at the spatial level, heavy patch-level redundancy remains under the high input resolution of Video-LLMs. Direct feature-level compression during training—such as average pooling or learnable Q-Formers—led to substantial performance degradation in our experiments, as it altered the pretrained input distribution, thereby undermining pretraining knowledge.

To address the spatial redundancy without disrupting pretrained features, we introduce a training-free voxel-based 3D spatial pruning strategy, which is also tailored for streaming video processing and KV cache memory. Specifically, we back-project the 2D image patches from the video stream into a shared 3D space using depth information. By discretizing this 3D space into uniform voxels, we can track the voxel indices of the patch tokens over time. If multiple tokens from different frames within a given duration are projected into the same voxel, only the token from the most recent observation is retained, as detailed in Algorithm 1. The voxel pruning mask M is then used to select the preserved token states.

Algorithm 1 Voxel-Based Spatial Pruning

Voxel map $V \in \mathbb{Z}^{T \times H \times W}$, stride K, threshold θ
Pruning Mask $M \in \{0,1\}^{T \times H \times W}$
Initialize $M \leftarrow \mathbf{0}$, map `latest` $\leftarrow \emptyset$
for each token (t, x, y) with $V_{t,x,y} \geq 0$ **do**
  $p \leftarrow \lfloor t/K \rfloor$, $v \leftarrow V_{t,x,y}$
  if (p, v) not in `latest` or t is newer **then**
    `latest`$[(p, v)] \leftarrow (t, x, y)$
  end if
end for
Set $M_{t,x,y} \leftarrow 1$ for all $(t, x, y) \in$ `latest`
For each t, if $\sum_{x,y} M_{t,x,y} < \theta \cdot H \cdot W$, set $M_{t,:,:} \leftarrow 0$
return M

D. Co-Training with Multi-Source Data.

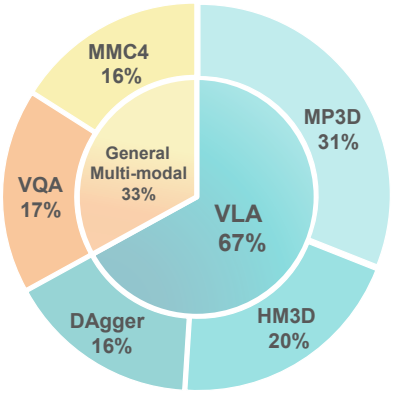

Fig. 3. Co-Training Data Recipe of StreamVLN

Vision-Language Action Data. We collect navigation-specific training data using the Habitat simulator across multiple public VLN datasets. Specifically, we collect 450K samples (video clips) from 60 Matterport3D [22] (MP3D) environments, sourced from R2R [6], R2R-EnvDrop [23] and RxR [7]. To further improve generalization through increased scene diversity, we incorporate an additional 300K samples from a subset of ScaleVLN [15], spanning 700 Habitat Matterport3D [24] (HM3D) scenes. In addition, we adopt the DAgger algorithm to enhance the model's robustness and generalization ability in novel scenes and during error recovery. Using Habitat's shortest-path follower as the expert

policy, we collect corrective demonstrations on model rollouts after the initial training stage. These DAgger-collected samples (240K) are then incorporated into the training set for co-training.

General Vision-Language Data. To retain the general reasoning capabilities of the pretrained Video-LLM, we incorporate a diverse set of multimodal training data that complements navigation supervision. Specifically, we include 248K video-based visual question-answering (VQA) samples sourced from publicly available datasets LLaVA-Video-178K [31] and ScanQA [32], which combine general video QA with 3D scene understanding to support spatial-temporal and geometric reasoning. To further augment the model's capacity for multi-turn vision-language interactions, we incorporate 230K interleaved image-text samples from MMC4 [33], which strengthens its ability to parse and generate contextually coherent responses with interleaved visual and textual reasoning.

IV. Experiments

A. Experimental Setup

Simulation Benchmark Setup. We evaluate our method on two public VLN-CE [14] benchmarks collected from Matterport3D scenes using the Habitat simulator: R2R-CE [6] and RxR-CE [7]. R2R-CE provides 5.6K English trajectories with an average length of 10 meters, while RxR-CE includes 126K multilingual instructions (English, Hindi, Telugu) and features longer, more diverse paths (avg.15 meters). The camera HFOVs are 79° for R2R-CE and RxR-CE. Both benchmarks require realistic indoor navigation under continuous control. As our goal is to assess the generalization ability, we focus on the validation unseen splits of both benchmarks We report standard VLN metrics, including Navigation Error (NE), Success Rate (SR), Oracle Success Rate (OS), and Success weighted by Path Length (SPL), following prior works.

Real-World Evaluation Setup. We perform real world experiments based on a Unitree Go2 robotic dog. The robot is equipped with a upward facing camera (Intel® RealSense™ D455) for RGB-D observations. We deploy StreamVLN on a remote workstation with an RTX 4090 GPU. The Go2 robot continuously streams visual data to the 4090 server for inference, which returns executable action commands to the robot. The averge inference (0.27s for 4 actions) and communication (0.2s for indoor and 1.0s for outdoor environments) latency enable real-time physical deployment.

B. Implementation Details

We build StreamVLN based on the LLaVA-Video [19] 7B model, which uses Qwen2-7B [34] as the language model. Training is conducted in two stages. First, we fine-tune it for one epoch solely on oracle VLN trajectories. Then, we use the model to collect DAgger trajectories and continue training for an additional epoch with a mixture of VLN and general multimodal data. During the warm-up phase, we apply a peak learning rate of 2e-5 for the language model and 5e-6 for the vision encoder. Each training step processes 128 video clips. Training is in around 1500 A100 GPU hours.

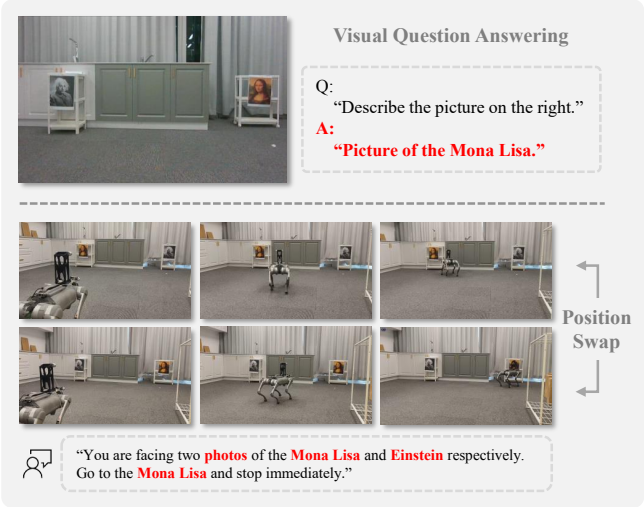

Fig. 4. StreamVLN transfers visual reasoning ability to interpreting out-of-domain navigation instructions.

C. Comparisons with State-of-the-Arts

Results on VLN-CE benchmark. Table I shows the performance of our method on the VLN-CE R2R and RxR benchmarks under the Val-Unseen setting, compared with existing VLN-CE methods. StreamVLN achieves state-of-the-art performance among RGB-only methods both without and with extra navigation datasets, reaching 56.9% SR and 51.9% SPL on R2R, and 52.9% SR and 46.0% SPL on RxR. These results highlight the robustness of StreamVLN across both standard and long-horizon navigation tasks. Notably, StreamVLN performs comparably to ETPNav [13], despite not relying on additional panoramic or waypoint supervision. Furthermore, compared to HMAT trained on entire ScaleVLN dataset with 3 million trajectories, StreamVLN surpasses it with a small subset of ScaleVLN (150k), showing better data efficiency.

Effectiveness of Voxel-Based Spatial Pruning. Table I bottom shows the effect of applying voxel-based spatial pruning during inference. We further evaluate two settings: 1) StreamVLN-Prune-Mem. Pruning is applied only on 8 memory frames, reducing memory tokens by 28% on R2R and 22% on RxR without noticeably affecting performance. On the shorter-horizon R2R benchmark, performance even improves, indicating that the current memory still contains redundant tokens for some tasks. Proper pruning thus helps the model focus on relevant tokens, thereby enhancing navigation accuracy.

2) StreamVLN-Prune-All. Pruning is applied to both memory tokens and the KV cache of streaming frame tokens, reducing visual tokens by 32% on R2R and 30% on RxR, with only a slight decrease in performance.

Results on Video Question Answering. To evaluate StreamVLN's spatial scene understanding capabilities, we

TABLE I

COMPARISON WITH STATE-OF-THE-ART METHODS ON VLN-CE R2R AND RXR VAL-UNSEEN SPLIT.

Method	Observation				R2R Val-Unseen				RxR Val-Unseen			
	Pano.	Odo.	Depth	S.RGB	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
HPN+DN* [25]	✓	✓	✓		6.31	40.0	36.0	34.0	-	-	-	-
CMA* [26]	✓	✓	✓		6.20	52.0	41.0	36.0	8.76	26.5	22.1	47.0
VLN⟳BERT * [26]	✓	✓	✓		5.74	53.0	44.0	39.0	8.98	27.0	22.6	46.7
Sim2Sim* [27]	✓	✓	✓		6.07	52.0	43.0	36.0	-	-	-	-
GridMM* [28]	✓	✓	✓		5.11	61.0	49.0	41.0	-	-	-	-
ETPNav* [13]	✓	✓	✓		4.71	65.0	57.0	49.0	5.64	54.7	44.8	61.9
ScaleVLN* [15]	✓	✓	✓		4.80	–	55.0	51.0	-	-	-	-
InstructNav [18]	✓	✓	✓	✓	6.89	–	31.0	24.0	-	-	-	-
AG-CMTP [29]	✓	✓	✓		7.90	39.2	23.1	19.1	-	-	-	-
R2R-CMTP [29]	✓	✓	✓		7.90	38.0	26.4	22.7	-	-	-	-
LAW [10]		✓	✓	✓	6.83	44.0	35.0	31.0	10.90	8.0	8.0	38.0
CM2 [11]		✓	✓	✓	7.02	41.5	34.3	27.6	-	-	-	-
WS-MGMap [12]		✓	✓	✓	6.28	47.6	38.9	34.3	-	-	-	-
ETPNav + FF [30]		✓	✓	✓	5.95	55.8	44.9	30.4	8.79	25.5	18.1	-
Seq2Seq [14]			✓	✓	7.77	37.0	25.0	22.0	12.10	13.9	11.9	30.8
CMA [14]			✓	✓	7.37	40.0	32.0	30.0	-	-	-	-
NaVid [1]				✓	5.47	49.1	37.4	35.9	-	-	-	-
MapNav [4]				✓	**4.93**	53.0	39.7	37.2	-	-	-	-
NaVILA [3]				✓	5.37	57.6	49.7	45.5	-	-	-	-
StreamVLN				✓	5.43	**62.5**	**52.8**	**47.2**	6.72	48.6	42.5	60.2
NaVILA† [3]				✓	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8
UniNaVid† [2]				✓	5.58	53.3	47.0	42.7	6.24	48.7	40.9	-
StreamVLN†				✓	4.90	63.6	56.4	50.2	**5.65**	**54.4**	**45.4**	**63.7**
StreamVLN-Prune-Mem†		✓*	✓*	✓	**4.73**	**65.5**	**57.4**	**51.1**	5.72	53.9	45.1	63.3
StreamVLN-Prune-All†		✓*	✓*	✓	4.82	65.7	56.0	48.5	5.86	53.3	44.0	63.5

∗ indicates methods using the waypoint predictor from [26]. † denotes methods using additional VLN data beyond the R2R-CE and RxR-CE benchmarks.
✓* indicates that Odo. and Depth are only used for back-projection in the **training-free voxel pruning.**

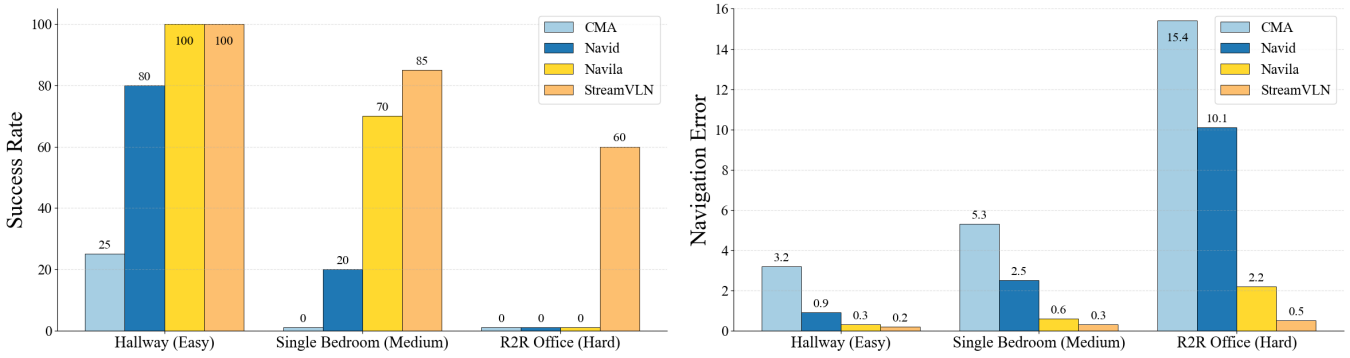

Fig. 5. Real-world experiments conducted across hallway (easy), bedroom (medium, single-room), and office (hard, room-to-room) scenarios.

conduct experiments on the widely-used ScanQA benchmark for 3D question answering based on real-world scans. StreamVLN answers questions by analyzing 16 multi-view images from each scan. As shown in Table II, StreamVLN outperforms state-of-the-art generalist navigation models such as NaviLLM [41] and NaVILA [3]. As shown in Figure 4, we observe that the strong VQA capabilities contribute to better generalization to novel navigation instructions.

Real-World Experimental Results. To quantitatively assess StreamVLN in real-world settings, we evaluated its performance in three distinct scenarios: hallway (easy), bedroom (medium difficulty, single-room), and office (hard, room-to-

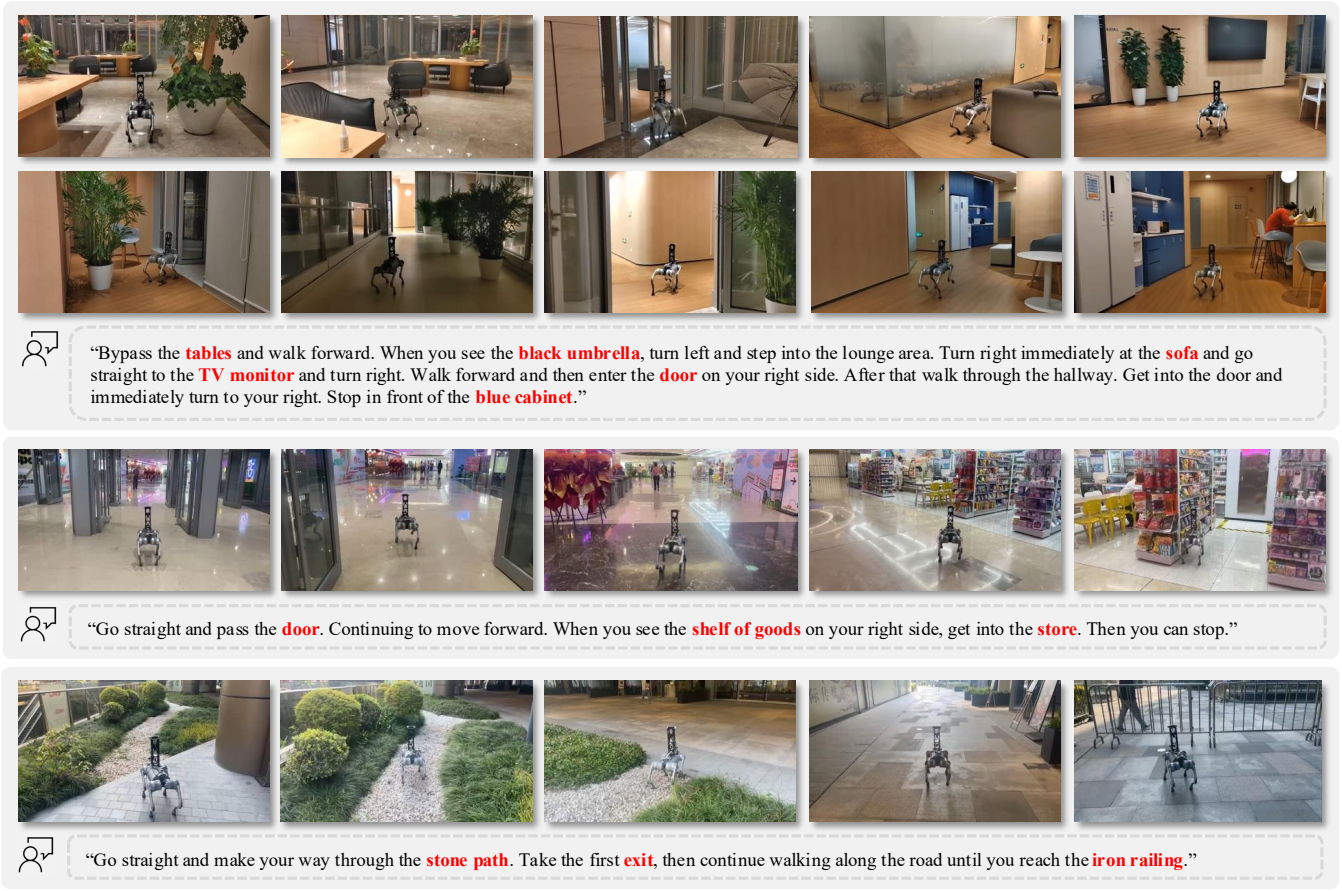

Fig. 6. Qualitative results of StreamVLN in several representative real-world environments. From top to bottom are `Home`, `Workspace`, `Mall` and `Outdoor`. StreamVLN achieves robust performance across diverse VLN scenarios, capable of accurately following complex instructions with various landmarks (marked as red) and handling real-world disturbances.

TABLE II
COMPARISON ON SCANQA VAL SET.

Method	ScanQA				
	Bleu-4 ↑	Rouge ↑	Meteor ↑	Cider ↑	EM ↑
ScanRefer [35]	7.9	30.0	55.4	11.5	18.6
ScanQA [32]	10.1	33.3	64.9	13.1	21.0
3D-VisTA [36]	10.4	35.7	69.6	13.9	22.4
3D-LLM* [37]	12.0	35.7	69.4	14.5	20.5
LEO [38]	13.2	49.2	101.4	20.0	24.5
ChatScene* [39]	14.0	-	87.6	-	-
Scene-LLM* [40]	12.0	40.0	80.0	16.6	27.2
NaviLLM [41]	12.0	38.4	75.9	15.4	23.0
NaVILA [3] (16)	15.2	48.3	99.8	19.6	27.4
StreamVLN (16)	**15.7**	**48.3**	**100.2**	**19.8**	**28.8**

∗ indicates task-specific fine-tuning. (16) means using 16 frames.

room). The baselines include the traditional learning-based CMA [26] and VLM-based approaches such as NaVid [1] and NaVILA [3]. For each method, we conducted 20 trials per scenario, measuring Success Rate (SR) and Navigation Errors (NE). As shown in Figure 5, CMA performs poorly, while NaVid succeeds only on simple, short-horizon tasks. NaVILA can follow longer-horizon instructions but fails in complex office scenarios. In contrast, StreamVLN successfully completes both simple and challenging long-horizon navigation tasks. Qualitative results in `Workspace`, `Mall`, and `Outdoor` are presented in Figure 6. Especially, the success cases in `Mall` and `Outdoor` environments highlight StreamVLN's strong generalization to novel scenes and tasks. Please refer to the *demo video* for full demonstrations.

D. Ablation Studies

Data Ablation. Table III presents an ablation study on different training data compositions. All results are reported without using voxel-based spatial pruning. The first row shows the first-stage performance when training with only oracle navigation data. After collecting DAgger data, we co-train oracle data, DAgger data, and vision-language (VL) data. In the second row, we use only VideoQA data as VL data. While the third row mixes VideoQA and MMC4 (M) data in an interleaved image-text format. For a fair comparison, the total number of VL Data is kept the same. We can observe that the second-stage co-training brings significant gains (+5.3 SR / +4.1 SPL) and incorporating MMC4 further improves performance (+2.0 SR / +1.5 SPL). Comparing the third and fourth rows, we see that adding

TABLE III

ABLATION STUDY OF DIFFERENT TRAINING DATA COMPOSITIONS ON VLN-CE R2R VAL-UNSEEN SPLIT.

R2R	RxR	DAgger	VL Data	ScaleVLN	NE↓	OS↑	SR↑	SPL↑
✓	✓				5.98	51.3	45.6	42.3
✓	✓	✓	VidQA		5.47	57.8	50.8	45.7
✓	✓	✓	VidQA+MMC4		5.43	62.5	52.8	47.2
✓	✓	✓	VidQA+MMC4	✓	**4.90**	**63.6**	**56.4**	**50.2**
✓	✓		VidQA+MMC4	✓	5.73	56.4	50.2	47.1
✓		✓	VidQA+MMC4	✓	5.90	55.9	47.9	43.6

TABLE IV

ABLATION ON THE IMPACT OF DIFFERENT MEMORY CONTEXT SIZES AND SLIDING WINDOW SIZES ON VLN-CE R2R VAL-UNSEEN SPLIT.

Memory	Window	NE↓	OS↑	SR↑	SPL↑
2*196	8	6.96	48.2	37.3	34.2
4*196	8	6.62	49.1	38.9	35.4
8*196	8	**6.05**	**53.8**	**45.5**	**41.6**
all	8	6.76	49.5	40.0	36.4
8*196	4	6.31	51.1	41.4	37.5
8*196	2	6.16	52.8	43.7	40.3

ScaleVLN data brings additional gains (+2.9 SR / +3.7 SPL). To assess the importance of DAgger data, we remove it from the co-training data, as shown in the fifth row. The results show that DAgger data plays a crucial role in boosting performance (+5.5 SR / +3.8 SPL). Furthermore, the last row highlights that incorporating RxR data yields notable performance gains (+7.8 SR / +7.3 SPL).

Memory Context Size. We study the impact of the memory context size in the hybrid context modeling strategy. As shown in Table IV (results shown are from first-stage training using only oracle VLN data), increasing the memory size from $2*196$ to $8*196$ while keeping the window size fixed at 8 significantly improves navigation performance, with SR rising from 37.3 to 45.5. This indicates the importance of fine-grained memory in supporting long-horizon reasoning. Notably, using the entire visual context (*all*) as memory doesn't yield the best results, suggesting that an overly long and varied context token sequence may introduce bias in training and hinder generalization at test time.

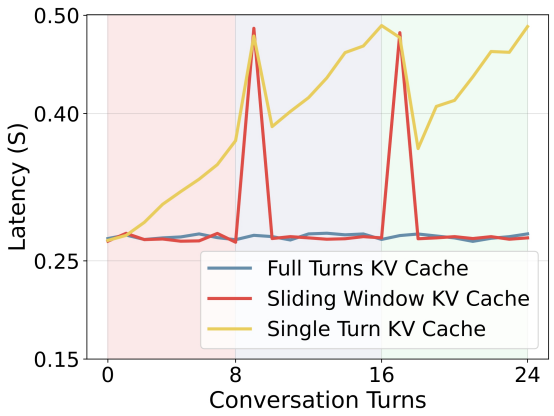

Fig. 7. Impact of KV cache reuse across multiple turns . The Sliding window size is 8 (8 conversation turns) in all three settings.

Sliding Window Size. We also evaluate the effect of the number of dialogue turns retained in the sliding window in Table IV. A smaller window size leads to more frequent shifts, resulting in a significantly larger number of training samples. For example, a window size of 8 yields approximately 450K samples, while sizes of 4 and 2 increase this to 815K and 1.5M respectively. This growth not only raises the training cost linearly but also introduces greater class imbalance, which may affect training stability. We find that retaining 8 continuous dialogue turns achieves the best balance—delivering strong navigation performance while maintaining the lowest training cost.

Effectiveness of KV-Cache Reuse. We evaluate the impact of KV cache reuse on the decoding latency under different settings. As shown in Figure 7, reusing the KV cache across all dialogue turns (Full Turns) achieves consistently low latency—since only the current observation tokens require prefill computation for generating the 8 action tokens—*but storing all the cache poses significant memory overhead*. If the KV cache is maintained only within 8 turns (Sliding Window), the decoding latency will increase at the beginning of each sliding window due to the need to prefill the previous window context tokens. Under the Single Turn setting, where the KV cache is not reused across turns (as in prior work), decoding latency steadily increases with the number of turns. Turns 0–8 incur lower latency since no historical context is included, while turns 8–16 and 16–24 have similar latency growth with a fixed memory size.

V. CONCLUSION

This paper presents *StreamVLN*, a new streaming vision-language-navigation framework based on Video-LLMs. Compared to previous Video-LLM-based VLN methods that treat each interaction as an independent dialogue and refresh history at every step, StreamVLN can reuse past key/value (KV) states through a hybrid memory design. By maintaining a fast-updating sliding window for immediate responsiveness and a slow-updating long-term memory for temporal reasoning, StreamVLN enables efficient, coherent, and scalable action generation over long video streams. Empirical results on standard VLN-CE benchmarks demonstrate that StreamVLN achieves superior performance with lower latency, paving the way for real-time long-horizon navigation.

Acknowledgements. This work is supported by Shanghai Artificial Intelligence Laboratory. The research work described in this paper was conducted in the JC STEM Lab of Autonomous Intelligent Systems funded by The Hong Kong Jockey Club Charities Trust.

REFERENCES

[1] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, "Navid: Video-based vlm plans the next step for vision-and-language navigation," *Robotics: Science and Systems*, 2024.

[2] J. Zhang, K. Wang, S. Wang, M. Li, H. Liu, S. Wei, Z. Wang, Z. Zhang, and H. Wang, "Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks," *Robotics: Science and Systems*, 2025.

[3] A.-C. Cheng, Y. Ji, Z. Yang, Z. Gongye, X. Zou, J. Kautz, E. Bıyık, H. Yin, S. Liu, and X. Wang, "Navila: Legged robot vision-language-action model for navigation," *Robotics: Science and Systems*, 2025.

[4] L. Zhang, X. Hao, Q. Xu, Q. Zhang, X. Zhang, P. Wang, J. Zhang, Z. Wang, S. Zhang, and R. Xu, "Mapnav: A novel memory representation via annotated semantic maps for vlm-based vision-and-language navigation," *arXiv preprint arXiv:2502.13451*, 2025.

[5] X. Huang, H. Zhou, and K. Han, "Prunevid: Visual token pruning for efficient video large language models," *arXiv preprint arXiv:2412.16117*, 2024.

[6] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[7] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," *arXiv preprint arXiv:2010.07954*, 2020.

[8] S. Chen, P.-L. Guhur, C. Schmid, and I. Laptev, "History aware multimodal transformer for vision-and-language navigation," *Advances in Neural Information Processing Systems*, 2021.

[9] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "Think global, act local: Dual-scale graph transformer for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 537–16 547.

[10] S. Raychaudhuri, S. Wani, S. Patel, U. Jain, and A. X. Chang, "Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments," *arXiv preprint arXiv:2109.15207*, 2021.

[11] G. Georgakis, K. Schmeckpeper, K. Wanchoo, S. Dan, E. Miltsakaki, D. Roth, and K. Daniilidis, "Cross-modal map learning for vision and language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[12] P. Chen, D. Ji, K. Lin, R. Zeng, T. H. Li, M. Tan, and C. Gan, "Weakly-supervised multi-granularity map learning for vision-and-language navigation," *arXiv preprint arXiv:2210.07506*, 2022.

[13] D. An, H. Wang, W. Wang, Z. Wang, Y. Huang, K. He, and L. Wang, "Etpnav: Evolving topological planning for vision-language navigation in continuous environments," *arXiv preprint arXiv:2304.03047*, 2023.

[14] J. Krantz, E. Wijmans, A. Majundar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision and language navigation in continuous environments," in *European Conference on Computer Vision (ECCV)*, 2020.

[15] Z. Wang, J. Li, Y. Hong, Y. Wang, Q. Wu, M. Bansal, S. Gould, H. Tan, and Y. Qiao, "Scaling data generation in vision-and-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[16] Y. Long, X. Li, W. Cai, and H. Dong, "Discuss before moving: Visual language navigation via multi-expert discussions," *arXiv preprint arXiv:2309.11382*, 2023.

[17] P. Chen, X. Sun, H. Zhi, R. Zeng, T. H. Li, G. Liu, M. Tan, and C. Gan, "a^2 nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models," *arXiv preprint arXiv:2308.07997*, 2023.

[18] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong, "Instructnav: Zero-shot system for generic instruction navigation in unexplored environment," *arXiv preprint arXiv:2406.04882*, 2024.

[19] Y. Zhang, J. Wu, W. Li, B. Li, Z. Ma, Z. Liu, and C. Li, "Video instruction tuning with synthetic data," *arXiv preprint arXiv:2410.02713*, 2024.

[20] J. Lin, H. Yin, W. Ping, P. Molchanov, M. Shoeybi, and S. Han, "Vila: On pre-training for visual language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.

[21] Y. Li, C. Wang, and J. Jia, "Llama-vid: An image is worth 2 tokens in large language models," in *European Conference on Computer Vision*, 2024.

[22] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017.

[23] H. Tan, L. Yu, and M. Bansal, "Learning to navigate unseen environments: Back translation with environmental dropout," in *Proceedings of NAACL-HLT*, 2019.

[24] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang *et al.*, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," *arXiv preprint arXiv:2109.08238*, 2021.

[25] J. Krantz, A. Gokaslan, D. Batra, S. Lee, and O. Maksymets, "Waypoint models for instruction-guided navigation in continuous environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[26] Y. Hong, Z. Wang, Q. Wu, and S. Gould, "Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[27] J. Krantz and S. Lee, "Sim-2-sim transfer for vision-and-language navigation in continuous environments," in *European Conference on Computer Vision*, 2022.

[28] Z. Wang, X. Li, J. Yang, Y. Liu, and S. Jiang, "Gridmm: Grid memory map for vision-and-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[29] K. Chen, J. K. Chen, J. Chuang, M. Vázquez, and S. Savarese, "Topological planning with transformers for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[30] Z. Wang, X. Li, J. Yang, Y. Liu, and S. Jiang, "Sim-to-real transfer via 3d feature fields for vision-and-language navigation," *arXiv preprint arXiv:2406.09798*, 2024.

[31] Y. Zhang, J. Wu, W. Li, B. Li, Z. Ma, Z. Liu, and C. Li, "Video instruction tuning with synthetic data," 2024. [Online]. Available: https://arxiv.org/abs/2410.02713

[32] D. Azuma, T. Miyanishi, S. Kurita, and M. Kawanabe, "Scanqa: 3d question answering for spatial scene understanding," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[33] W. Zhu, J. Hessel, A. Awadalla, S. Y. Gadre, J. Dodge, A. Fang, Y. Yu, L. Schmidt, W. Y. Wang, and Y. Choi, "Multimodal C4: An open, billion-scale corpus of images interleaved with text," *arXiv preprint arXiv:2304.06939*, 2023.

[34] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei *et al.*, "Qwen2. 5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[35] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[36] Z. Zhu, X. Ma, Y. Chen, Z. Deng, S. Huang, and Q. Li, "3d-vista: Pre-trained transformer for 3d vision and text alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2911–2921.

[37] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan, "3d-llm: Injecting the 3d world into large language models," *Advances in Neural Information Processing Systems*, 2023.

[38] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang, "An embodied generalist agent in 3d world," *arXiv preprint arXiv:2311.12871*, 2023.

[39] H. Huang, Y. Chen, Z. Wang, R. Huang, R. Xu, T. Wang, L. Liu, X. Cheng, Y. Zhao, J. Pang *et al.*, "Chat-scene: Bridging 3d scene and large language models with object identifiers," *arXiv preprint arXiv:2312.08168*, 2023.

[40] R. Fu, J. Liu, X. Chen, Y. Nie, and W. Xiong, "Scene-llm: Extending language model for 3d visual understanding and reasoning," *arXiv preprint arXiv:2403.11401*, 2024.

[41] D. Zheng, S. Huang, L. Zhao, Y. Zhong, and L. Wang, "Towards learning a generalist model for embodied navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.