

MethodAndAnalyze (System2-Focused, Challenge-Driven)

```
> ##### Introduction #####“”-> #####-> #####-> “”-> #####”##### InternVLN  System2
#####
```

0. Intro

Challenge A: Viewpoint-Induced Reuse Failure □□□□□□□□ token □□□□

- Intro □□□□□□□□□□□□□□□□□□□□ token□
- □□□□□□ position-wise matching□□□□ view-aware □□□□□□

Challenge B: Instruction-Conditioned Semantic Drift

- Intro `XXXXXXXXXXXXXXXXXXXXX“XXXX”`
- `XXXXXX` instruction-guided refresh`XXXXXXXXXXXX`

...

1. Analyze

> □□□□□□□□□□“□□□□□□”□□□“□□□□□□”□

1.1 Challenge A VLN

A-Analysis-1: Token []-[]

- token
- /
-
-

A-Analysis-2: “□□□□ vs □□□□□□”□□□□□□

- 000000
- Baseline → position-wise matching
- Ours-Align → view-aware alignment + matching
- 000000 token 0000Recall@Reusable →

- 我们的Ours-Align 模块

A-Analysis-3: 模块分析

- 模块的输入输出
- 模块“做什么”

1.2 Challenge B 模块分析

B-Analysis-1: 模块分析

- 模块“做什么” cross-attention/grounding score
- 模块 landmark
- 模块/模块

B-Analysis-2: 模块 vs 模块

- 模块
- Fixed refresh K
- Semantic refresh
- 模块 + 模块
- 模块Semantic refresh

B-Analysis-3: 模块分析

- 模块/模块 vs 模块
- 模块

2. 模块Method

> 模块**VLN-Cache-S2**System2-only enhancement

2.1 Method-A: View-Aware Reuse Module Challenge A

模块

- System2 token view-aware

- 避免 token reuse

实现

- A1
- A2 $\text{depth} + \text{}$
- A3

System2 InternVLN

- ``internnav/model/basemodel/intervla_n1/intervla_n1_policy.py``
- ``s2_step(...)`` 返回
- ``internnav/agent/intervla_n1_agent.py``
- ``step(...)``

2.2 Method-B: Instruction-Guided Semantic Refresh Challenge B

实现

- refresh
- instruction-grounded saliency

实现

- B1
- B2 cross-attention
- B3

System2 InternVLN

- ``internnav/agent/intervla_n1_agent.py``
- ``should_infer_s2(...)``
- ``internnav/model/basemodel/intervla_n1/intervla_n1_policy.py``
- ``s2_step(...)`` saliency drift score

2.3 Method-C: Decode Budget Controller System2

環境

- 環境変数/コンフィグファイル `max_new_tokens` 環境変数で指定
- 環境変数で指定

環境

- C1環境変数で指定
- C2環境変数で指定 draft 環境speculative-lite環境
- C3環境変数draft+verify 環境 speculative decoding環境

System2 環境InternVLN

- `internnav/model/basemodel/internvla_n1/internvla_n1_policy.py`
- `generate(...)` 環境変数で指定
- `scripts/eval/configs/habitat_dual_system_cfg.py`
- 環境変数で指定

3. 環境変数“環境”環境変数

3.1 環境 S環境

- 環境 $A1 + B1 + C1$
- 環境変数で指定
- 環境変数で指定
- 環境変数で指定

3.2 環境 M環境

- 環境 $A2 + B2 + C1/C2$
- 環境変数-環境変数で指定
- 環境変数で指定
- 環境IROS 環境

3.3 環境 X環境

- 環境 $A3 + B3 + C3$
- 環境変数で指定

- 数据集/模型训练/推理
- 模型部署/模型更新

4. 模型部署与推理

4.1 模型部署

1. **Fig.1 Challenge Overview** A/B 模型部署挑战
2. **Fig.2 Viewpoint Mismatch Analysis** 模型 vs 模型部署挑战
3. **Fig.3 Semantic Drift Analysis** 模型部署 + refresh 模型
4. **Fig.4 Method Pipeline** A/B/C 模型部署 System2 模型部署
5. **Fig.5 Pareto Curve** 模型-模型部署 Latency vs SR/SPL

4.2 模型部署

1. **Tab.1 Main Results** NE/OS/SR/SPL + 模型 System2 模型
2. **Tab.2 Ablation by Challenge** Base + A + B + A + B + A + B + C
3. **Tab.3 Overhead Breakdown** 模型部署挑战

5. 模型部署“挑战”

5.1 模型部署挑战

- E0: Baseline 模型 System2
- E1: Baseline + Method-A 模型 Challenge A
- E2: Baseline + Method-B 模型 Challenge B
- E3: Baseline + A + B 模型
- E4: Baseline + A + B + C 模型

5.2 模型部署挑战

- 模型部署挑战/模型部署挑战
- 模型部署挑战/token
- 模型部署挑战SR/SPL/NE

5.3 实验设置

- 数据集SR, SPL, NE, OS
- 数据集System2 数据集 token S2 数据集
- 数据集A 数据集B 数据集/数据集

6. 实验结果

图 1 1-2 实验结果

- $C1 + B1$
- 数据集

图 2 2-3 实验结果

- $A2 + B2$
- 图 Fig.2 / Fig.3 / Tab.2

图 3 实验结果

- $C2$ $C3$ speculative 数据集

7. 实验结果 claim

- “System2 数据集”数据集“数据集”
- “layer-adaptive reuse” 数据集/数据集
- claim 数据集

8. 实验结果

图 1 实验结果

“For Challenge A, we introduce a view-aware reuse module that aligns cross-frame representations before token reuse, mitigating viewpoint-induced mismatch.”

Figure 2: Overview

“For Challenge B, we design an instruction-guided semantic refresh mechanism that updates cached regions only when task relevance shifts, reducing stale-cache errors.”

Figure 3: Architecture

“All modules are integrated into the System2 branch of InternVLN, leaving training setup unchanged and focusing on inference-time acceleration.”

9. Appendix

- Figure 4: “A2 + B2 + C1/C2” formula **M: A2 + B2 + C1/C2**
- Figure 5: Overview

- 1) Intro to the system
- 2) Overview of System2 architecture
- 3) Detailed description of the system