ICLR International Conference On Learning Representations

Bootstrapping Language-Guided Navigation Learning with Self-Refining Data Flywheel

Zun Wang[1,2]*

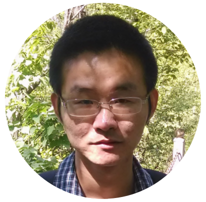
Jialu Li[2]

Yicong Hong[3]

Songze Li[1]

Kunchang Li[1]

Shoubin Yu[2]



Yi Wang[1,5]

Yu Qiao[1]

Yali Wang[1]

Mohit Bansal[2]

Limin Wang[1,4]

*Previously interned at Shanghai AI Laboratory

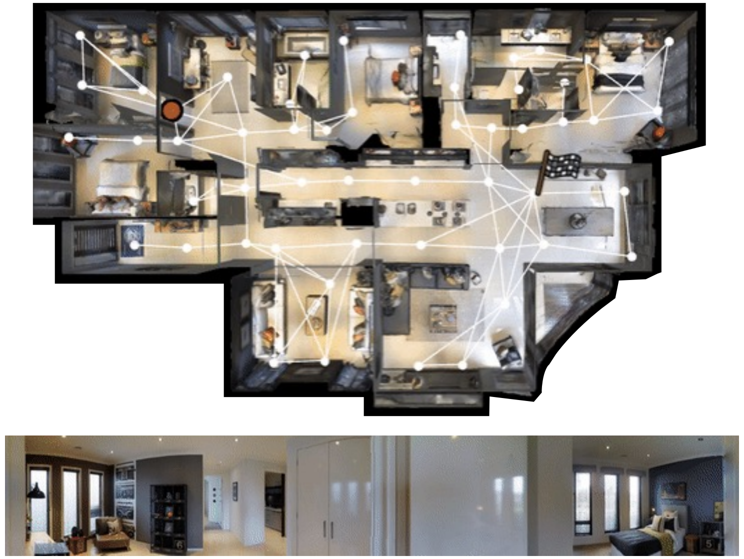
[1]Shanghai AI Laboratory [2]UNC, Chapel Hill [3]Adobe Research

[4]Nanjing University [5]Shanghai Innovation Institute

1. Background

- Vision-and-Language Navigation
  - Visual navigation following natural language instructions in unseen environments
  - Data scarcity problem: 14k instruction-trajectory pairs within 61 environments for training on R2R

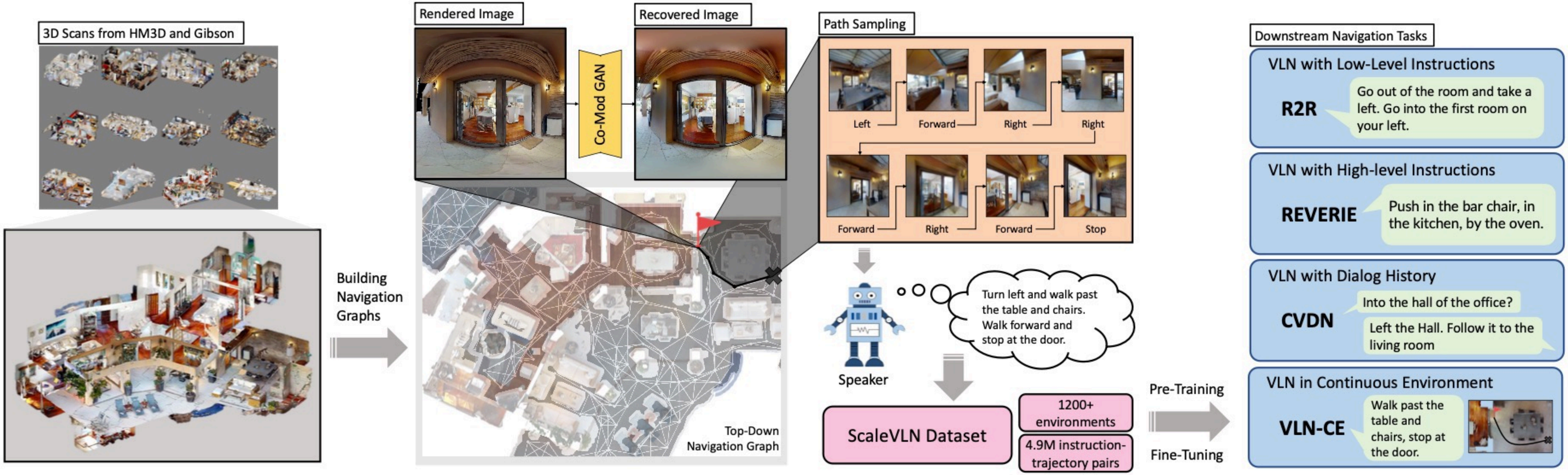
Instruction: Exit the bathroom and turn left. Walk past kitchen and stop by the dining table.



Demonstration of the Room-to-Room Vision-and-Language Navigation (R2R-VLN) Task.

1. Background

- Data Augmentation for VLN
  - ScaleVLN: Sampling path from unlabeled environments, then generate instructions with a trained path-to-instruction generator



1. Background

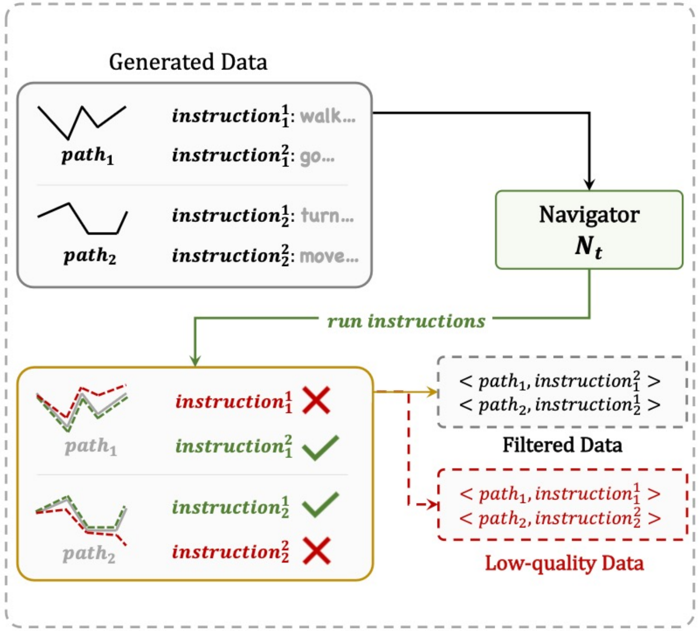
- Data Augmentation for VLN
  - ScaleVLN: Sampling path from unlabeled environments, then generate instructions with a trained path-to-instruction generator
  - However, Data quality is Low

Table 1: Performance (on R2R validation unseen split) on different datasets solely. Directly training with R2R yields the best SPL compared to training with other augmentation datasets.

Training Data	#data	#Env.	SR↑	SPL↑
R2R	14K	61	65.9	**55.9**
Prevalent	1.0M	60	**67.1**	54.8
ScaleVLN	4.9M	800	63.9	50.1

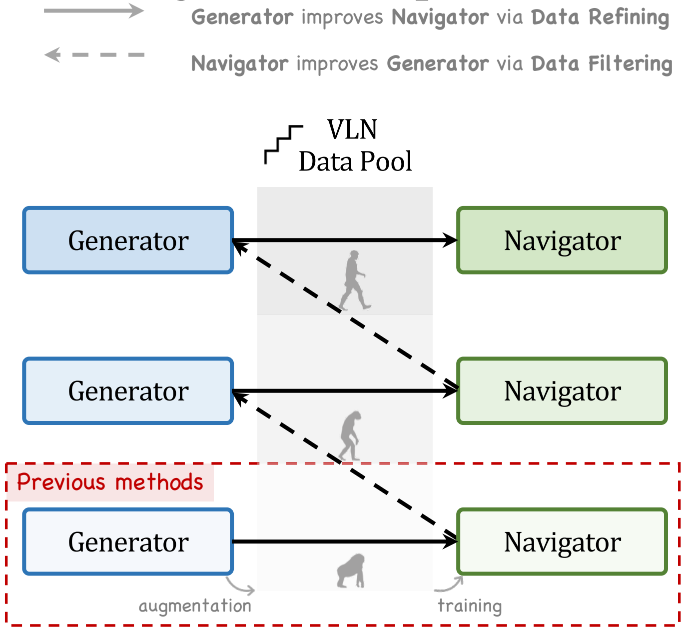
2. Self-Refining Data Flywheel

- How can we evaluate and improve language-guided navigational data?
- Evaluation: Self-evaluation
  - Using the trained navigator to re-run the instruction, It should follow the original path
  - Complex instruction-trajectory similarity -> Simple trajectory-trajectory similarity



(b) Filter Pipeline

- Improvement:
  - With the evaluation method, we can filter a high-quality subset, which can be used to improve the generator, while the improved generator can in turn improve the navigator, establishing a data loop



(a) *Pipeline*

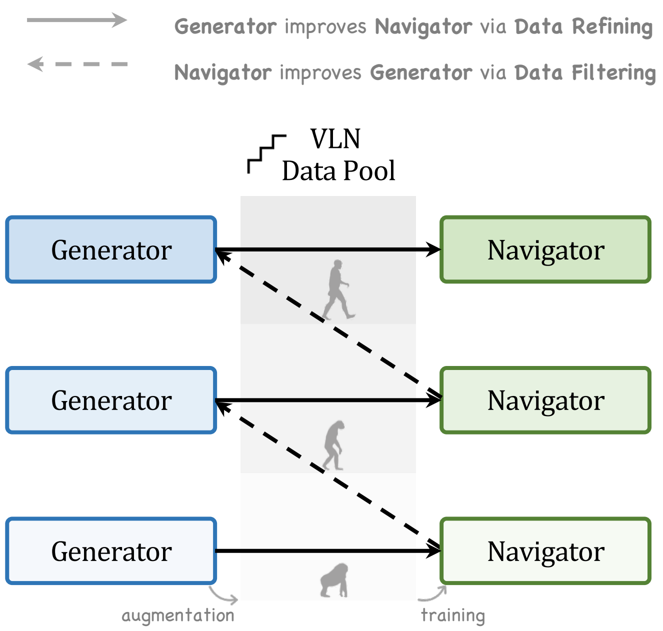
2. Self-Refining Data Flywheel

- Generate Data Pool via Base Instruction Generator
- Train Base Navigator with Generated Data
- Filter High-Quality Data using Trained Navigator
- Train Better Instruction Generator with Filtered Data
- Refine the Data Pool with Better Instruction Generator
- Train Better Navigator with the Refined Data Pool

Algorithm 1 Pipeline of Self-Refining Data Flywheel (SRDF)

Require: Seed data D_{Seed} (Human-annotated), Unlabelled trajectories D_{Traj}, Total iterations T.

Train base instruction generator G_1 with D_{Seed}.
Use G_1 to generate nav-training data D_1^N and gen-training data D_2^G for D_{Traj}.
/* D_1^N *is generated via random sampling while* D_1^G *via greedy decoding*
Train base navigator N_1 with D_1^N.
Use N_1 to filter high-quality subsets FD_2^G from D_2^G and $FD_{<2}^N$ from D_1^N.
for each iteration t $(1 < t \leqslant T)$ **do**
    /* *Note: Seed data* D_{Seed} *is used in training stages of both* G_t *and* N_t *but omitted for simplicity*
    Train generator G_t with FD_t^G.
    Use G_t to generate nav-training data ND_t^N for LD_t^N and gen-training data D_{t+1}^G for D_{Traj}.
    /* ND_t^N *is generated via random sampling while* D_{t+1}^G *via greedy decoding*
    Combine ND_t^N and $FD_{<t}^N$ to form D_t^N.
    Train navigator N_t with D_t^N.
    Use N_t to filter high-quality subsets FD_{t+1}^G from D_{t+1}^G and FND_t^N from ND_t^N.
    Combine FND_t^N and $FD_{<t}^N$ to form $FD_{<t+1}^N$.
end for



(a) *Pipeline*

2. Self-Refining Data Flywheel

- Statistics of Generated Dataset (3-round flywheel running)

Dataset	Instruction	#Env.	#Instr.	#Vocab.	Instr. Length
R2R (Anderson et al., 2018b)	Manually Labelled	61	14,039	3,063	26.33
RxR-en (Ku et al., 2020)		60	26,464	7,249	102.13
REVERIE (Qi et al., 2020)		60	10,466	1,140	18.64
CVDN (Thomason et al., 2020)		57	4,742	2,068	53.21
SOON (Zhu et al., 2021)		34	2,780	735	44.09
R4R (Zhu et al., 2020)		59	233,532	3,004	52.25
Prevalent (Hao et al., 2020)	Generated	60	1,069,620	993	24.23
Marky (Wang et al., 2022b)		60	333,777	2,231	99.45
AutoVLN (Chen et al., 2022c)		900	217,703	1,696	20.52
ScaleVLN (Wang et al., 2023e)		1289	4,941,710	172	21.61
SRDF-20M (Ours)		860	20,417,874	10,363	24.05

- Downstream Datasets
  - Fine-grained VLN (R2R)
  - High-level VLN (REVERIE, SOON)
  - Long-horizon VLN (R4R, RxR-english)
  - Dialog-based VLN (CVDN)
  - VLN in continuous environment (R2R-CE)
  - VLN instruction generation (R2R)

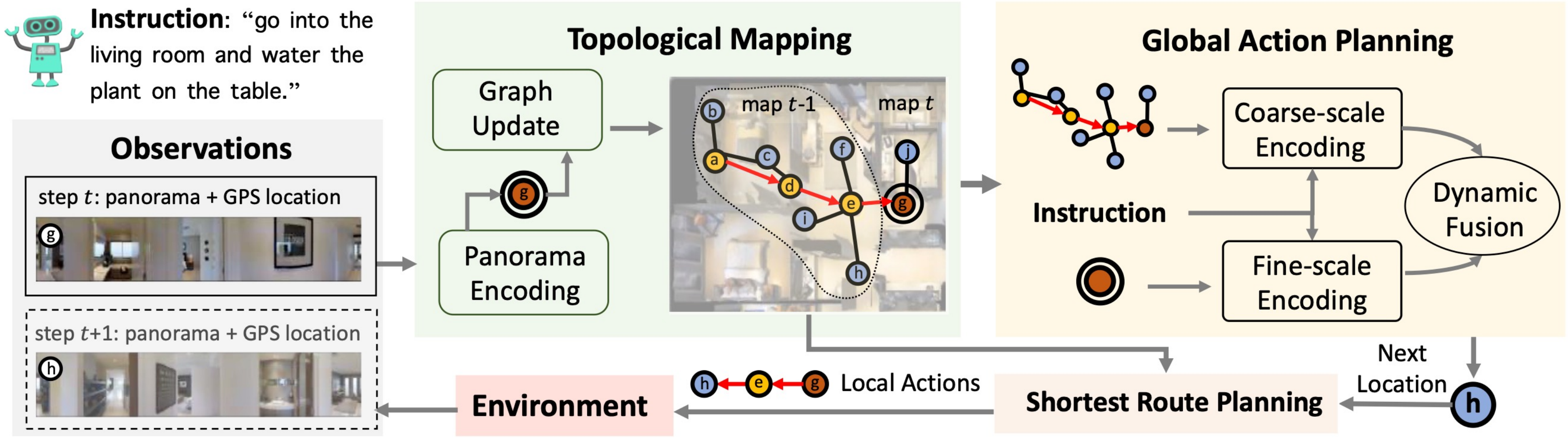
2. Self-Refining Data Flywheel

- Implementation
  - The path to instruction generator is finetuned using an interleaved MLLM, Mantis, with LoRA

```
This is a navigation trajectory consists of several image sequences:
(Viewpoint 1: Image: <image>, Action: right (175.49 degree) and up (1.03 degree)),
(Viewpoint 1: Image: <image>, Action: forward),
(Viewpoint 2: Image: <image>, Action: forward),
(Viewpoint 3: Image: <image>, Action: left (16.44 degree) and down (0.12 degree)),
(Viewpoint 3: Image: <image>, Action: forward),
(Viewpoint 4: Image: <image>, Action: left (52.83 degree) and down (0.14 degree)),
(Viewpoint 4: Image: <image>, Action: forward),
(Viewpoint 5: Image: <image>, Action: forward),
(Viewpoint 6: Image: <image>, Action: right (83.02 degree) and down (0.53 degree)),
(Viewpoint 6: Image: <image>, Action: forward),
(Viewpoint 7: Image: <image>, Action: stop).

Could you give me its corresponding navigation instruction in details?
```

  - Navigator based on DUET



3. Experiments

- Multi-Round Flywheel Running
  - Generator improves Navigator via Data Refining
  - Navigator improves Generator via Data Filtering

Table 3: Navigator and instruction generator results in different rounds.

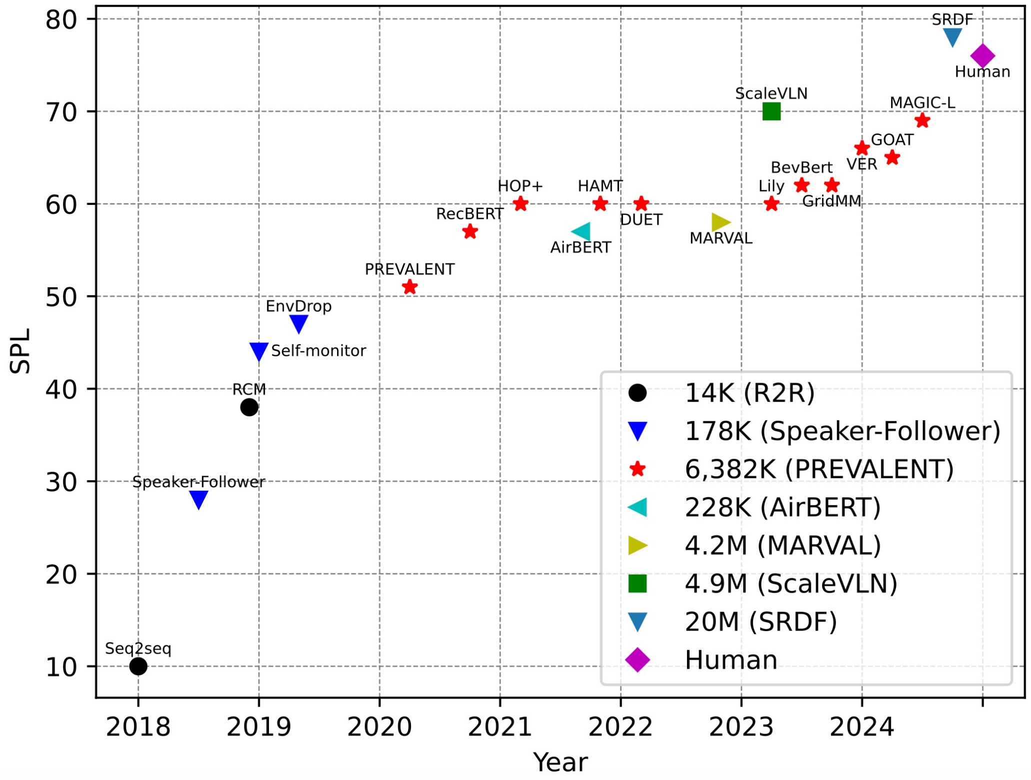
Method	Instruction Following				Instruction Generation						
	NE↓	OSR↑	SR↑	SPL↑	SPICE↑	SPICE-D↑	Bleu-1↑	Bleu-4↑	CIDEr↑	Meteor↑	Rouge↑
Baseline	2.37	85.5	78.6	69.9	21.8	28.0	72.5	27.7	42.2	23.6	49.0
Ours (round 1)	1.95	87.1	82.4	75.9	23.7	28.4	71.4	29.5	46.5	23.1	50.2
Ours (round 2)	1.81	88.5	83.6	77.3	25.2	29.9	73.7	**31.0**	**50.7**	24.2	51.3
Ours (round 3)	**1.76**	**89.6**	**84.4**	**77.6**	**25.7**	**30.4**	**74.5**	30.8	49.7	**24.5**	**51.3**

3. Experiments

- Downstream Navigation Tasks (R2R + R2R-CE)

Table 7: Comparison of single-run performance on R2R and R2R-CE datasets. Best results are marked in **bold blue** and second best in **bold**.

Methods	Room-to-Room Dataset								Room-to-Room-CE Dataset					
	Validation-Unseen				Test-Unseen				Validation-Unseen			Test-Unseen		
	NE↓	OSR↑	SR↑	SPL↑	NE↓	OSR↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	NE↓	SR↑	SPL↑
Human	-	-	-	-	1.61	90	86	76	-	-	-	-	-	-
Seq2Seq (Anderson et al., 2018b)	7.81	28	21	-	7.85	27	20	-	-	-	-	-	-	-
Speaker Follower (Fried et al., 2018)	6.62	45	36	-	6.62	-	35	28	-	-	-	-	-	-
RCM (Wang et al., 2019)	6.09	50	43	-	6.12	50	43	38	-	-	-	-	-	-
EnvDrop (Tan et al., 2019)	5.22	-	52	48	5.23	59	51	47	-	-	-	-	-	-
PREVALENT (Hao et al., 2020)	4.71	-	58	53	5.30	61	54	51	-	-	-	-	-	-
NvEM (An et al., 2021)	4.27	-	60	55	4.37	-	58	54	-	-	-	-	-	-
AirBert (Guhur et al., 2021)	4.10	-	62	56	4.13	-	62	57	-	-	-	-	-	-
VLN↻BERT (Hong et al., 2021)	3.93	-	63	57	4.09	70	63	57	-	-	-	-	-	-
HAMT (Chen et al., 2021)	2.29	-	66	61	3.93	72	65	60	-	-	-	-	-	-
HOP (Qiao et al., 2022)	3.80	-	64	57	3.83	-	64	59	-	-	-	-	-	-
HOP+ (Qiao et al., 2023a)	3.49	-	67	61	3.71	-	66	60	-	-	-	-	-	-
DUET (Chen et al., 2022b)	3.31	81	72	60	3.65	76	69	59	-	-	-	-	-	-
Lily (Lin et al., 2023)	2.90	-	74	62	3.44	-	72	60	-	-	-	-	-	-
DreamWalker (Wang et al., 2023a)	-	-	-	-	-	-	-	-	5.53	49	44	5.48	49	44
BEVBert (An et al., 2023a)	2.81	84	75	64	3.13	81	73	62	4.57	59	50	**4.70**	**59**	**50**
ScaleVLN (Wang et al., 2023e)	**2.09**	**88**	**81**	**70**	**2.27**	**86**	**80**	**70**	4.80	55	51	5.11	55	50
GridMM (Wang et al., 2023d)	2.83	-	75	64	3.13	-	73	62	5.11	49	41	5.64	46	39
ETPNav (An et al., 2023b)	-	-	-	-	-	-	-	-	4.71	57	49	5.12	55	48
DualAction (Zhang & Kordjamshidi, 2024)	-	-	-	-	-	-	-	-	-	58	49	-	56	48
HNR (Wang et al., 2024e)	-	-	-	-	-	-	-	-	**4.42**	**61**	**51**	4.81	58	50
NaviLLM (Zheng et al., 2024)	3.51	-	67	59	3.71	-	68	60	-	-	-	-	-	-
NavGPT-2 (Zhou et al., 2024a)	2.84	84	74	61	3.33	80	72	60	-	-	-	-	-	-
VER (Liu et al., 2024)	2.80	-	76	65	2.74	-	76	66	-	-	-	-	-	-
MAGIC-L (Wang et al., 2024b)	2.22	86	79	**70**	2.75	82	77	69	-	-	-	-	-	-
GOAT (Wang et al., 2024a)	2.40	85	78	68	3.04	80	75	65	-	-	-	-	-	-
SRDF (Ours)	**1.62**	**90**	**86**	**79**	**1.82**	**89**	**85**	**78**	**4.12**	**65**	**57**	**4.35**	**64**	**56**



R2R-SoTA Timeline

3. Experiments

- Downstream Navigation Tasks (Others)

Table 8: Comparison with previous methods on various downstream tasks. † indicates the RxR-en results are reproduced using their officially released checkpoints. ⋆ means pre-exploration methods. Best results are marked in **bold blue** and second best in **bold**.

Methods	RxR-english		R4R		CVDN		REVERIE				SOON			
	Val unseen		Val unseen		Val	Test	Val unseen		Test unseen		Val unseen		Test unseen	
	SR↑	nDTW↑	SR↑	sDTW↑	GP↑	GP↑	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
HAMT† (Chen et al., 2021)	56.4	63.0	44.6	31.8	5.13	5.58	33.0	30.2	30.4	26.7	-	-	-	-
MARVAL (Kamath et al., 2022)	64.7	70.4	-	-	-	-	-	-	-	-	-	-	-	-
DUET (Chen et al., 2022b)	-	-	-	-	-	-	47.0	33.7	52.5	36.1	36.3	22.6	33.4	21.4
AutoVLN (Chen et al., 2022c)	-	-	-	-	-	-	55.9	40.9	55.2	38.9	**41.0**	**30.7**	40.4	**27.9**
RREx-Bot⋆ (Sigurdsson et al., 2023)	-	-	-	-	-	-	61.0	58.8	65.9	62.0	49.2	48.6	47.5	47.1
BEVBert† (An et al., 2023a)†	66.7	69.6	-	-	-	-	51.8	36.4	52.8	36.4	-	-	-	-
KERM (Li et al., 2023b)	-	-	-	-	-	-	50.4	35.4	52.4	39.2	38.1	23.2	-	-
ScaleVLN (Wang et al., 2023e)	-	-	-	-	6.12	6.97	57.0	41.9	56.1	39.5	-	-	-	-
PanoGen (Li & Bansal, 2023)	-	-	**47.8**	-	5.93	7.17	-	-	-	-	-	-	-	-
BSG (Liu et al., 2023)	-	-	47.0	**34.0**	-	-	52.1	35.6	56.5	38.7	-	-	-	-
MiC (Qiao et al., 2023b)	-	-	-	-	-	-	57.0	43.6	55.7	42.0	-	-	-	-
NaviLLM (Zheng et al., 2024)	-	-	-	-	**6.16**	**7.90**	44.6	36.6	43.5	34.5	38.3	29.2	35.0	26.2
VER (Liu et al., 2024)	-	-	47.0	33.0	-	-	56.0	39.7	56.8	38.8	-	-	-	-
PRET† (Lu et al., 2024)	71.0	**70.9**	-	-	-	-	-	-	-	-	-	-	-	-
MAGIC-L (Wang et al., 2024b)	**72.9**	68.1	-	-	-	-	-	-	-	-	-	-	-	-
VLN-Copilot (Qiao et al., 2024)	-	-	-	-	-	-	**57.4**	**43.6**	**57.8**	**42.3**	-	-	-	-
GOAT (Wang et al., 2024a)	68.2	66.8	-	-	-	-	53.4	36.7	57.7	40.5	40.4	28.1	**40.5**	25.2
SRDF (Ours)	**78.8**	**74.4**	**64.4**	**44.6**	**7.67**	**8.19**	**60.4**	**45.4**	**61.4**	**47.7**	**50.3**	**41.7**	**46.6**	**37.9**

3. Experiments

- R2R Instruction Generation

Table 9: Performance of different instruction generators on R2R. † means reproduced results. Best results are marked in **bold blue**, second best in **bold**, and third best in underlined.

Methods	R2R Validation Unseen						
	SPICE↑	SPICE-D↑	Bleu-1↑	Bleu-4↑	CIDEr↑	Meteor↑	Rouge↑
BT-speaker (Fried et al., 2018)	18.9	25.1	68.2	26.3	37.9	21.7	48.0
LandmarkSelect (Agarwal et al., 2019)	19.7	-	54.8	15.9	13.2	23.1	35.7
EnvDrop (Tan et al., 2019)	21.8	28.0	72.3	27.1	41.7	23.6	49.0
CCC (Wang et al., 2022a)	21.4	27.8	70.8	27.2	46.1	23.1	47.7
FOAM† (Dou & Peng, 2022)	21.7	28.1	72.5	27.3	42.4	23.4	49.2
KEFA (Zeng et al., 2023)	23.4	29.3	<u>73.8</u>	28.3	42.7	<u>24.4</u>	50.3
LANA (Wang et al., 2023b)	22.6	-	73.6	28.9	45.7	23.7	49.8
LANA+ (Wang et al., 2023c)	22.8	-	73.2	29.5	46.0	24.1	49.6
C-Instructor (Kong et al., 2024)	21.2	-	71.3	26.6	44.7	23.9	47.3
BEVInsructor (Fan et al., 2024)	20.8	-	69.9	26.4	44.9	23.0	46.7
SRDF (Ours, round 2)	<u>25.2</u>	<u>29.9</u>	73.7	**31.0**	**50.7**	24.2	**51.3**
SRDF (Ours, round 3)	**25.7**	**30.4**	**74.5**	<u>30.8</u>	**49.7**	**24.5**	**51.3**
SRDF (Ours, round 3 fine-tuned w/ FD_4^G)	**26.2**	**30.9**	**75.3**	**31.1**	<u>49.2</u>	**25.0**	**51.4**

3. Experiments

- Analysis
  - Navigator and Generator Improves Each Other
  - Scalability: Increasing training environments and instruction diversity improves performance

Table 3: Navigator and instruction generator results in different rounds in R2R val unseen split.

Method	Instruction Following				Instruction Generation						
	NE↓	OSR↑	SR↑	SPL↑	SPICE↑	SPICE-D↑	Bleu-1↑	Bleu-4↑	CIDEr↑	Meteor↑	Rouge↑
Baseline	2.37	85.5	78.6	69.9	21.8	28.0	72.5	27.7	42.2	23.6	49.0
Ours (round 1)	1.95	87.1	82.4	75.9	23.7	28.4	71.4	29.5	46.5	23.1	50.2
Ours (round 2)	1.81	88.5	83.6	77.3	25.2	29.9	73.7	**31.0**	**50.7**	24.2	**51.3**
Ours (round 3)	**1.76**	**89.6**	**84.4**	**77.6**	**25.7**	**30.4**	**74.5**	30.8	49.7	**24.5**	**51.3**

Table 5: Results of instruction diversity (#instr. per path) in navigator training in val unseen split.

Aug Data	NE↓	SR↑	SPL↑
Prev $_{\#Instr=1}$	3.21	71.86	61.04
Ours $_{\#Instr=1}$	**2.97**	**73.86**	**63.58**
Prev $_{\#Instr=3}$	3.12	72.67	62.53
Ours $_{\#Instr=3}$	**2.81**	**75.21**	**64.56**
Prev $_{\#Instr=6}$	3.07	72.84	63.12
Ours $_{\#Instr=6}$	**2.55**	**76.93**	**66.89**
Ours $_{\#Instr=12}$	2.59	77.05	66.53

Table 6: Results of different additional augmentation data in instruction generator training in val unseen split.

Additional Data	SPICE↑	Bleu-4↑	CIDEr↑	Rouge↑
-	23.7	29.5	46.5	50.2
ScaleVLN Data	23.5	29.0	46.1	49.8
Prevalent Data	23.6	29.3	46.7	50.1
100-HM3D-Env Ours	23.9	29.8	47.8	50.0
200-HM3D-Env Ours	24.2	30.1	49.1	50.3
400-HM3D-Env Ours	24.6	30.3	48.9	50.7
800-HM3D-Env Ours	**25.2**	**31.0**	**50.7**	**51.3**
800-HM3D-Env Ours (Sample)	24.8	30.3	48.3	51.0

3. Summary

- Contributions
  - building a novel VLN data-self-improving framework for the first time that iteratively improves the navigator and generator with their mutual feedback to create a substantially high-quality VLN dataset
  - substantially strong/SoTA results over eight challenging VLN tasks, even surpassing/approaching humans in some cases

- Future Work
  - Iteratively introducing more environments to support more loop
  - Applying Self-Refining Data Flywheel to more embodied tasks