Trajectory-Diversity-Driven Robust Vision-and-Language Navigation

Supplementary Material

1. GRPO Variants Implementation

We provide detailed formulations of the group-based policy optimization variants evaluated in our experiments. All variants share the same trajectory sampling strategy but differ in advantage estimation and policy update mechanisms.

GRPO [2]. Uses group-wise advantage normalization with both variance and length normalization:

$$\hat{A}_k = \frac{r_k - \text{mean}(\{r_1, ..., r_K\})}{\text{std}(\{r_1, ..., r_K\})} \tag{1}$$

The policy objective aggregates with length normalization:

$$\mathcal{J}_{\text{GRPO}} = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\frac{1}{|\tau_k|}\sum_{t=1}^{|\tau_k|}\rho_{k,t}\hat{A}_k\right] \tag{2}$$

where $\rho_{k,t}(\theta) = \pi_\theta(a_{k,t}|s_{k,t})/\pi_{\text{old}}(a_{k,t}|s_{k,t})$ with clipping omitted for brevity. Variance normalization ensures consistent gradient scales across batches, while the $1/|\tau_k|$ term computes per-token average gradients. Token-level importance sampling applies independent ratio clipping at each time step.

GSPO [5]. Shifts importance sampling to sequence level. The sequence-level importance ratio is:

$$s_k(\theta) = \exp\left(\frac{1}{|\tau_k|}\sum_{t=1}^{|\tau_k|}\log\frac{\pi_\theta(a_{k,t}|s_{k,t})}{\pi_{\text{old}}(a_{k,t}|s_{k,t})}\right) \tag{3}$$

The objective applies trajectory-level clipping:

$$\mathcal{J}_{\text{GSPO}} = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}s_k\hat{A}_k\right] \tag{4}$$

The sequence-level importance ratio s_k aggregates token probabilities across the entire trajectory before clipping. This can be overly aggressive as it discards all tokens once triggered.

GMPO [4]. Replaces arithmetic mean with geometric mean for token-level rewards, computed in log space for numerical stability. The objective is:

$$\mathcal{J}_{\text{GMPO}} = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\text{sgn}(\hat{A}_k)\cdot\left(\prod_{t=1}^{|\tau_k|}|\rho_{k,t}\hat{A}_k|\right)^{\frac{1}{|\tau_k|}}\right] \tag{5}$$

where clipping is omitted for brevity. The geometric mean inherently suppresses outliers, yielding more stable importance sampling ratios. GMPO employs token-level clipping with an exponential range $(e^{-\epsilon}, e^{\epsilon})$ that is wider than GRPO's linear range $(1-\epsilon, 1+\epsilon)$.

2. Dr.GRPO for VLN

2.1. Background: Standard GRPO vs. Dr.GRPO

Standard GRPO [2], designed for language model alignment, applies two normalization terms to the advantage estimation:

$$\hat{A}_k^{\text{GRPO}} = \frac{1}{|\tau_k|}\cdot\frac{r_k - \text{mean}(\{r_1, ..., r_K\})}{\text{std}(\{r_1, ..., r_K\})} \tag{6}$$

where $|\tau_k|$ is the trajectory length and $\text{std}(\{r_1, ..., r_K\})$ is the standard deviation of rewards across K rollouts. While these normalizations are appropriate for open-ended text generation, they introduce optimization biases detrimental to navigation learning.

Dr.GRPO [3] eliminates both normalizations through unbiased advantage estimation:

$$\hat{A}_k = r_k - \text{mean}(\{r_1, ..., r_K\}) \tag{7}$$

and aggregates gradients without normalization:

$$\mathcal{J}_{\text{Dr.GRPO}} = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{t=1}^{|\tau_k|}\rho_{k,t}\hat{A}_k\right] \tag{8}$$

This recovers the theoretically grounded Monte Carlo policy gradient with an unbiased baseline.

2.2. Why Standard GRPO Fails for VLN

The two normalization terms in standard GRPO create specific problems for embodied navigation tasks:

Length Normalization Bias. The $1/|\tau_k|$ term divides the advantage by trajectory length to treat each token equally in language model training. However, in VLN, trajectory length naturally reflects scene complexity and navigation difficulty—longer paths may traverse larger buildings or require more precise localization. Empirically, R2R trajectories range from 5 to 15+ meters with an average of approximately 10 meters, exhibiting substantial variance. This creates asymmetric gradient updates:

- **Correct short paths** receive amplified gradients ($\propto 1/\text{short}$), encouraging shortcuts even when inappropriate.
- **Incorrect long paths** receive diluted penalties ($\propto 1/\text{long}$), failing to discourage wandering behavior when the agent is lost.

Variance Normalization Bias. The $1/\text{std}(\{r_1, ..., r_K\})$ term amplifies gradients for instructions where all K rollouts yield similar outcomes (low variance). This assigns disproportionate weight to:

Algorithm 1 Dr.GRPO for VLN

Require: Pretrained policy π_{ref}, instruction set $\mathcal{D}$, Batch size B, group size K, hyperparameters α, δ, β

Initialize $\pi_\theta \leftarrow \pi_{\text{ref}}$, $\mathcal{B}_{\text{hard}} \leftarrow \emptyset$
for iteration $= 1, 2, \ldots$ **do**
  Sample $\{W_i\}_{i=1}^{B} \sim \mathcal{D}$; $\pi_{\text{old}} \leftarrow \pi_\theta$
  for each W_i **do**
    for $k = 1$ to K **do**
      Sample $\tau_k \sim \pi_{\text{old}}(\cdot|W_i)$; compute r_k (Eq. 8)
    end for
    Compute $\hat{A}_k = r_k - \frac{1}{K}\sum_{j=1}^{K} r_j$ for all k
    if $\sum_{k=1}^{K} \mathbb{I}(d_k < \epsilon) = 0$ **then**
      $\mathcal{B}_{\text{hard}} \leftarrow \mathcal{B}_{\text{hard}} \cup \{W_i\}$
    end if
  end for
  Update π_θ via gradient ascent (Eq. 5)
  if $|\mathcal{B}_{\text{hard}}| \geq M$ **then**
    SFT on $\mathcal{B}_{\text{hard}}$ (Eq. 11); $\mathcal{B}_{\text{hard}} \leftarrow \emptyset$
  end if
end for
return π_θ

- **'Easy' instructions** where all rollouts succeed (low learning value).
- **'Hard' instructions** where all rollouts fail (no positive signal to exploit).

Both scenarios provide minimal actionable gradients, yet receive higher weight than instructions with mixed success/failure (high learning value). For VLN, where learning should prioritize instructions that distinguish successful from failed behaviors, this variance-based weighting is counterproductive.

By removing both normalizations, Dr.GRPO ensures that gradient magnitudes reflect true action importance rather than being artificially modulated by path length or scenario-specific variance, yielding more principled policy updates and improved sample efficiency.

Algorithm 1 presents the complete training procedure.

3. Additional Experiments

3.1. Comparison with More RL Methods

We compare with HAMT [1], a representative value-based RL approach that applies policy gradient optimization with auxiliary value networks for vision-language navigation. We implement HAMT's training pipeline on ScaleVLN under identical training data and computational budget to ensure fair comparison.

Table 1 demonstrates NavGRPO's clear advantage over HAMT across all settings. Under standard conditions, NavGRPO achieves +6.53% SPL improvement. Under perturbations, NavGRPO exhibits substantially stronger robustness: degrading 1.6× less than HAMT under global perturbation and 1.5× less under early perturbation.

More revealing is the comparison across learning paradigms. While HAMT improves robustness over imitation learning, degrading roughly half as much as ScaleVLN under global perturbations, it suffers severe success rate collapse under extreme scenarios. HAMT's success rate drops 7.82% under 2-step early perturbation compared to ScaleVLN's 3.71% drop, revealing that traditional value-based RL struggles to maintain task completion capability when facing critical early errors.

The performance gap stems from two fundamental differences. First, value-based methods require training critic networks to approximate state values, which suffers from instability in VLN's high-dimensional action space. Second, traditional RL optimizes trajectories independently using absolute rewards, learning what is "correct." NavGRPO instead performs group-relative optimization—sampling diverse rollouts per instruction and learning through within-group comparison. This enables distinguishing relative quality among strategies under identical instructions, extracting supervision from both successful and failed trajectories that value-based methods cannot capture when optimizing trajectories in isolation.

Table 1. Comparison with value-based RL method (HAMT) under standard and perturbed settings on R2R Val Unseen. Both methods build upon ScaleVLN baseline.

Method	OSR↑	NE↓	SR↑	SPL↑	ΔSPL
Standard (No Perturbation)					
ScaleVLN + HAMT	83.07	2.65	74.98	66.12	-0.00
ScaleVLN + NavGRPO	**89.18**	**2.19**	**81.88**	**72.65**	**-0.00**
Global Perturbation ($p = 0.8$)					
ScaleVLN + HAMT	82.14	2.83	72.01	61.92	-4.20
ScaleVLN + NavGRPO	**88.49**	**2.27**	**79.95**	**69.98**	**-2.67**
Early Perturbation (Steps=2)					
ScaleVLN + HAMT	75.24	3.09	67.16	55.15	-10.97
ScaleVLN + NavGRPO	**83.85**	**2.70**	**76.50**	**65.10**	**-7.55**

Table 2. Ablation study on path efficiency weight α in R_{path} on R2R Val Unseen split.

α	OSR↑	NE↓	SR↑	SPL↑
0.0	88.64	2.26	**81.92**	71.48
0.25	**89.18**	**2.19**	81.88	**72.65**
0.5	88.71	2.23	81.34	71.92
1.0	88.23	2.31	80.76	70.85

Table 3. Ablation study on entropy regularization coefficient β on R2R Val Unseen split.

β	OSR↑	NE↓	SR↑	SPL↑
0.0	88.52	2.27	81.64	72.28
0.01	**89.18**	**2.19**	**81.88**	**72.65**
0.05	88.73	2.24	81.52	71.84
0.1	88.21	2.32	80.92	70.56

Table 4. Impact of hard case buffer size M on R2R Val Unseen split. All variants substantially outperform Sequential SFT-RL baseline.

Training Strategy	OSR↑	NE↓	SR↑	SPL↑
Sequential SFT-RL	88.95	2.22	81.25	72.08
Ours (M=100)	88.89	2.23	81.64	72.41
Ours (M=200)	89.18	2.19	81.88	72.65
Ours (M=400)	89.02	2.20	81.77	72.53

3.2. Extended Ablation Studies

Effect of Path Efficiency Weight α. Table 2 demonstrates the impact of path efficiency weight α in R_{path}. Without path guidance at $\alpha = 0$, the agent achieves highest success rate but suffers 1.17 SPL degradation due to inefficient detours, revealing the trade-off between task completion and path quality. Our default $\alpha = 0.25$ strikes optimal balance, jointly optimizing both metrics. As α increases, performance degrades monotonically: $\alpha = 1.0$ causes 1.8 SPL and 1.12 SR drops. This pattern reveals a limitation of excessive path efficiency enforcement. When deviations from oracle trajectories are heavily penalized, the agent develops brittle behavior that rigidly follows expected routes. In scenarios requiring adaptive replanning around obstacles or unexpected states, such rigidity leads to navigation failure rather than flexible recovery.

Robustness to Clipping and Entropy Regularization. We examine the sensitivity of our method to the entropy regularization coefficient β, as shown in Table 3. Starting from a well-performing SFT policy, we find that a small amount of entropy regularization with $\beta = 0.01$ slightly improves performance compared to no regularization at $\beta = 0.0$. This can be attributed to the entropy term providing a mild regularization effect that prevents the policy from deviating too aggressively from the initial SFT policy during RL fine-tuning, thereby preserving the beneficial behaviors learned from supervised demonstrations. However, when β is set too high at 0.1, performance degrades significantly. Excessive entropy regularization overly restricts the policy's ability to explore and deviate from the SFT initialization, limiting the potential improvements that RL can achieve through reward-driven optimization. The relatively small performance gap between $\beta = 0.0$ and $\beta = 0.01$ suggests that our training framework strikes a good balance without requiring strong entropy constraints.Regarding the clipping threshold δ, we do not conduct extensive ablation experiments as our training procedure accumulates gradients across all samples within a single rollout before performing a unified policy update. This ensures that the policy update remains on-policy, and the gradient accumulation naturally stabilizes the update magnitude without requiring aggressive clipping. We use the standard value $\delta = 0.2$ throughout our experiments.

Table 5. Computational cost comparison between ScaleVLN and NavGRPO on R2R training.

	ScaleVLN	Ours(K=4)	Ours(K=8)
NE↓	2.34	2.28	2.19
SR↑	79.40	80.37	81.88
SPL↑	69.97	71.98	72.65
FLOPs(G)	4.95	4.95	4.95
Batchsize	16	8	8
Rollouts per Sample	2	4	8
Peak GPU Memory(GB)↓	20.274	14.788	28.652
Training Time(min/1k steps)↓	46	51	86

Hard Case Buffer Size. Table 4 shows the impact of hard case buffer size M. Performance remains stable across $M \in \{100, 200, 400\}$, with minimal variation in SR and SPL. This robustness stems from two factors: first, hard case updates are triggered only when all K sampled trajectories fail, making the actual update frequency relatively low regardless of buffer size; second, our GRPO training already maintains reasonable coverage of diverse navigation scenarios, reducing reliance on explicit hard case intervention. We include this mechanism primarily as a training stabilizer to prevent catastrophic forgetting on persistently challenging instructions, rather than as a core algorithmic component. We adopt M = 200 as a conservative choice that balances memory overhead and potential benefit without introducing additional hyperparameter sensitivity.

4. Computational Cost Analysis

Table 5 presents a comprehensive computational cost comparison between ScaleVLN and NavGRPO on a single NVIDIA A800 GPU. NavGRPO with K=4 demonstrates superior memory efficiency, consuming only 14.788 GB of peak GPU memory compared to ScaleVLN's 20.274 GB, despite generating twice the number of rollouts per sample. This efficiency stems from our decoupled sampling-training architecture: the sampling phase computes visual feature embeddings and graph representations without maintaining computational graphs, caching only the processed features, while the training phase reconstructs gradients solely through the policy network using these cached representations, thereby eliminating redundant forward passes for visual feature computation and substantially reducing the

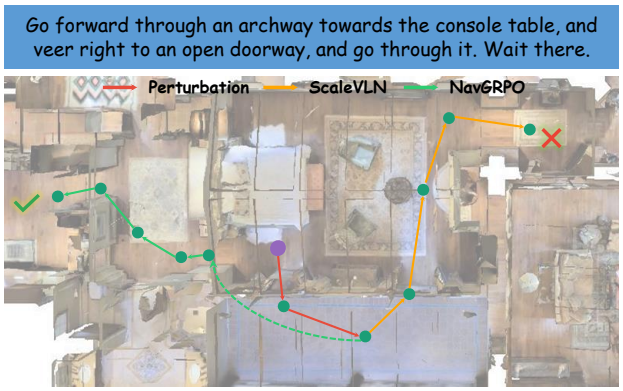

Figure 1. Qualitative comparison of navigation trajectories. Purple point indicates the starting location. NavGRPO's trajectory (green) successfully completes the instruction by navigating through the archway and reaching the target doorway, while ScaleVLN's trajectory (yellow) fails to execute the complete route, demonstrating NavGRPO's superior robustness against environmental complexity.

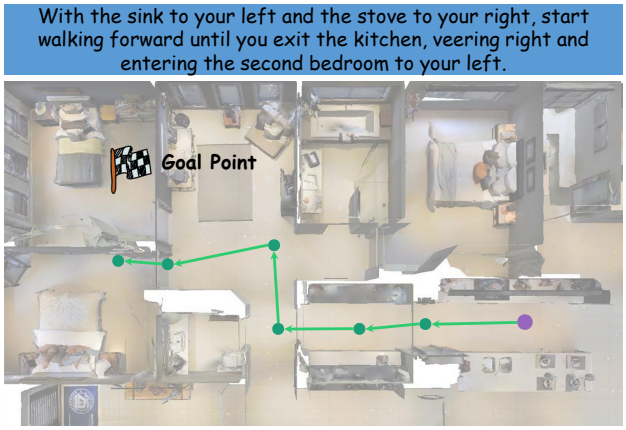

Figure 2. Failure case analysis. NavGRPO incorrectly interprets the spatial reference "second bedroom to your left" in an instruction with multiple ambiguous directional cues, highlighting challenges in compositional spatial reasoning.

memory burden of multi-trajectory optimization. The memory footprint scales linearly with rollout number K, doubling to 28.652 GB at K=8, representing a manageable and predictable scaling behavior. Training time per thousand steps increases moderately with K, yet our approach typically converges within 10k steps, maintaining reasonable total training cost while achieving improved performance. With K=8 achieving 72.65% SPL compared to ScaleVLN's 69.97%, NavGRPO demonstrates a favorable performance-efficiency trade-off that justifies the moderate computational overhead.

5. Additional Visualizations

5.1. Qualitative Results

Figure 1 presents a representative navigation example that highlights the robustness differences between methods. Starting from the purple initial point, the agent must navigate through the archway toward the console table and then veer right to reach the open doorway. The red trajectory represents an initial perturbation that could mislead the agent. NavGRPO demonstrates stronger robustness by recovering from potential distractions and adhering to the instructional semantics, while ScaleVLN shows greater sensitivity to such perturbations. This qualitative observation aligns with our quantitative results, confirming that multi-trajectory optimization with balanced rewards enhances policy stability in complex navigation scenarios.

5.2. Failure Analysis

Figure 2 illustrates a challenging scenario where NavGRPO fails to reach the correct goal. The instruction requires the agent to navigate from the kitchen, veer right, and enter the second bedroom to the left. However, the spatial reference "second bedroom" is ambiguous given multiple visually similar bedroom entrances, and the conflicting directional cues "right" and "left" within the same instruction introduce additional complexity. The agent successfully exits the kitchen but misidentifies the target bedroom, revealing limitations in resolving compositional spatial relations across long-horizon instructions. This case underscores the challenge of grounding ambiguous linguistic spatial references in environments with repetitive structures, suggesting future directions in enhanced spatial reasoning and disambiguation mechanisms.

6. Limitations

Dependency on Base Model Quality. Our method fine-tunes pretrained vision-language models and inherits their limitations. When the base model exhibits systematic failures (e.g., misinterpreting certain spatial relations or object attributes), group-relative optimization may amplify rather than correct these biases, as all sampled trajectories suffer from the same foundational errors.

References

[1] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34:5834–5847, 2021. 2

[2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1

[3] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025. 1

[4] Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025. 1

[5] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. 1