Discrete-Continuous Factor Graph Inference for Vision-and-Language Navigation

Author Names Omitted for Anonymous Review

***Abstract*—To follow natural language navigation instructions in unfamiliar environments robots need to be able to ground the references in the instruction to observations of the environment. These references are typically ambiguous when considered in isolation: they rely on the context of the overall route to disambiguate instances of an object category present in the environment. Factor graphs are a robust state-of-the-art solution to this visual disambiguation problem without language grounding in traditional vision-based navigation. We introduce DisCo-VLN, a training-free method that parses a language instruction into a discrete-continuous factor graph and refines it online during vision-based navigation. Discrete factors maintain a belief over the correspondence between each instruction reference and candidate object detections, rather than committing to a single match. Continuous factors ground the route to regions and to the observed environment geometry in addition to objects. Because the instruction orders its references, it provides a prior on when each is likely to be observed, which we use to modulate each factor's uncertainty by the robot's progress along the route: the part it has reached defers to what it observes, while parts still ahead are held to the structure the instruction predicts. The maintained belief enables principled disambiguation between visually similar instances and recovery from initial misinterpretations.**

I. Introduction

How should a robot represent the ambiguity inherent in a natural language navigation instruction? Natural language often leaves distances and turn angles under-specified and references objects and rooms by category rather than coordinate without disambiguation of instances of the same category. Even for a human, these ambiguities can only be resolved using observations of the physical environment, gathered while moving through it according to the instruction. Vision-and-language navigation [4, 24] (VLN) studies this capability in robots: given an instruction, the robot must ground it to an unfamiliar environment to follow the described route.

Early methods learned sequence-to-sequence models trained in simulation [11, 12, 19, 2], but suffer a large sim-to-real gap [5]. The advent of large language models (LLMs) and visual foundation models trained on real-world data has motivated training-free VLN. Some methods use an LLM to directly output actions [37, 28], requiring a call at every step. Others construct explicit environment representations: value maps [27, 21] score regions or actions for consistency with the instruction but ignore the global trajectory structure the instruction implies, while graph-based methods [35, 32] use a single LLM call to parse the whole instruction into a structured representation, grounded to the environment and iteratively planned over. GC-VLN [35] represents instruction structure as graph constraints with tree-search backtracking over entity alignments, but treats them as hard constraints and so cannot recover from parsing error or detection failure. LIFGIF [32] encodes the structure as a continuous factor graph but is limited to object-centric instructions and commits to the best-matching observation for each referenced object; its two main failure cases, multiple object instances and incorrect detections, illustrate the downside of this early commitment.

Establishing correspondences between entities is the well-studied problem of *data association*, especially in the context of simultaneous localization and mapping (SLAM). Early approaches committed to a single best hypothesis [29], which is brittle: an early wrong commitment cannot be revised when later observations contradict it. More recent approaches maintain a probabilistic belief over plausible associations and resolve them as evidence accumulates [7, 30, 16]. VLN has exactly the conditions where this matters: partial observations, ambiguous language, and visually similar instances that may each match the same instruction reference.

We propose DisCo-VLN, a training-free VLN method parses an instruction with an LLM to form a discrete-continuous (hybrid) factor graph [17, 1] and uses it to follow the instruction while refining its grounding to the observed environment online. Unlike LIFGIF [32], we model data association probabilistically through discrete factors and add continuous factors that ground the graph beyond objects, to region containment and geometric structure. Our probabilistic treatment of the full instruction structure gives the robot a principled mechanism to reason about ambiguous grounding candidates and errors in the initial estimated geometry.

Contributions:

- We formulate route-based natural language instruction following as online probabilistic data association. DisCo-VLN parses an instruction into a hybrid discrete-continuous factor graph and maintains a belief over the correspondences between instruction references and observed entities, jointly with the route geometry, taking into account region containment and the observed environment geometry.
- We adapt discrete-continuous data association from the re-observation regime it is designed for to the anticipatory regime of instruction following, in which the references are instantiated from language before, and sometimes without ever, being observed. Because the instruction, and hence the graph, is ordered, we modulate the uncertainty of its factors by the robot's progress: the current part defers to observations of the environment, while later parts are held closer to the predicted route structure.

II. Related Work

A. Vision-and-language Navigation (VLN)

VLN was originally formulated for discrete environments [4], where the agent navigates between predefined

panoramic viewpoints, and later adapted to continuous environments (VLN-CE) [24] closer to deployment. Continuous-environment methods often use waypoint-prediction networks [20, 23] trained in simulation. Early methods used no explicit scene representation [4, 11, 19]; later work added topological maps [12, 3], grid maps [18], or both [2, 33], and some fine-tune large vision-language models on navigation data [36, 13]. Recent LLMs and vision-language models [31, 26, 6, 9] enable training-free VLN by providing zero-shot grounding of language to observations. Many such methods for discrete VLN rely on an LLM to select an edge to follow at each node based on visual observations [37, 28]. MapGPT [10] additionally provides the LLM with a topological map of explored nodes. For VLN-CE, VLMaps [21] uses an LLM to translate the instruction into a sequence of subgoals over a vision-language feature map, executed sequentially without reasoning over the instruction's global structure, while InstructNav [27] queries the LLM at each decision step to identify the current sub-instruction and plans toward it with a fused value map. Most closely related to our work are methods that extract the instruction's global structure with a single LLM call at the start of the episode and reason over it jointly as observations accumulate [32, 35]. LIFGIF [32] represents the instruction as a continuous factor graph with landmark variables for referenced objects and waypoint variables for the agent's position between sub-parts, committing to the best-matching observation per landmark based on vision-language feature similarity. GC-VLN [35] represents the instruction as graph constraints solved by a constraint solver, with tree-search backtracking over alternative entity alignments when no satisfying assignment is found. We differ in modeling data association probabilistically through discrete factors in a hybrid factor graph, and in extending the grounding to region containment and geometric structure.

B. Data association and semantic SLAM

Early approaches resolved ambiguity by selecting a single best hypothesis as observations arrive [29]. This is brittle in online settings: choices made under partial information cannot be revised when later observations contradict them. Probabilistic approaches instead maintain a belief over multiple plausible associations that is refined as evidence accumulates. Max-mixture methods [30, 16] represent each as a max-mixture of Gaussian factors, but the resulting multi-modal cost surface can trap the optimizer in a sub-optimal minimum. Expectation-maximization (EM) approaches [7] alternate between posterior association weights and continuous optimization. DC-SAM [17] introduced an alternating framework for discrete-continuous factor graphs, but like EM can converge to fixed points where neither estimate can be improved given the other. Variable-elimination-based hybrid inference [1] instead enables joint inference over discrete and continuous variables without alternation; we leverage this tooling. VLN imposes distinct considerations from semantic SLAM: predicted reference positions come from the route description rather than observations, giving weak initial estimates that worsen local-optima sensitivity; the instruction couples associations, since referenced entities of a shared class should map to distinct detections, requiring factors between association variables; and references are instantiated in advance of observation, so what the robot expects to see depends on its progress. The first two motivate joint inference [1]; the last motivates modulating factor uncertainty by route progress.

III. Method

Language-specified navigation instructions typically describe scene layouts — objects, regions, and their spatial relations — but not precisely enough to follow directly. The robot must refine the instruction by grounding it to observations as it moves. This is non-trivial: multiple instances of the same object category typically appear, instructions rely on the global route context to disambiguate, while imperfect detection adds missed and spurious detections. At the core of DisCo-VLN is a hybrid factor graph that grounds the instruction to the robot's observations. A large language model (LLM) extracts its structure: the waypoints the robot should pass through, the landmarks it uses to localize itself, the relations between them, and the region each lies in or is excluded from. This structure is refined online as observations accumulate. Each landmark's correspondence to detected objects is modeled probabilistically through discrete factors; geometric relations, region containment, and observed scene geometry enter as continuous factors, optimized jointly with the discrete ones.

A. 2D maps

DisCo-VLN builds a set of 2D top-down maps of its surroundings online from RGB-D observations, supplying the observations the factor graph grounds against. The **occupancy map** back-projects depth into a grid in which each cell is free, occupied, or as yet unobserved. From this we read the signed distance $s_o(x)$ to the nearest obstacle, positive in free space, and the extent of the observed free space. The **object map** is built by using an open-vocabulary detector (SAM 3 [9]) prompted with the labels of references; detections are fused into a persistent semantic map where recurring same-label detections at one location merge into a single instance. The centroids of instances carrying a reference's label form its candidate set $\{c_k\}$. The **region map** extracts regions from a dense language-aligned feature map, formed by ground-projecting features from Talk2DINO [6] and fusing them using the weighted average. Cells whose feature best matches a region label are grouped into connected instances, each yielding a signed-distance field s_r, positive inside.

B. Hybrid factor graph

Our hybrid factor graph [15, 17, 1] jointly represents the route geometry and the association of landmarks to observed objects. Its nodes are the waypoints $W = \{w_i \in SE(2)\}$, the locations the robot navigates to in order, and the landmarks $L = \{l_j \in \mathbb{R}^2\}$, the objects it uses to localize itself. A region containing a node is recorded as a label on that node; a region referred to with nothing inside it (e.g. "go past the kitchen") is given an auxiliary region node $q_k \in \mathbb{R}^2$, constrained to

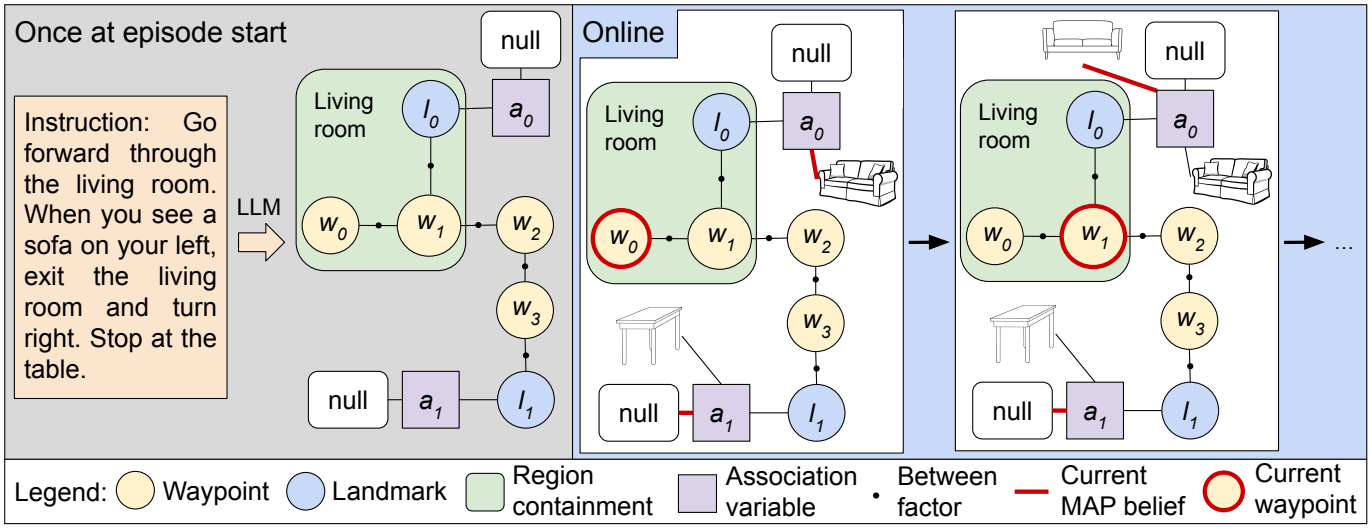

Fig. 1: **Hybrid factor graph creation and data association example.** At episode start, the LLM parses the instruction into waypoints, landmarks, regions and their relations, initializing the hybrid factor graph. Once navigation starts, w_0 is anchored to the robot's starting position and landmarks are probabilistically grounded to observed objects via the association variables; this belief is updated as observations arrive and the robot progresses, refining the estimated route geometry. (Only route-structure and data-association factors are shown.)

lie within it. The nodes and their relations are extracted from the instruction by an LLM. We prompt it with the instruction and in-context examples [8, 25, 34, 32] in a JSON format, and parse its response to construct the waypoints, landmarks, regions and their relations. The graph structure is fixed at the start of the episode. Data associations between landmarks and object detections are represented as discrete factors; all other relations, including region containment, as continuous factors. We optimize both jointly with a hybrid solver [14, 1] as the robot progresses along the waypoints in order, grounding and refining the graph as it moves Figure 1 shows an example of the initial extraction of the hybrid factor graph from the instruction, and data association at two steps.

1) Route structure factors: These factors impose the relations the instruction specifies between nodes. Because the instruction gives reliable direction but only loose distance, each relation is constrained tightly in the specified direction and loosely in range, rather than as a rigid relative pose. Let $w_i = (p_i, \theta_i)$ be a waypoint with position $p_i \in \mathbb{R}^2$ and heading θ_i. We model the relations between waypoints with a translating waypoint-to-waypoint relation which supplies a nominal offset $t \in \mathbb{R}^2$ in the frame of w_i and contributes a positional residual $R(\theta_i)^\top(p_{i+1} - p_i) - t$ with anisotropic covariance (loose along t, tight across it), applied asymmetrically, together with a heading residual $\theta_{i+1} - \mathrm{atan2}(p_{i+1} - p_i)$ that ties w_{i+1}'s orientation to its direction of travel from w_i, so changes in route geometry propagate forward as heading corrections.

A waypoint-to-landmark relation given as a direction supplies a nominal bearing β in the frame of w_i; with $d = R(\theta_i)^\top(l_j - p_i)$ the landmark in that frame, we impose a bearing residual $r^{\mathrm{bear}} = \mathrm{atan2}(d) - \beta$ with small variance and leave the range $\|d\|$ loose. A proximity relation ("near", "can see") instead constrains $\|d\|$ directly — tightly for "near", loosely for "can see" — with no bearing.

2) Data association: Grounding the instruction in the scene requires a consistent association between landmarks and the objects detected in the environment. A detection that appears where a landmark is predicted to lie is likely its referent; but when none lies near the prediction, or several fit it about equally well, the correspondence cannot be settled on present evidence — especially as the prediction itself is uncertain and still being refined. We therefore represent each object landmark l_j's correspondence as a discrete association variable a_j and maintain a belief over it rather than committing to a single match. Each a_j ranges over a null hypothesis and the current detected object instances matching l_j's label, $a_j \in \{0, 1, \ldots, K_j\}$, where each $k \geq 1$ indexes a detected object c_k and $a_j = 0$ is null. Detections enter this candidate set as they are observed. The association acts on the landmark's estimated position through a per-candidate factor that pins l_j to the selected detection and is inactive under null:

$$\psi_j(l_j, a_j) = \begin{cases} \|l_j - c_{a_j}\|^2_{\Sigma_c}, & a_j \geq 1, \\ 0, & a_j = 0. \end{cases} \tag{1}$$

The pin covariance Σ_c is tight, so selecting a candidate commits the landmark to that detection, while the unselected candidates contribute nothing. Discrete potentials weight each candidate by detector confidence and couple associations across landmarks: a non-equality potential forbids two same-label landmarks from claiming one detection, and a hysteresis potential mildly favors a landmark's current assignment to prevent oscillation. Because each a_j is a a graph variable, the associations are estimated jointly with the continuous waypoints and landmarks by the hybrid solver, and the marginal over a_j is a belief about the correspondence — consistency with the whole route disambiguates candidates through the continuous factors. Unlike schemes that resolve correspondence from current information, the belief here is revised by the inference itself as evidence accumulates, so an

early association contradicted by later observations is corrected rather than retained. Exact discrete inference is exponential in the number of variables; the solver [1] stays tractable through hypothesis pruning and dead-mode removal — committing a variable once its assignment is confident and dropping it from inference — so cost tracks the number of still-ambiguous associations rather than the total.

3) Continuous region containment factors: Instructions refer to regions as a destination, an object's location, or a place to pass; we ground each through a continuous containment constraint on a node — a waypoint when the region is a destination, a landmark when it contains an object, a region node q_k when it is only passed. From the region map's signed-distance field s_r (positive inside), a node required to lie inside a region contributes a one-sided hinge $\max\big(0,\, -s_r(x)\big)$ and one required to lie outside contributes $\max(0, s_r(x))$; the hinge is zero when satisfied, so it acts only to correct a violating node. Ideally a region would be associated through a discrete variable as an object is, but that would enlarge the active discrete set that must stay small for tractability, so we ground it through the containment constraint alone. We weight it weakly: with association unmodeled and region detection noisier than for objects, a strong factor would let a mislocated region warp the chain geometry. The cost is that a correctly detected region cannot strongly correct the graph.

4) Occupancy and reachability factors: Route-structure and data-association factors position the nodes but ignore where the robot can actually go: an optimized waypoint can come to rest inside an obstacle or be cut off from observed free space. Two factors tie waypoints to the navigable map. Using the occupancy map's signed distance $s_o(x)$, a waypoint is kept a clearance δ from obstacles via a hinge $\max(0,\, \delta - s_o(p_i))$. A second factor keeps it within reach of observed free space $\mathcal{F}$. Let $g(x)$ be the distance from x to $\mathcal{F}$ through free and unobserved space and a margin m; this factor applies $\rho\big(\max(0,\ g(p_i) - m)\big)$ under a Huber kernel ρ for numerical stability. Both apply only to the current and already-visited waypoints, not those ahead, which lie in still-unmapped space and should remain where the instruction predicts.

5) Temporal modulation of uncertainty: Data association is normally designed for re-observation: a landmark enters the graph when first observed and is recognized in later views. Instruction following violates this: a reference is instantiated from language before it is observed, and some are never observed, due to route errors or detector failures. The belief over each a_j is thus anticipatory, maintained before the reference is observed and left open for a late detection. This conflicts with dead-mode removal [1], which would confidently commit the association to null when no good candidate is present; we therefore exempt null from commitment. Because the instruction names references in encounter order, the robot's progress is a prior on when each should be observed, which we use to set each factor's uncertainty. Let w_c be the waypoint the robot is currently going to. Each waypoint-to-waypoint relation is scaled by $\rho_i \in (0, 1]$ (lower ρ_i means higher covariance): for $i > c$ the surrounding space is unobserved and the predicted structure is enforced ($\rho_i = 1$), while for $i \leq c$, if observations contradict the predicted position — the waypoint is unreachable, or its maximum a posteriori (MAP) association is null despite candidates present — ρ_i drops, freeing the bearing to fit what is observed. The penalty κ_i for leaving a reference unassociated runs oppositely (small ahead, large behind), so a candidate consistent with the current belief is associated at any i, while one requiring a large adjustment is deferred until $i \leq c$.

TABLE I: **Ablation study for probabilistic data association and temporal uncertainty modulation**

Method	SR	SPL	nDTW
w/o probabilistic data association	**17.00**	6.56	17.52
w/o temporal uncertainty modulation	**17.00**	9.87	30.10
DisCo-VLN	16.00	**10.89**	**31.16**

IV. Preliminary Results

We conduct an ablation study on 100 episodes of the val-unseen split of R2R-CE [24], sampled across scenes, reporting Success Rate (SR), Success weighted by inverse Path Length (SPL), and normalized Dynamic-Time Warping (nDTW) [4, 24, 22]: SR measures reaching the route end, SPL efficiency, and nDTW how closely the ground-truth path is followed.

We ablate two components. *w/o probabilistic data association* replaces the discrete association variables with a greedy assignment following LIFGIF [32], associating each landmark with the matching-class detection whose location in the vision-language feature map has the highest cosine similarity to its label, updated as detections arrive, with all continuous factors unchanged. *w/o temporal uncertainty modulation* fixes factor uncertainties regardless of progress while retaining the null-exemption from commitment and the gating of occupancy and reachability factors to current and past waypoints, isolating the modulation itself. As shown in Table I, our full method outperforms both ablations on SPL and nDTW despite a marginally lower SR. The gap is largest for w/o probabilistic data association — roughly half the SPL and nDTW of the full method — indicating that greedy commitment causes substantial deviations from the described route, the failure mode probabilistic association is designed to address. Temporal modulation contributes a smaller but consistent improvement on both path-quality metrics.

V. Conclusion

We present DisCo-VLN, a training-free vision-and-language navigation method that grounds natural language instructions to observations through a hybrid discrete-continuous factor graph. Modeling data association as discrete variables lets the robot maintain a belief over each landmark's correspondence and revise it as evidence accumulates, rather than committing to a single match. Continuous factors extend the grounding beyond objects to regions and the observed environment geometry. We adapt this machinery to the anticipatory regime of instruction following — where references are instantiated from language before they are observed, and sometimes never observed — by modulating each factor's uncertainty according to the robot's progress along the route. Preliminary ablations on R2R-CE indicate that both probabilistic data association and temporal uncertainty modulation improve how closely the robot follows the route described by the instruction.

References

[1] Varun Agrawal and Frank Dellaert. Variable elimination in hybrid factor graphs for discrete-continuous inference & estimation. *arXiv preprint arXiv:2601.00545*, 2026.

[2] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[3] Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[4] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton 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*, pages 3674–3683, 2018.

[5] Peter Anderson, Ayush Shrivastava, Joanne Truong, Arjun Majumdar, Devi Parikh, Dhruv Batra, and Stefan Lee. Sim-to-real transfer for vision-and-language navigation. In *Conference on Robot Learning*, pages 671–681. PMLR, 2021.

[6] Luca Barsellotti, Lorenzo Bianchi, Nicola Messina, Fabio Carrara, Marcella Cornia, Lorenzo Baraldi, Fabrizio Falchi, and Rita Cucchiara. Talking to dino: Bridging self-supervised vision backbones with language for open-vocabulary segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22025–22035, 2025.

[7] Sean L Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J Pappas. Probabilistic data association for semantic slam. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1722–1729. IEEE, 2017.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[10] Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee K. Wong. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

[11] 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.

[12] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan 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*, pages 16537–16547, 2022.

[13] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. In *RSS*, 2025.

[14] Frank Dellaert and GTSAM Contributors. borglab/gtsam, May 2022. URL https://github.com/borglab/gtsam).

[15] Frank Dellaert, Michael Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.

[16] Kevin J Doherty, David P Baxter, Edward Schneeweiss, and John J Leonard. Probabilistic data association via mixture models for robust semantic slam. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1098–1104. IEEE, 2020.

[17] Kevin J Doherty, Ziqi Lu, Kurran Singh, and John J Leonard. Discrete-continuous smoothing and mapping. *IEEE Robotics and Automation Letters*, 7(4):12395–12402, 2022.

[18] Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15460–15470, 2022.

[19] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1643–1653, 2021.

[20] Yicong Hong, Zun Wang, Qi Wu, and Stephen 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*, pages 15439–15449, 2022.

[21] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[22] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019.

[23] Jacob Krantz and Stefan Lee. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In *European conference on computer vision*, pages 588–603. Springer, 2022.

[24] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In

Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16, pages 104–120. Springer, 2020.

[25] Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pages 100–114, 2022.

[26] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[27] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.

[28] Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. Discuss before moving: Visual language navigation via multi-expert discussions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17380–17387. IEEE, 2024.

[29] José Neira and Juan D Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on robotics and automation*, 17(6): 890–897, 2001.

[30] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *The International Journal of Robotics Research*, 32(7):826–840, 2013.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[32] Sonia Raychaudhuri, Duy Ta, Katrina Ashton, Angel X Chang, Jiuguang Wang, and Bernadette Bucher. Zero-shot object-centric instruction following: Integrating foundation models with traditional navigation. *arXiv preprint arXiv:2411.07848*, 2024.

[33] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 15625–15636, 2023.

[34] Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1423–1436, 2023.

[35] Hang Yin, Haoyu Wei, Xiuwei Xu, Wenxuan Guo, Jie Zhou, and Jiwen Lu. Gc-vln: Instruction as graph constraints for training-free vision-and-language navigation. *arXiv preprint arXiv:2509.10454*, 2025.

[36] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *Robotics: Science and Systems*, 2024.

[37] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv preprint arXiv:2305.16986*, 2023.