Dynamic Objects Relocalization in Changing Environments with Flow Matching

Francesco Argenziano[1,*], Miguel Saavedra-Ruiz[2,3,*], Sacha Morin[2,3], Daniele Nardi[1], and Liam Paull[2,3]

***Abstract*— Task and motion planning are long-standing challenges in robotics, especially when robots have to deal with dynamic environments exhibiting long-term dynamics, such as households or warehouses. In these environments, long-term dynamics mostly stem from human activities, since previously detected objects can be moved or removed from the scene. This adds the necessity to find such objects again before completing the designed task, increasing the risk of failure due to missed relocalizations. However, in these settings, the nature of such human-object interactions is often overlooked, despite being governed by common habits and repetitive patterns. Our conjecture is that these cues can be exploited to recover the most likely objects' positions in the scene, helping to address the problem of unknown relocalization in changing environments. To this end we propose FlowMaps, a model based on Flow Matching that is able to infer multimodal object locations over space and time. Our results present statistical evidence to support our hypotheses, opening the way to more complex applications of our approach. The code is publically available at `https://github.com/Fra-Tsuna/flowmaps`.**

I. INTRODUCTION

Executing long-horizon tasks in real-world environments remains a major challenge for mobile robots. Even a simple instruction like "fetch me the mug from the kitchen" requires robust mapping, planning, and decision-making pipelines. Examples of such tasks in robotics include **object navigation** and **object retrieval**: object navigation involves finding and navigating to a target object, whereas object retrieval also requires approaching, manipulating, and lastly delivering the object to a specified destination.

Recent advances in Large Language Models (LLMs) and Vision-Language Models (VLMs) have improved the performance of embodied agents in such tasks by enabling richer semantic queries. Specifically, LLMs support commonsense reasoning [1], while VLMs provide compact, high-dimensional scene features (e.g., CLIP [2]) that allow agents to better perceive and reason about objects in the environment [3].

However, many approaches assume *static* environments, where objects move only if the robot acts on them. This assumption is unrealistic: real-world scenes are *dynamic*, as humans constantly move objects and reorganize rooms. Consequently, robots cannot rely on objects staying in their last observed locations, which poses a major risk for both navigation and retrieval tasks.

Humans, though, exhibit repetitive patterns [4]: for example, a bottle taken from the kitchen table is likely placed on a desk and later returned to the kitchen sink. These *semantically consistent patterns* [5], [6] can be exploited to recover from missed retrievals by relocating objects at likely human-induced placements. Since these locations are inherently multimodal, a distributional approach is required.

We therefore propose **FlowMaps**, a Flow Matching [7] model that recovers multimodal object distributions over space and time. Using a transformer-based map encoder and observations of object placements over time, FlowMaps infers plausible locations for query object classes, improving retrieval success.

We summarize our contributions as follows:

- **FlowMaps**: a Flow Matching–based model for multimodal object relocalization in long-term dynamic scenes.
- **FlowSim**: a procedurally generated dataset that simulates multimodal object placements over time.
- A qualitative evaluation empirically demonstrating the benefits of our approach, and a quantitative comparison against an MLP baseline.

This paper is organized as follows: Section II reviews related work, Section III details FlowMaps and FlowSim, Section IV presents the experimental results, and Section V concludes.

II. RELATED WORK

Object navigation and object retrieval. The goal of object navigation is to reach a target object specified by category or natural language, typically through semantic mapping and goal-conditioned exploration [8], [9]. Object retrieval extends this to finding, grasping, and delivering the object, with recent benchmarks enabling open-vocabulary mobile manipulation in realistic household environments [10], [11]. Despite their effectiveness, many pipelines still assume static scenes or rely on short-horizon replanning when target objects move, which is a recurring source of failure in environments undergoing human activities.

Flow Matching in robotics. Generative models are increasingly prominent in robotics, aided by large, cross-embodiment datasets such as Open-X Embodiment [12]. Building on this, recent work has applied diffusion [13], [14] and Flow Matching (FM) [15], [16] to learn action policies,

*Authors contributed equally.

[1] Department of Computer, Automation and Management Engineering, Sapienza University of Rome, 00181 RM Rome, Italy. {argenziano, nardi}@diag.uniroma1.it

[2] Department of Computer Science and Operations Research, Université de Montréal, Montréal, QC, Canada.

[3] Mila - Quebec AI Institute, Montréal, QC, Canada. {miguel-angel.saavedra-ruiz, sacha.morin, paulll}@mila.quebec

while others have developed end-to-end vision-language-action systems [17], [18]. While these approaches have shown impressive capabilities, they largely focus on policy learning. In contrast, we leverage FM to recover spatio-temporal, multimodal posterior distributions over potential object placements rather than generate action tokens. To our knowledge, this is the first application of FM to posterior inference in this setting.

III. METHODOLOGY

This section is organized as follows. Section III-A provides an overview of Flow Matching, followed by Section III-B, which outlines the problem formulation for dynamic objects relocalization. Section III-C details the data collection procedure, and Section III-D gives a complete description of our system's architecture, with implementation details provided in Section III-E.

A. Preliminaries

An *Ordinary Differential Equation* (ODE) is defined by a time-dependent *vector field* $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ that specifies a velocity $u_t(x) \in \mathbb{R}^d$ for each position $x \in \mathbb{R}^d$ and time t. A solution to an ODE for a given initial condition x_0 is called a *trajectory* $X : [0,1] \to \mathbb{R}^d$, describing the path of that single point over time with $X_t = x_t$. More generally, a *flow* $\psi : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is the solution function of the ODE for all initial points x_0, satisfying

$$\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x)), \qquad \psi_0(x_0) = x_0.$$

The goal of *Flow Matching* (FM) is to learn a vector field u_t^θ, parameterized by θ, whose flow ψ_t^θ transports a simple base distribution $p_0 \triangleq p_{\text{init}}$ to a target data distribution $p_1 \triangleq p_{\text{data}}$. In practice, FM constructs a target velocity field $u_t^{\text{target}}(x)$ from sample pairs and trains u_t^θ to match it, so that integrating $u_t^\theta(X_t)$ moves p_{init} along some predefined paths to generate samples from p_{data}. These *probability paths* $(p_t)_{0\le t\le 1}$ specify a gradual interpolation between noise p_{init} and data p_{data}. A standard probability path for example is the *linear* path: given $X_0 \sim p_0$, $X_1 \sim p_1$ and $t \sim \text{Unif}[0,1]$, we can sample $X_t = tX_1 + (1-t)X_0 \sim p_t$. The FM objective is

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t\sim\text{Unif},x\sim p_t}[||u_t^\theta(x) - u_t^{target}(x)||^2],$$

but since the *marginal* vector field $u_t^{\text{target}}(x)$ is not tractable in practice, one minimizes the *Conditional Flow Matching* (CFM) loss

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t\sim\text{Unif},z\sim p_{\text{data}},x\sim p_t(\cdot|z)}[||u_t^\theta(x) - u_t^{target}(x|z)||^2],$$

because it holds that $\nabla\mathcal{L}_{\text{FM}} = \nabla\mathcal{L}_{\text{CFM}}$, where now z is sampled from p_{data}, x is sampled from the *conditional* probability path $p_t(\cdot|z)$, and $u_t^{target}(x|z)$ is the *conditional* vector field. For a more complete explanation and formalization we remind [19].

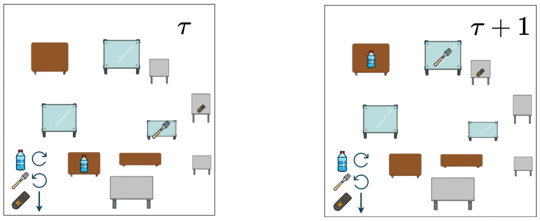


Fig. 1: A FlowSim environment across two consecutive timestamps.

B. Problem formulation

At time $\tau \ge 0$ we represent the map as $\mathcal{M}_\tau = (\mathcal{O}_\tau, O_{\text{BG}})$, where $\mathcal{O}_\tau = \{O_\tau^i\}_{i=1}^N$ is the set of objects, and O_{BG} denotes the static background that is assumed to remain unchanged over time. Each object is represented as

$$O_\tau^{(i)} = \left(X_\tau^i, f_\tau^i, l^i\right),$$

where $X_\tau^i \in \mathbb{R}^2$ denotes its 2D position in space, f_τ^i represents object features (e.g., appearance or shape descriptors), and l^i is a text descriptor (e.g., `red coffee mug`).

Given a prediction horizon $\Delta\tau \ge 0$ and a text-object query O^q, our goal is to infer the likely (multimodal) future locations at time $\tau_q = \tau + \Delta\tau$. Formally, we aim to compute the posterior

$$p\left(X_{\tau_q}^q \,\middle|\, \mathcal{M}_\tau, O^q, \Delta\tau\right). \tag{1}$$

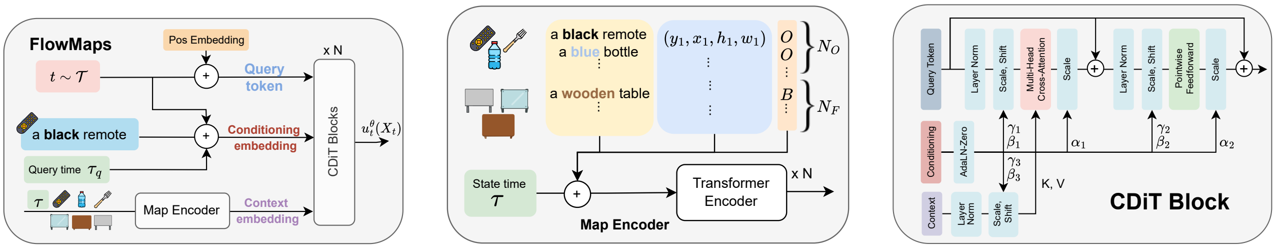
As the posterior in (1) can be highly multimodal and intractable, we employ FM to approximate it and generate samples from it [7]. In particular, given a random sample $X_{\text{init}} \sim p_0$ obtained from a simple distribution $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{I})$, our goal is to make $X_{\text{final}} \sim p_1$ with $p_1 = p\left(X_{\tau_q}^q \,\middle|\, \mathcal{M}_\tau, O^q, \Delta\tau\right)$.

C. FlowSim

FM (and generative models in general) are *data-hungry*: they require a significant amount of data to be able to approximate well a target distribution. Despite recent community efforts toward large-scale dynamic datasets [20], to the best of our knowledge there is no resource that fully matches our needs: indoor scenes with objects that move along semantically consistent patterns over space and time.

To bridge this gap, we procedurally generate data with our proposed simulator, FlowSim. In FlowSim, objects move across static pieces of furniture following predefined, category-specific patterns. Figure 1 illustrates the motion between two consecutive timestamps, mimicking distinct human interaction patterns: a *bottle* moving counterclockwise, a *fork* clockwise, and a *remote* top-to-bottom[1]. Furthermore, transitions are stochastic: at each timestamp an object advances to the next location along its trajectory with probability 0.7, remains at where it is with probability 0.2, and skips one step (advancing by two locations) with probability 0.1. This controlled stochasticity induces multimodal spatio-temporal distributions.

[1] We prototype the simulator with three object categories, but it is easily extensible to many more.



(a) FlowMaps (b) Map Encoder (c) CDiT Block

Fig. 2: FlowMaps (a) architecture and its modules, the (b) Map Encoder and the (c) CDiT Block.

FlowSim logs the bounding boxes and descriptors per timestamp for all objects in the scene (both static and dynamic) providing exactly the supervision required for FlowMaps.

D. FlowMaps

In Fig. 2 the FlowMaps architecture is shown (2a), alongside its two main modules: the **Map Encoder** (2b) and the **Conditional Diffusion Transformer** (CDiT) block (2c).

The Map Encoder embeds the map at state time τ by encoding object and furniture descriptors (i.e., *colors* or *labels*) and bounding boxes into tokens $\mathbf{e} \in \mathbb{R}^{(N_O+N_F)\times D}$, where $N_O + N_F$ is the number of tokens. Similar to [21], we also encode an object-type flag indicating whether the i-th token is an object or a piece of furniture (denoted as O and B in Fig. 2b). Tokens are padded to a maximum length S, augmented with a state time embedding of τ, and passed through various transformer encoders [22] to yield a scene-context map embedding. Although Fig. 1 shows a rendered environment, FlowMaps is *not* a visual model: to avoid trivializing the task, it consumes FlowSim's structured annotations (bounding boxes and descriptors), not images.

The Diffusion Transformer (DiT) block [23] is a scalable backbone for training latent generative models. We use a modified CDiT of [24] with one change: we drop its Multi-Head Self-Attention, since we diffuse only the query object rather than the full scene. Each CDiT block takes (i) a **query token**, (ii) a **conditioning embedding** that drives *AdaLN-Zero* gates (α, β, γ), and (iii) a **context embedding** used to cross-attend the **query**. To form a **query token**, we sample $t \sim U[0, 1]$, interpolate X_t along a probability path p_t between $X_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{4\times4})$ and the ground-truth box $X_1 = (y, x, h, w)$, then add a sinusoidal positional embedding. The **conditioning embedding** encodes and sums the query time τ_q, the query object descriptor, and FM time t; the **context embedding** is the Map Encoder output.

CDiT learns the vector field $u_t^\theta(X_t)$, and given ground-truth $u_t(X_t)$, we minimize $\mathcal{L} = ||u_t(X_t) - u_t^\theta(X_t)||^2$, backpropagating through the Map Encoder and the CDiT blocks.

During inference, we draw $X_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{4\times4})$ and numerically integrate the ODE to obtain the predicted position $\hat{X}_{\tau_q}^q$ at future time τ_q, conditioned on the current map $\mathcal{M}_\tau$ and the object descriptor.

E. Implementation details

We set embedding dimension to $D = 256$ and $N = 8$ blocks in both the CDiT and the Transformer encoder, maximum timestamp horizon $\tau_{max} = 20$, and chose *color* information as descriptors. We tested training with and without dropout and observed better performance with dropout enabled at $p = 0.1$. For the time-sampling distribution over t, we evaluated the standard *uniform*, *beta* [17], and *logit-normal* [25] distributions; only the *uniform* choice degraded performance. As the probability path, we adopt the *linear* path [19]. For ODE integration we use the *midpoint* method [19]. We train with an *AdamW* optimizer with $\beta_1 = 0.95$ and $\beta_2 = 0.999$, *Exponential Moving Average* (EMA) smoothing for weights with rate of 0.9999, and a *cosine* learning-rate schedule with starting LR $\eta = 1e-4$ for 30k iterations.

IV. RESULTS

We compare FlowMaps with an MLP-based baseline that also employs the same Map Encoder and regresses over a Gaussian distribution centered on the ground truth bounding box. Figure 3 reports the KL divergence $KL(q||p_\theta)$ computed between the ground truth distribution q and the predicted one p_θ. The results show that our method has better mode distribution coverage than the MLP baseline, highlighting the importance of a multimodal distributional approach for this task. Consistent with this trend, Fig. 4 shows FlowMaps inference results for multiple objects at future timestamps. For each query, we generate 25 samples to obtain multiple plausible bounding box positions. As shown in the image, the model successfully distinguishes object-specific behaviors and recovers a multimodal distribution. Together, these qualitative and quantitative results indicate that FlowMaps effectively captures object-conditioned dynamics and the multimodal nature of future object positions.

V. CONCLUSIONS

In this paper we introduced FlowMaps, a Flow Matching based model that infers multimodal posterior distributions over future object locations in household environments with long-term dynamics, together with FlowSim, a procedurally generated dataset that captures semantically consistent, human-induced object motions. Our results indicate that FlowMaps learns object-conditioned routines and provides robust priors that can help recover from missed relocalizations during object navigation and retrieval tasks.

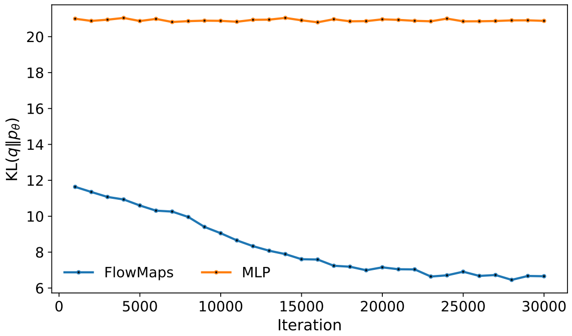
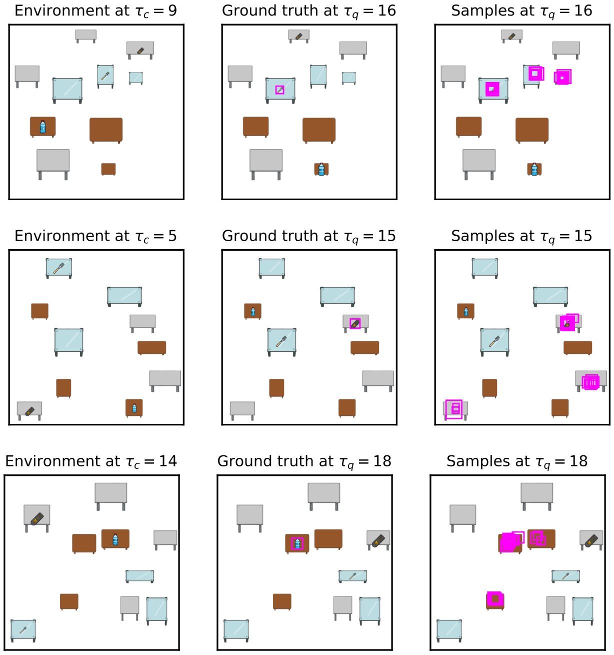


Fig. 3: KL divergence comparison between FlowMaps (blue) and the MLP baseline (orange).


Environment at $\tau_c = 9$

Ground truth at $\tau_q = 16$

Samples at $\tau_q = 16$

Environment at $\tau_c = 5$

Ground truth at $\tau_q = 15$

Samples at $\tau_q = 15$

Environment at $\tau_c = 14$

Ground truth at $\tau_q = 18$

Samples at $\tau_q = 18$

Fig. 4: Results when predicting locations for a fork (top), remote (mid) and bottle (bottom).

Looking ahead, we will (i) scale from simulation to real homes with longer horizons and richer object vocabularies and full text object queries, (ii) integrate FlowMaps as an uncertainty-aware placement prior within exploration and planning stacks, and (iii) extend the formulation to 3D Scene Graphs [3] and online updating so that the posterior can be refined as new evidence arrives. We believe this is a step toward robot behaviors that exploit human regularities to operate reliably in changing environments.

ACKNOWLEDGMENT

This work has been carried out while Francesco Argenziano was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome. This work was partially funded by an FRQNT B2X Scholarship with DOI 10.69777/335998 [MS]. This research was conducted while Francesco Argenziano was enrolled as a visiting researcher at Mila - Quebec AI Institute. We acknowledge partial financial support from PNRR MUR project PE0000013-FAIR.

REFERENCES

[1] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, “Language models as zero-shot planners: Extracting actionable knowledge for embodied agents,” in *International conference on machine learning*, 2022.

[2] A. Radford *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*, 2021.

[3] Q. Gu *et al.*, “Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning,” in *Proc. IEEE Int. Conf. Robot. and Automation*, 2024.

[4] N. F. Troje, “Retrieving information from human movement patterns,” *Understanding events: From perception to action*, vol. 4, 2008.

[5] L. Schmid, J. Delmerico, J. L. Schönberger, J. Nieto, M. Pollefeys, R. Siegwart, and C. Cadena, “Panoptic multi-tsdfs: a flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency,” in *Proc. IEEE Int. Conf. Robot. and Automation*, 2022.

[6] V. Yugay, T. Kersten, L. Carlone, T. Gevers, M. R. Oswald, and L. Schmid, “Gaussian mapping for evolving scenes,” *arXiv preprint arXiv:2506.06909*, 2025.

[7] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, “Flow matching for generative modeling,” in *The Eleventh International Conference on Learning Representations*, 2023.

[8] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, “Object goal navigation using goal-oriented semantic exploration,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[9] J. Sun, J. Wu, Z. Ji, and Y.-K. Lai, “A survey of object goal navigation,” *IEEE Transactions on Automation Science and Engineering*, vol. 22, 2025.

[10] S. Yenamandra *et al.*, “Homerobot: Open vocabulary mobile manipulation,” 2023. [Online]. Available: https://github.com/facebookresearch/home-robot

[11] B. Han, M. Parakh, D. Geng, J. A. Defay, G. Luyang, and J. Deng, “Fetchbench: A simulation benchmark for robot fetching,” in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 270, 2025.

[12] A. O’Neill *et al.*, “Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0,” in *Proc. IEEE Int. Conf. Robot. and Automation*, 2024.

[13] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics Research*, 2024.

[14] Z. Hou *et al.*, “Diffusion transformer policy,” *arXiv preprint arXiv:2410.15959*, 2024.

[15] E. Chisari, N. Heppert, M. Argus, T. Welschehold, T. Brox, and A. Valada, “Learning robotic manipulation policies from point clouds with conditional flow matching,” in *Conference on Robot Learning*, 2025.

[16] F. Zhang and M. Gienger, “Affordance-based robot manipulation with flow matching,” *arXiv preprint arXiv:2409.01083*, 2024.

[17] K. Black *et al.*, “pi0: A vision-language-action flow model for general robot control,” *arXiv preprint arXiv:2410.24164*, 2024.

[18] M. J. Kim *et al.*, “Openvla: An open-source vision-language-action model,” in *Conference on Robot Learning*, 2025.

[19] Y. Lipman *et al.*, “Flow matching guide and code,” 2024. [Online]. Available: https://arxiv.org/abs/2412.06264

[20] T. Sun, Y. Hao, S. Huang, S. Savarese, K. Schindler, M. Pollefeys, and I. Armeni, “Nothing stands still: A spatiotemporal benchmark on 3d point cloud registration under large geometric and temporal change,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 220, 2025.

[21] J. Wald, H. Dhamo, N. Navab, and F. Tombari, “Learning 3d semantic scene graphs from 3d indoor reconstructions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[23] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

[24] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun, “Navigation world models,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

[25] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel *et al.*, “Scaling rectified flow transformers for high-resolution image synthesis,” in *Forty-first international conference on machine learning*, 2024.