Zero-Shot Adaptation of Behavioral Foundation Models to Unseen Dynamics

Anonymous Authors[1]

Abstract

Behavioral Foundation Models (BFMs) like successor measure-based methods excel in zero-shot policy generation but struggle with dynamic changes, limiting real-world applicability (*e.g.*, , robotics). We show that Forward-Backward (FB) representations fail to distinguish between different dynamics, causing latent interference. To fix this, we propose an FB model with a transformer-based belief estimator, enabling better zero-shot adaptation. Additionally, clustering the policy space by dynamics improves performance. Our method adapts to trained dynamics and generalizes to unseen ones, achieving up to 2x higher zero-shot returns in discrete and continuous tasks compared to baselines.

1. Introduction

A key goal in reinforcement learning (RL) is fast adaptation to new tasks or environment changes, ideally in a zero-shot manner—without test-time interaction (Touati et al., 2022). Behavioral Foundation Models (BFMs) (Sikchi et al., 2024; Tirinzoni et al.) advance this by learning diverse policies from offline data, independent of rewards. At inference, they extract near-optimal task-specific policies. Notably, Forward-Backward (FB) (Touati & Ollivier, 2021) representations, a BFM variant, excel at imitating behaviors from unlabeled data.

At the same time, FB possesses a fundamental drawback that limits its adaptation ability. In our paper, we show that FB is unable to generalize across different dynamics, such as changes in a transition function (*e.g.*, new obstacles) or an environment with some latent factor variation (*e.g.*, wind direction). This limitation stems from the way the *successor measure* (Dayan, 1993) is estimated: FB averages the future-occupancy state distribution over all observed dynamics, which inevitably causes interference in policy representations. This fact alone may severely constrain the applicability of FB in the real-world scenarios. For example, one of the largest robotics dataset, Open X-Embodiment (Collaboration, 2023), consists of 22 different robot embodiments, and training FB on each of them simultaneously is infeasible. In Section 3.1, we discuss this limitation and support our claims theoretically.

To remedy this, we introduce Belief-FB (BFB), a conditioning method for FB through a *belief* estimation, a popular technique of uncertainty quantification in Meta-RL (Dorfman et al., 2021; Zintgraf et al., 2020). To implement this, we use a transformer encoder f_{dyn} that, given a trajectory from data, outputs a dynamics-specific vector h, which is then passed as a condition to the future encoding function $F(\cdot, \cdot, h, \cdot)$. We pre-train f_{dyn} in a self-supervised fashion, thus posing no additional requirements on the data structure or the trajectory re-labeling. We discuss the implementation of Belief-FB in Section 3.2.

Remarkably, Belief-FB enables the generalization capabilities of FB not only through the dynamics seen **in the training dataset**, but also on the **unseen test dynamics** never present in the offline data. We also find that in order to align *belief* estimation better with FB, one also needs to change sampling procedure of encoding direction, which we term Rotation-FB (RFB). We present the theoretical support and the implementation details of Rotation-FB in Section 3.3. Empirically, both BFB and RFB outperform baselines for seen and unseen dynamics, as gathered in Figure 1 and discussed in Section D.1.

We believe that our work sufficiently broadens the possible applicability of BFMs, yet keeping the zero-shot setting unchanged. Our contributions are as follows:

- **We demonstrate the limitation of Forward-Backward (FB) representations** (Touati & Ollivier, 2021), which lies in its inability to generalize *per se* across different dynamics (*e.g.*, new layouts). We refer to Section 3.1 for more discussion.

- **We propose Belief–FB (BFB)**, which employs a transformer encoder to infer a belief over the agent's current dynamics (Dorfman et al., 2021; Zintgraf et al., 2020). Analyzing BFB's policy space reveals that additional disentanglement is beneficial, motivating our **Rotation–FB (RFB)**. Both approaches outperform significantly other methods (Figure 1)

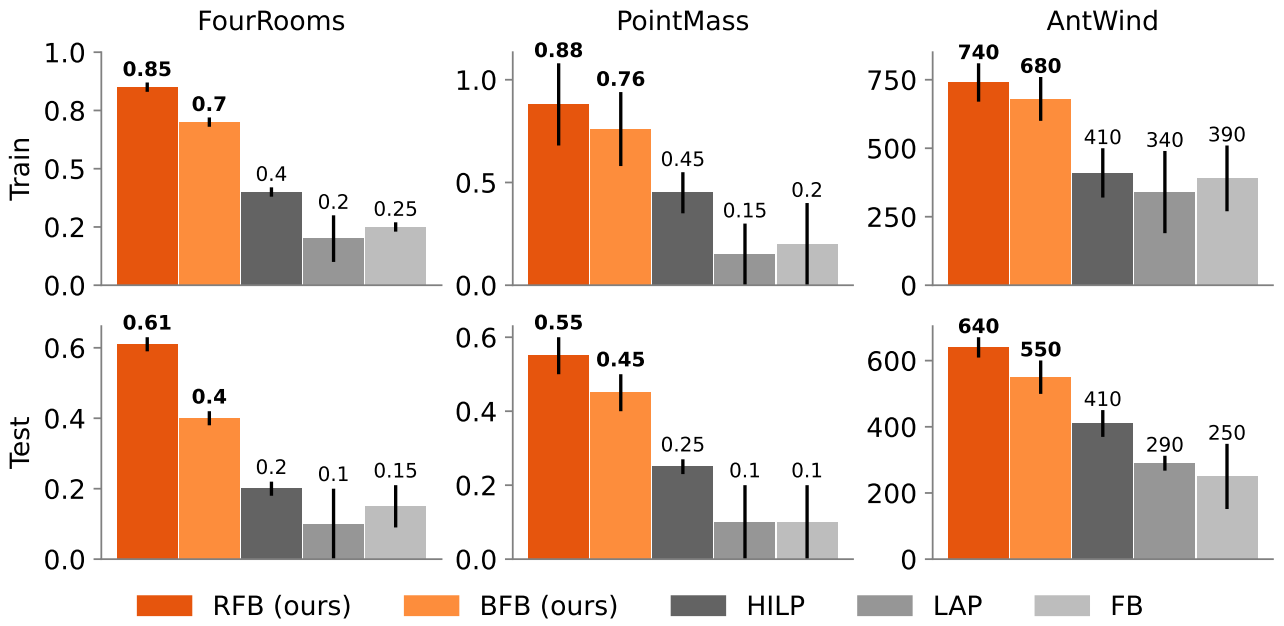


Figure 1. **Summary of results**. Aggregate mean performance over *seen* (train) and *unseen* (test) dynamics for zero-shot RL. The error bars indicate standard deviation over three seeds. Notably, both BFB and RFB adapt not only to the dynamics seen during training but are also able to generalize to unseen dynamics. There are 30 (20) training (test) dynamics for FourRooms and PointMass and 16 (4) for AntWind environments.

2. Behavioral Foundation Models

For a reward-free MDP, a Behavioral Foundation Model (BFM) (Frans et al., 2024; Pirotta et al., 2023; Sikchi et al., 2025; Tirinzoni et al.) is a RL agent trained in an unsupervised manner on a task-agnostic dataset of transitions. The objective of a BFM is to approximate an optimal policy for a broad class of reward functions that are specified only at inference.

Forward-Backward Representation (FB) (Touati & Ollivier, 2021) approximates a successor measure for near-optimal policies across diverse tasks. The successor measure $M^\pi(s_0, a_0, X)$ for subset $X \subset \mathcal{S}$ is defined as cumulative discounted time spend at X starting at (s_0, a_0) and following π thereafter. More formally, for tabular example:

$$M^\pi(s_0, a_0, X) = \sum_{t \geq 0} \gamma^t \mathrm{P}(s_t \in X | s_0, a_0, \pi), \quad (1)$$

with the corresponding Q-function for a specific task r:

$$Q_r^\pi(s_0, a_0) = \sum_{s^+ \in X} r(s^+) M^\pi(s_0, a_0, s^+). \quad (2)$$

In continuous case, the FB representation aims to approximate successor measure through finite-rank approximation under diverse policies through *forward* $F : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to \mathbb{R}^d$ and *backward* $B : \mathcal{S} \to \mathbb{R}^d$ functions. Given a set of policies π_z parametrized by task variable drawn uniformly from sphere $z_{\text{FB}} \in \text{Unif}(\mathcal{Z} = \mathbb{S}^{\lceil -\infty})$. Given ρ as a probability distribution over states within the offline dataset, the objective for FB is written as $M^{\pi_z}(s_0, a_0, X) \approx \int_{s^+ \in X} F(s_0, a_0, z)^T B(s^+) \rho(ds)$. Then the policy can be obtained greedily as:

$$\pi_z(s) \approx \arg\max_a F(s, a, z)^T z. \quad (3)$$

For continuous case, the greedy policy is parametrized as Gaussian. During test time the task policy parametrization is approximated as $z_{test} \approx \mathbb{E}_{(s,a) \in \mathcal{D}_{test}} \{r_{test}(s, a) B(s)\}$. If the inferred task vector z_{test} lies within the task sampling distribution (in a linear span) $\mathcal{Z}$ used during training, then the optimal policy for task r_{test} is obtained from Equation 2 as $\pi_z(s) \approx \arg\max_a Q^{\pi_z}_{r_{test}}(s, a)$. For more details on training and inference procedures of FB, we refer reader to Appendix A.3. More detailed discussion on the other related works is included in the Appendix A.

3. Method

Problem Statement. Given the set of contexts $\mathcal{C}_{train} = \{c_{train} \in \mathcal{C}\}$, the goal is learn an agent so that it is able to generalize to unseen ones dynamics changes during test time, *i.e.*, *zero-shot*[1]. We collect diverse dataset, consisting

[1]We use the term "zero-shot RL" following (Touati & Ollivier, 2021).

of mix of highly exploratory or expert-like unknown policies from varying environment layouts, differing either in dynamics (*e.g.*, wind, friction, etc.) or environment specifications (*e.g.*, positions of obstacles and doors). At test time, the agent is provided with small (up to episode termination steps) reward-free transitions from test context and should adapt its policy. In an ideal scenario, the agent maximizes the expected discounted return across both train and test contexts. We refer to Appendix A for details.

3.1. Investigating latent directions space under multiple dynamics

We begin by addressing the following question: Why does FB representations fail to generalize effectively (both for train and test) to different situations under dynamics variations, *i.e.*, if learned on data sampled from diverse CMDPs? A closer look at the geometry of the latent directions $z_{\text{FB}} \in \mathcal{Z}$, each indexing a policy π_z uncovers why Forward-Backwards (FB) learning struggles in custom partially-observable "Randomized Doors" grid world (Appendix B.1). In this setting door and wall positions change every episode; random rollouts give near-uniform coverage of (x, y) states, so the same state s can require different optimal actions across layouts. During training we sample $z_{\text{FB}} \sim \text{Uniform}(\mathbb{S}^{d-1})$ and evaluate $F(s, \cdot, z_{\text{FB}})$. Because z_{FB} is not forced to separate layout-specific futures, latent directions for conflicting behaviours overlap, causing interference. Figure 10 illustrates how, when FB is trained on a single layout family, a dominant direction emerges and recovers the optimal policy π^*. Mixing transitions from multiple layouts instead makes z_{FB} blend dynamics-specific information, averaging over futures and yielding policies that are sub-optimal even on the training layouts. The formal theorem can be found in the Appendix F.

3.2. Belief State Modeling

To address the interference issue (Section 3.1), we infer the latent environment context and augment FB input with this belief. A transformer encoder f_{dyn} processes a set of transitions $\{(s_t, a_t, s'_{t+1})\}_{t=1}^{N}$ to produce a latent context $h \in \mathbb{R}^d$, where H represents all possible inferred contexts. Since ordering is discarded, the encoder must identify dynamics-specific mismatches (e.g., layout geometry). Permutation invariance is crucial, as latent environment factors are order-agnostic. This setup enables zero-shot and few-shot learning (Snell et al., 2017).

Given episodes $(\{(s_t, a_t, s'_{t+1})_{c_i}\}_{t=1}^{N}$ we train a transformer encoder on unlabeled episodes (context length n) to infer a latent variable h that captures episode dynamics. The loss has two components: 1) h follows a Gaussian prior and is shared across trajectories and 2) A projection head combines h with (s_t, a_t) to predict s_{t+1}. Training can be end-to-end or staged - we found separate training of FB and z_{FB}.

For each trajectory we concatenate the inferred context vector h with the task vector z_{FB} to obtain augmented input $[h; z_{\text{FB}}]$ and condition only forward network as $\hat{M}_{\pi_z}(s_t, a_t, s_{t+1}) = F(s_t, a_t, [h; z_{\text{FB}}])^T B(s_{t+1})$. Empirically, conditioning the backward network B hurt performance, yielding an oversmoothed Q function that ignored environment structure. Thus, we kept B context-shared in all experiments (see Algorithm 1).

At test time, the agent is provided with a short, reward-free trajectory and it is passed to f_{dyn} to obtain h. By plugging the result into Equation 3, the greedy policy is obtained.

3.3. Structuring directions in the latent space

Insights from Section 3.1 showed that sampling task-vectors z_{FB} uniformly on the hypersphere encodes averaged policies, while Section 3.3 provided a solution through explicit context identification. We now combine these observations together through enhanced sampling z_{FB} around the inferred context h.

In Vanilla-FB, each state s draws $z_{\text{FB}} \sim \text{Unif}(\mathbb{S}^{d-1})$ with no inductive bias, so resulting policies π_z conflict with each other in CMDP setting, **even if additional explicit conditioning is introduced as before**. We replace uniform prior with a *von Mises-Fisher* (vMP) distribution centered at the context direction for episode $h = f_{\text{dyn}}(\{(s_i, a_i, s_{i+1})\})$ as $z_{h+\text{FB}} \sim \text{vMF}(\mu = h, \kappa)$ with κ controlling the spread or *diversity* of policies (left and middle figures from Figure 11). In practice, to draw $z_{h+\text{FB}}$ we first pick a simple vector (*e.g.*, the first basis vector), perturb with vMF noise, and finally rotate the result onto h with Householder reflection.

This enhancement has several benefits: 1) because directions h that differ in dynamics now occupy disjoint cones on the hypersphere, FB can fit the successor measure locally inside each cone, avoiding the destructive averaging effect quantified in Section 3.1 and 2) alignment procedure encourages the agent to explore policies that are plausible under its current belief while still injecting controlled diversity through κ.

Importantly, such a procedure not only has empirical benefits as we will show in Section 4, but also lowers bound from above in Theorem 1, *making it non dependent on number of environments* k. We provide formal theoretical result, intuition and proof in the Theorem F.2 (Section F).

4. Experiments

Can the belief estimation enable adaptation in FB? Previously, we provided the theoretical foundations and speculated on the matter why FB is unable to differentiate between

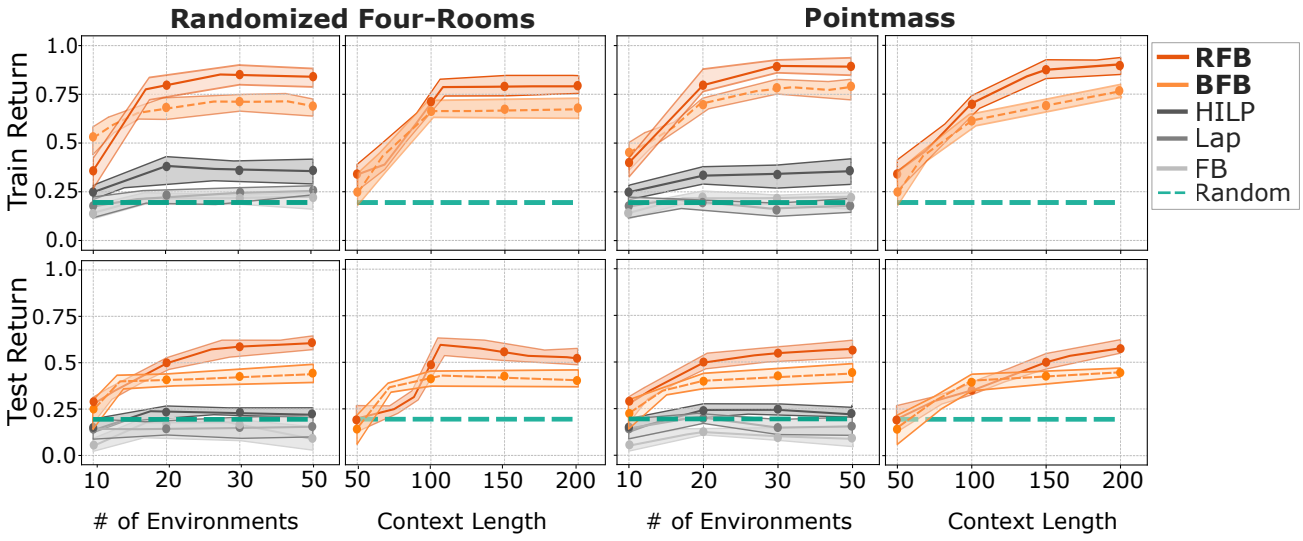


Figure 2. **Ablations on data diversity and context length of transformer encoder.** We show the influence of number of environments (data diversity) and context length on train and test performance in Four-Rooms and Pointmass environments. For data-diversity ablation, we see a clear performance boost up until some point, after which it platoes, as the Theorem 1 predicts. In our context-length ablation, we observe similar behaviour: performance improves as the context grows up to the length of a single episode, and then levels off. The results are averaged across three seeds, the opaque fill indicates standard deviation.

distinct dynamics and how we can use the belief estimation to overcome this. We refer to Table 1 and Figure 1 that show our empirical findings to support our claims.

Neither FB nor LAP are able to outperform a simple random baseline in PointMass and FourRoom, indicating that the policy they learn is most likely stuck due to averaging (see Section 3.1. Only HILP, which utilizes temporal structure to learn policy representations, is able to perform better than random policy.

In contrast, Belief-FB and Rotation-FB outperform every baseline method. Notably, our methods also demonstrate generalization capabilities beyond train data on unseen test environments.

Does change in context length input to the f_{dyn} impacts performance? We test how input trajectory length affects performance by varying context length (50 to 200) in Randomized Four-Rooms and Pointmass environments. Performance is poor with context shorter than an episode (100) as short trajectories only capture near-term goals. Longer sequences offer no extra benefit f_{dyn} already encodes sufficient information. Results show f_{dyn} generates robust representations h that distinguish between contexts in both train and test settings.

Does increase in dataset diversity make policies more robust? We investigate if diversifying CMDP training configurations improves performance. Intuitively, broader state-action coverage should yield more accurate successor measure estimation. Experiments support this: Figure 2 shows rapid improvement (up to $N \sim 25$) or BFB and BFB, while baselines match random policy performance. Once learned representationsor BFB and BFB, while baselines match random policy performance. Once learned representations h from f_{dyn} capture all variation modes (i.e., contexts), additional data provides marginal gains ($< 3\%$). These results align with Theorem 1.

5. Conclusion & Limitations

Belief-FB (BFB) and Rotation-FB (RFB) extend Forward-Backward (FB) representations to work with new dynamics. As we show, naive sampling of policy latents mixes conflicting transitions, causing interference. We propose to mitigate this by proposing BFB which adds a permutation-invariant transformer that encodes context (belief states) and conditions the policy on it. Moreover, additional rotation of the latent vectors so task-relevant abstractions align with environment-specific features, keeping policy regions separate. Both methods improve on prior FB variants theoretically and empirically, but they were tested on limited dynamics, add a diversity hyper-parameter κ and transformers become costly with long contexts. Future work should check if other zero-shot RL schemes suffer similar interference and scale these ideas to larger suites such as XLand-MiniGrid (Nikulin et al., 2024; 2025) and Kinetix (Matthews et al., 2025).

References

Agarwal, S., Sikchi, H., Stone, P., and Zhang, A. Proto successor measure: Representing the space of all possible solutions of reinforcement learning, 2025. URL https://openreview.net/forum?id=s9SVlWOcLt.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., and Silver, D. Successor features for transfer in reinforcement learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/350db081a661525235354dd3e19b8c05-Paper.pdf.

Beck, J., Vuorio, R., Liu, E. Z., Xiong, Z., Zintgraf, L., Finn, C., and Whiteson, S. A survey of meta-reinforcement learning, 2024. URL https://arxiv.org/abs/2301.08028.

Bickel, S., Brückner, M., and Scheffer, T. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pp. 81–88, 2007.

Blier, L., Tallec, C., and Ollivier, Y. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021.

Borsa, D., Barreto, A., Quan, J., Mankowitz, D. J., van Hasselt, H., Munos, R., Silver, D., and Schaul, T. Universal successor features approximators. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1VWjiRcKX.

Collaboration, O. X.-E. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

Dayan, P. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

Dorfman, R., Shenfeld, I., and Tamar, A. Offline meta learning of exploration, 2021. URL https://arxiv.org/abs/2008.02598.

Eysenbach, B., Chaudhari, S., Asawa, S., Levine, S., and Salakhutdinov, R. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=eqBwg3AcIAK.

Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pp. 2960–2967, 2013.

Frans, K., Park, S., Abbeel, P., and Levine, S. Unsupervised zero-shot reinforcement learning via functional reward encodings. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 13927–13942. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/frans24a.html.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Gregor, K., Jimenez Rezende, D., Besse, F., Wu, Y., Merzic, H., and van den Oord, A. Shaping belief states with generative environment models for rl. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2c048d74b3410237704eb7f93a10c9d7-Paper.pdf.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Jeen, S. and Cullen, J. Dynamics generalisation with behaviour foundation models. In *Workshop on Training Agents with Foundation Models at RLC 2024*, 2024. URL https://openreview.net/forum?id=A1u8YM7vuP.

Jeen, S., Bewley, T., and Cullen, J. Zero-shot reinforcement learning from low quality data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=79eWvkLjib.

Kirk, R., Zhang, A., Grefenstette, E., and Rocktäschel, T. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76: 201–264, 2023.

Kouw, W. M. and Loog, M. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785, 2019.

Laskin, M., Wang, L., Oh, J., Parisotto, E., Spencer, S., Steigerwald, R., Strouse, D., Hansen, S., Filos, A., Brooks, E., Gazeau, M., Sahni, H., Singh, S., and Mnih, V. In-context reinforcement learning with algorithm distillation, 2022. URL https://arxiv.org/abs/2210.14215.

Lee, J. N., Xie, A., Pacchiano, A., Chandak, Y., Finn, C., Nachum, O., and Brunskill, E. Supervised pretraining can learn in-context reinforcement learning, 2023. URL https://arxiv.org/abs/2306.14892.

Matthews, M., Beukman, M., Lu, C., and Foerster, J. N. Kinetix: Investigating the training of general agents through open-ended physics-based control tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zCxGCdzreM.

Modi, A., Jiang, N., Singh, S., and Tewari, A. Markov decision processes with continuous side information. In *Algorithmic learning theory*, pp. 597–618. PMLR, 2018.

Nikulin, A., Kurenkov, V., Zisman, I., Agarkov, A., Sinii, V., and Kolesnikov, S. Xland-minigrid: Scalable meta-reinforcement learning environments in jax, 2024. URL https://arxiv.org/abs/2312.12044.

Nikulin, A., Zisman, I., Zemtsov, A., and Kurenkov, V. Xland-100b: A large-scale multi-task dataset for in-context reinforcement learning, 2025. URL https://arxiv.org/abs/2406.08973.

Park, S., Kreiman, T., and Levine, S. Foundation policies with hilbert representations. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=LhNsSaAKub.

Pirotta, M., Tirinzoni, A., Touati, A., Lazaric, A., and Ollivier, Y. Fast imitation via behavior foundation models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. URL https://openreview.net/forum?id=SHNjk4h0jn.

Polubarov, A., Lyubaykin, N., Derevyagin, A., Zisman, I., Tarasov, D., Nikulin, A., and Kurenkov, V. Vintix: Action model via in-context reinforcement learning, 2025. URL https://arxiv.org/abs/2501.19400.

Rakelly, K., Zhou, A., Quillen, D., Finn, C., and Levine, S. Efficient off-policy meta-reinforcement learning via probabilistic context variables, 2019. URL https://arxiv.org/abs/1903.08254.

Sikchi, H., Agarwal, S., Jajoo, P., Parajuli, S., Chuck, C., Rudolph, M., Stone, P., Zhang, A., and Niekum, S. Rl zero: Zero-shot language to behaviors without any supervision. *arXiv preprint arXiv:2412.05718*, 2024.

Sikchi, H., Tirinzoni, A., Touati, A., Xu, Y., Kanervisto, A., Niekum, S., Zhang, A., Lazaric, A., and Pirotta, M. Fast adaptation with behavioral foundation models. *arXiv preprint arXiv:2504.07896*, 2025.

Sinii, V., Nikulin, A., Kurenkov, V., Zisman, I., and Kolesnikov, S. In-context reinforcement learning for variable action spaces, 2024. URL https://arxiv.org/abs/2312.13327.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning, 2017. URL https://openreview.net/forum?id=B1-Hhnslg.

Sønderby, C. K., Caballero, J., Theis, L., Shi, W., and Huszár, F. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Tarasov, D., Nikulin, A., Zisman, I., Klepach, A., Polubarov, A., Nikita, L., Derevyagin, A., Kiselev, I., and Kurenkov, V. Yes, q-learning helps offline in-context RL. In *Scaling Self-Improving Foundation Models without Human Supervision*, 2025. URL https://openreview.net/forum?id=B86JMHZUnc.

Teoh, J., Varakantham, P., and Vamplew, P. On generalization across environments in multi-objective reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tuEP424UQ5.

Tirinzoni, A., Touati, A., Farebrother, J., Guzek, M., Kanervisto, A., Xu, Y., Lazaric, A., and Pirotta, M. Zero-shot whole-body humanoid control via behavioral foundation models.

Touati, A. and Ollivier, Y. Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23, 2021.

Touati, A., Rapin, J., and Ollivier, Y. Does zero-shot reinforcement learning exist? *arXiv preprint arXiv:2209.14935*, 2022.

Uehara, M., Sato, I., Suzuki, M., Nakayama, K., and Matsuo, Y. Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint arXiv:1610.02920*, 2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wu, Y., Tucker, G., and Nachum, O. The laplacian in RL: Learning representations with efficient approximations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJlNpoA5YQ.

Xing, J., Nagata, T., Chen, K., Zou, X., Neftci, E., and Krichmar, J. L. Domain adaptation in reinforcement learning via latent unified state representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10452–10459, 2021.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

Zhu, C., Wang, X., Han, T., Du, S. S., and Gupta, A. Distributional successor features enable zero-shot policy optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=8IysmgZte4.

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: A very good method for bayes-adaptive deep rl via meta-learning, 2020. URL https://arxiv.org/abs/1910.08348.

Zisman, I., Kurenkov, V., Nikulin, A., Sinii, V., and Kolesnikov, S. Emergence of in-context reinforcement learning from noise distillation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=Y8KsHT1kTV.

Zisman, I., Nikulin, A., Sinii, V., Tarasov, D., Lyubaykin, N., Polubarov, A., Kiselev, I., and Kurenkov, V. N-gram induction heads for in-context rl: Improving stability and reducing data needs, 2025. URL https://arxiv.org/abs/2411.01958.

A. Extended Related Works and Background

A.1. Background

Contexual Markov Decision Process. Throughout paper we will be dealing with a Contextual Markov Decision Process (CMDP), defined by a tuple $\langle \mathcal{C}, \mathcal{S}, \mathcal{A}, \gamma, \mathcal{M} \rangle$, where $\mathcal{C}$ is a context space and $\mathcal{S}, \mathcal{A}$ are shared state and action spaces across environments. Function $\mathcal{M}$ maps particular context $c \in \mathcal{C}$ to respective MDP, *i.e.*, $\mathcal{M}(c) = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}^c, R^c, \mu^c, \gamma \rangle$ with context-dependent transition function $\mathcal{T}^c : \mathcal{S} \times \mathcal{A} \times \mathcal{C} \to \mathcal{S}$, μ^c being an initial distribution over states and $\gamma \in (0, 1)$ a discount factor. Intuitively, the context $c \in \mathcal{C}$ represents a fixed environmental configuration, such as obstacle positions, layout geometry, dynamics vector parameters or seed. Throughout this work, the context remains static within each episode, consistent with prior literature (Kirk et al., 2023; Modi et al., 2018; Teoh et al., 2025). A policy $\pi : \mathcal{S} \to \Delta\mathcal{A}$ is optimal for context c for the reward function R if it maximizes expected discounted future reward, *i.e.*, $\pi^*_{c,R}(s_0, a_0) = \arg\max_\pi \mathbb{E}[\sum \gamma^t R(s_t, a_t) | s_0, a_0, \pi, c]$.

When the context is fully observable, augmenting the state space with the given context reduces the CMDP to a standard MDP, eliminating the need to model distinct dynamics $\mathcal{T}^c$, rewards R^c or initial states μ^c. However, if the context is partially observable, the learned model must infer and track the uncertainty over true hidden configuration to maintain theoretical optimality guarantees. Such task can be framed as posterior estimation $p(c|\mathcal{H})$ or *belief* over possible contexts c given accumulated history H.

Most successful methods for deriving an optimal policy across arbitrary tasks from a task-agnostic dataset leverage successor features (Barreto et al., 2017; Borsa et al., 2019; Dayan, 1993; Park et al., 2024; Zhu et al., 2024) or their continuous counterpart, successor measures (Agarwal et al., 2025; Blier et al., 2021; Jeen et al., 2024; Touati & Ollivier, 2021; Touati et al., 2022). In this work, we focus on the latter framework, specifically its instantiation via forward-backward representations (Touati & Ollivier, 2021). Below, we briefly outline its key properties.

Zero-Shot RL. Given an offline dataset of transitions $\mathcal{D} = \{(s_i, a_i, s_{i+1})\}_{i=1}^{|\mathcal{D}|}$ generated by an unknown behavior policies, the agent's objective is to learn a unified abstraction of the environment without additional interaction. At test time, this abstraction helps to obtain optimal policy for *any* reward function r_{test} which defines a particular *task*. Reward function can be specified either as a small dataset of reward-labeled states $\mathcal{D}_{test} = \{(s_i, r_{test}(s_i)\}_{i=1}^{k}$ or as a direct mapping $s \to r_{test}(s)$. While some prior works assume access to the context labels (Gregor et al., 2019), we focus on the setting where the context is unknown and must be inferred from the data. Alternative formulations of zero-shot RL exist under other formalisms, and we refer to (Kirk et al., 2023) for comprehensive overview.

A.2. Literature

Domain Adaptation and Transfer Learning in RL. While our work will focus on domain adaptation applied to estimating successor measure for various dynamics mismatches, we start by briefly reviewing more general ideas in classic domain adaptation and refer to (Kouw & Loog, 2019) for detailed overview. Most methods for domain adaptation can be categorized into *importance-weighting* (Bickel et al., 2007; Sønderby et al., 2016; Uehara et al., 2016) and *domain-invariant feature learning* (Eysenbach et al., 2021; Fernando et al., 2013; Xing et al., 2021; Zhang et al., 2020) approaches. Former methods estimate the likelihood ratio of examples under samples from target domain versus samples from source, which is then used to recalibrate examples from the source domain. The latter approaches learn a unified representation of the environment, targeting to extract only task-relevant abstraction, negating distracting information.

The most relevant approach which enables FB representations to generalize across dynamics is *Contexual FB* (Jeen & Cullen, 2024). This approach uses importance-weighting formalism and introduces two classifiers, which estimate the likelihood of transitions (s_t, a_t) and (s_t, a_t, s_{t+1}) being from train or test context and augment the reward function to account for those discrepancies in the dynamics. If augmented reward function lies in the linear span of the $\mathcal{Z}$ space during FB training, then the policy can be extracted as described in Equation 3. However, such an approach requires training classifiers from scratch for each novel layout of the environment, limiting its applicability.

Meta-RL. Another major line of related works, Meta-Reinforcement Learning (Meta-RL), focuses on few-shot domain adaptation to unseen tasks or dynamics (Beck et al., 2024). The significant part of research in Meta-RL is dedicated to explicitly learning the *belief* by collecting a history of interactions with the environment on inference during test-time (Dorfman et al., 2021; Rakelly et al., 2019; Zintgraf et al., 2020). However, recent works show that it is possible to quantify

the *belief* without learning the posterior implicitly (Laskin et al., 2022; Lee et al., 2023; Polubarov et al., 2025; Sinii et al., 2024; Tarasov et al., 2025; Zisman et al., 2024; 2025). Leveraging in-context ability of transformers (Vaswani et al., 2017), one can learn an end-to-end supervised model, while the transformer's context will absorb into robust representation the adaptation-relevant information thus enabling fast adaptation. We also leverage this in-context ability to construct the belief representation of the dynamics the agent currently in, but instead operating in a zero-shot manner.

A.3. FB Training

In this section we describe the training procedure of FB in more details. Everything follows the notation from Touati & Ollivier (2021).

Assume that ρ is supported over all provided data, *i.e.*, it is non-zero everywhere.

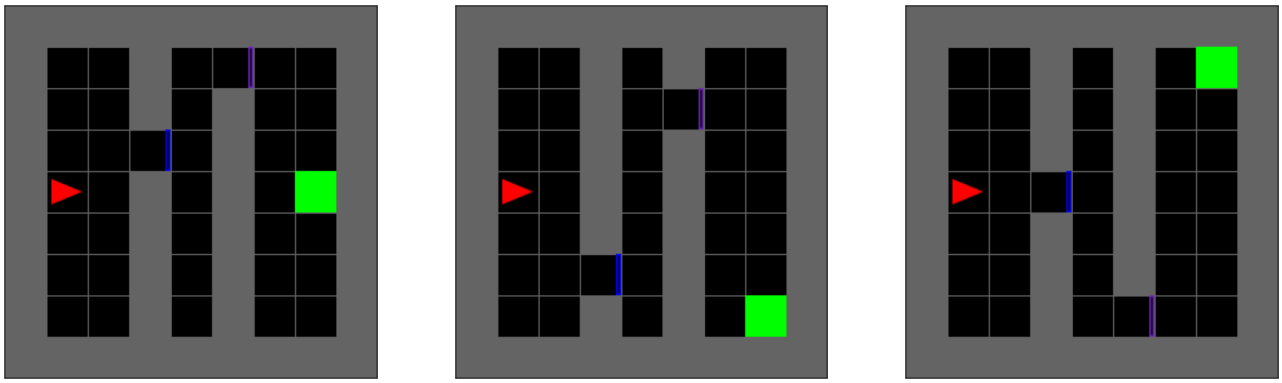
$$\mathcal{L}_{\text{FB}} = \mathbb{E}_{(s_t,a_t,s_{t+1},s_+)\sim\mathcal{D},z\sim\mathcal{Z}}[(F(s_t,a_t,z)^T B(s_+) - \gamma\hat{F}(s_{t+1},\pi_z(s_{t+1},z)^T\hat{B}(s_+))^2 - 2F(s_t,a_t,z)^T B(s_{t+1})] \quad (4)$$

Here, s_+ is a future outcome either from the same trajectory or randomly sampled from data. $\hat{F}, \hat{B}$ are target networks with Z being a task space, encoding all possible policies. The policy π_z is trained in an actor-critic formulation and parametrized as Boltzmann policy $\pi_{z_i}(\cdot|s_i) = \texttt{softmax}(F(s_i,\cdot,z_i)^T z_i/\tau)$ for continuous environments. Additionally, B is forced to be orthogonal for different s, which is enforced by contrastive loss $\mathbb{E}_{(s,s+)}[B(s)^T B(s_+)]$.

B. Environment Descriptions

B.1. Randomized-Doors

The Randomized-Doors MiniGrid environment (Figure 3) is a discrete-state, discrete-action finite horizon deterministic environment in which agent has an objective to go to goal location with maximum return of 1. Each episode terminates after 100 steps or after reaching goal location. The randomization determines possible open doors locations, fully specifying particular layout. In our experiments, the observation state of an agent consists of (x, y) coordinates tuple, making it partially observable. Such setting requires to properly update beliefs over unobservable layout configuration type. The action space consists of four actions, namely {`up`, `down`, `right`, `left`}, while (x, y) coordinates across both axes are bounded by grid size, which we take to be 9×9.



(a) First type (b) Second Type (c) Third Type

Figure 3. Several possible layouts are visualized, each corresponding to unique possible doors configurations. The agent is denoted as a red triangle. The task specification (goal position) with reward of 1 is denoted by green square and is also randomized. It is a custom implementation based on Empty MiniGrid (`https://minigrid.farama.org`).

B.2. Randomized Four-Rooms

The Randomized Four-Rooms MiniGrid environment Figure 4 is a modification of classic Four-Rooms and is a discrete-state, discrete-action, deterministic partially observable environment. For each episode, the maze layout (grid type) is generated

randomly, ensuring all of the four rooms are connected with exactly single door. Observation state consists of (x, y) coordinates, making this environment hard and checks whether agent could successfully estimate uncertainty over hidden configurations solely based on number of occurrence of each transition, recovering dynamics. In our experiments, we consider 11×11 bounds for height and width.

Observation space consists of raw discrete (x, y) coordinates on the grid, while actions correspond to a set of possible moves {`up, down, left, right`}. For every layout we record 500 episodes of length 100, yielding a dataset that covers almost all possible (s, a) transitions. For testing on unseen configurations, we fix agent starting position to coordinates of the first empty cell and evaluate performance across 3 static goal positions, farhest away from starting position.

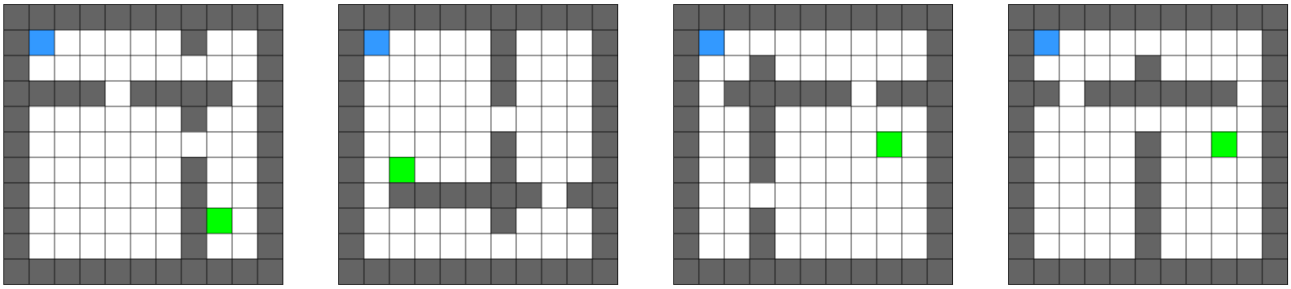


Figure 4. **Different layout configurations from randomized Four-Rooms environment.** During inference, the goal for the agent (depicted in blue) is to achieve green location. In our experiments we fix starting agent position and fix 3 goals, one for each room.

B.3. Ant-Wind

The AntWind environment is a modified version of the Ant locomotion task from the MuJoCo simulator, commonly used to test an agent's adaptability to changing dynamics. In this environment, an ant-like robot must learn to move forward while being subjected to external wind forces varying in magnitude and direction. In our experiments we consider 17 environments for training, covering three quadrants of possible wind directions on the circle, while leaving others for test, checking extrapolation on the fourth quadrant.

For our experiment, we collect dataset by training SAC (Haarnoja et al., 2018) on $3/4$ of all possible directions, which results in 16 environments and hold out the other $1/4$ for evaluation. Resulting dataset consists of 3400 transition tuples, where each environment configuration is represented as trajectory of length 256.

B.4. Randomized Pointmass

Randomized Pointmass is a modification of pointmass environment from D4RL (Fu et al., 2020). Each episode the environment grid structure is randomized, ensuring all cells are interconnected. The observation space consists of (x, y) transitions. Start position is determined as a first empty cell, while goal location is chosen to be the fartherst away from start (based on Manhattan distance) and ensuring existence of at least one valid trajectory (*e.g.*, through BFS).

Observation space consists of $(\texttt{global}\ x, \texttt{global}\ y)$ position, similar to Four-Rooms. We fix dataset size to be $1e^6$, only varying number of layouts and episodes per layout, while fixing episode length to 250. We use explore policy, which is a random policy with a portion of actions repeated ("sticky-actions").

C. Experimental Setup & Baselines

For experiments, we consider the following experimental setups: **(i)** discrete, partially observable Randomized Four-Rooms (Appendix B.2), **(ii)** continuous AntWind (Appendix B.3), and lastly **(iii)** continuous partially observable Randomized-Pointmass (Appendix B.4). We vary the number of train layouts for each experiment, while fixing the number of held-out *unseen* context settings to 20 for Randomized Four-Rooms and Randomized-Pointmass, and 4 for Ant-Wind. We perform comparisons against following **baselines**:

HILP (Park et al., 2024) is a method that learns state representations from offline data so that the distance in the learned representation space is proportional to the number of steps between two states in original space. **FB** (Touati & Ollivier,

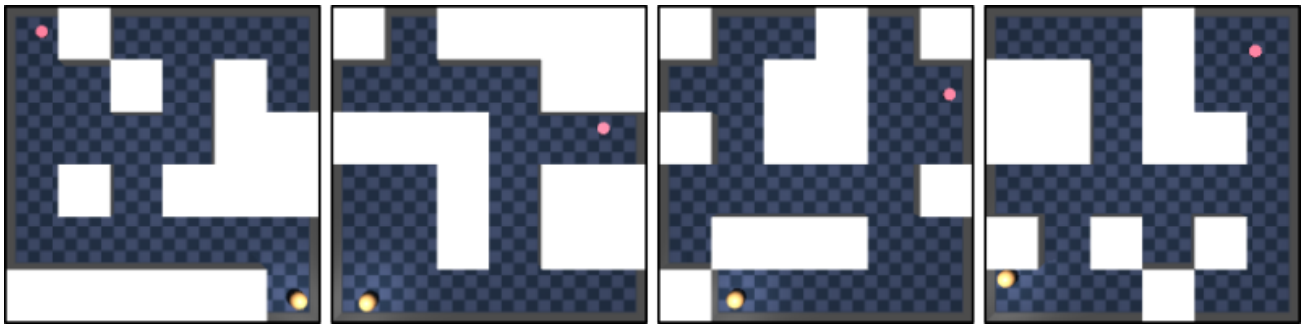


Figure 5. **Examples of pointmass grid variations.**

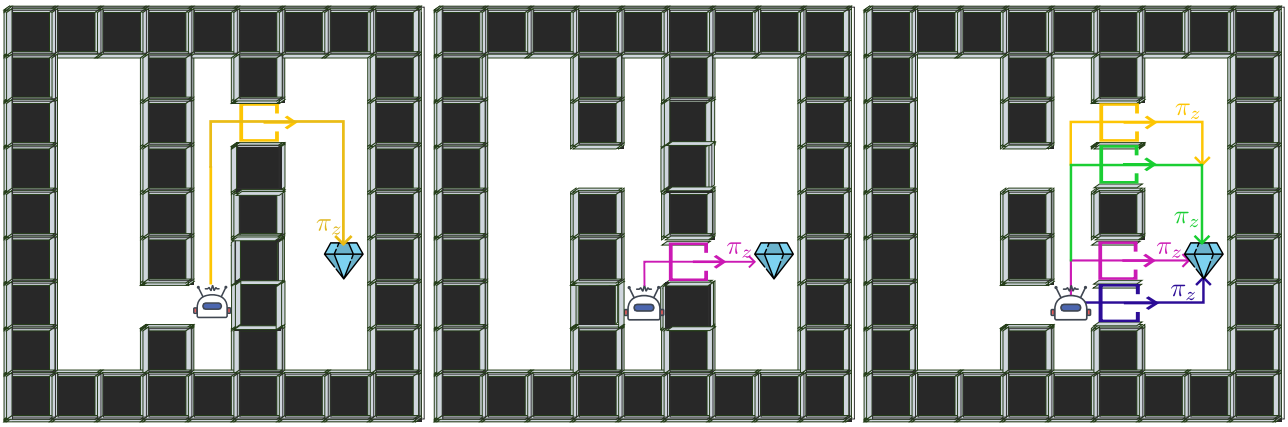


Figure 6. **Randomized-Doors environment for three different layouts, each produced through varying the grid structure (exact randomization procedure is a hidden variable)** (*left-middle*) From state s, the goal of an agent is to capture a diamond at target location by picking up the most probable policy π_z (yellow for the first type and purple for the second) to move to the closest open door based on internal representation. (*middle*) When there are multiple possible future outcomes in the training data from the same state, the π_z's (different colors) interfere with each other, leading to picking up an averaged policy.

2021) is an original version of the FB, described in Section 2. **Laplacian RL (LAP)** (Wu et al., 2019) constructs a graph Laplacian over state transitions from experience replay, then computes its eigenvectors to form low-dimensional representations that capture the environment's intrinsic structure. **Random** agent, which randomly explores the environment in a task-independent manner.

D. Additional Experiments

The interference problem discussed in Section 3.1 highlights a fundamental trade-off. Namely, FB is expressive enough to model any task, yet when it is trained in unsupervised manner across environments with distinct unobserved parameters, the lack of contextual conditioning forces it to average different dynamics rather than separate them. The resulting successor measure merges transitions from distinct layouts and entangles directions in the latent space $\mathcal{Z}$. To disentangle these directions we must represent uncertainty about the hidden context explicitly.

D.1. Do BFB and RFB capture hidden properties of the environment?

For an agent to refine its policy, it needs to keep track and update the uncertainty over possible environment configurations. Both Belief-FB and Rotation-FB accomplish this. Figure 9 illustrates this phenomenon visually. In Randomized-Door (left), the episodic trajectories from five layouts form non-overlapping clusters in the first two principal components of h, effectively disentangling different dynamics.

In Ant-Wind, the embeddings lie almost perfectly on a circle whose azimuth matches the underlying wind direction, generalizing smoothly to the 4 held-out wind angles. The quantitative results for evaluation in Table 1 (averaged across all

environments) reveal that the baseline methods fail to recover those environment-specific properties and therefore produce sub-optimal policies even for train cases. In particular, HILP tends to predict an average direction in Randomized Four-rooms and ignores obstacles, while FB outputs same policy and Q function for almost all environments. Figure 8 shows that Q function is properly estimated only for BFB and RFB, respecting wall positions.

D.2. How κ in RFB influences performance?

As described in Section 3.3, RFB concentration κ regularizes the diversity of policies for each environment. One the one hand, concentration should be high to ensure non-overlapping policy parametrized clusters π_z for different h, while at the same time it should not exceed certain value to control the diversity of policies in the environment, preventing collapsed solutions. Figure **??** shows that lower values of κ, meaning task-vectors z_{FB} are sampled with high deviation around h, likely producing overlapping clusters. As κ grows, task-vectors become more specialized, lowering variance which results in higher performance.

E. Experiments Details

Randomized-Doors. For didactic example from Section 3.1 we collect diverse dataset from different layout configurations (open door locations) such that visitation distribution over all states is non-zero. Black color denotes obstacles. The episode length is set to be 100, which is equal to the context length of the transformer encoder for this experiment. Overall, we collect 500 episodes per layout and coverage heatmap is visualized in Figure 7.

Table 1. Comparison of proposed approaches against baselines on **test** (unseen) environments. Results for Fourrooms and Pointmass are averaged across 20 mazes configurations.

Environment (Test)	Method					
	Random	Vanilla-FB	HILP	Lap	**Belief-FB**	**Rotation-FB**
Randomized-Fourrooms	0.05 $\pm$0.01	0.15 $\pm$0.06	0.2 $\pm$0.02	0.1 $\pm$0.1	0.4 $\pm$0.02	0.61 $\pm$0.02
Randomized-Pointmass	0.03 $\pm$0.01	0.1 $\pm$0.1	0.25 $\pm$0.02	0.1 $\pm$0.1	0.45 $\pm$0.05	0.55 $\pm$0.05
Ant-Wind	250 $\pm$200.0	250 $\pm$98.5	410 $\pm$40.5	290 $\pm$22.5	550 $\pm$50.5	640 $\pm$30.7

Table 2. Comparison of proposed approaches against baselines on **train** environments. Results for Fourrooms and Pointmass are averaged across 20 mazes configurations.

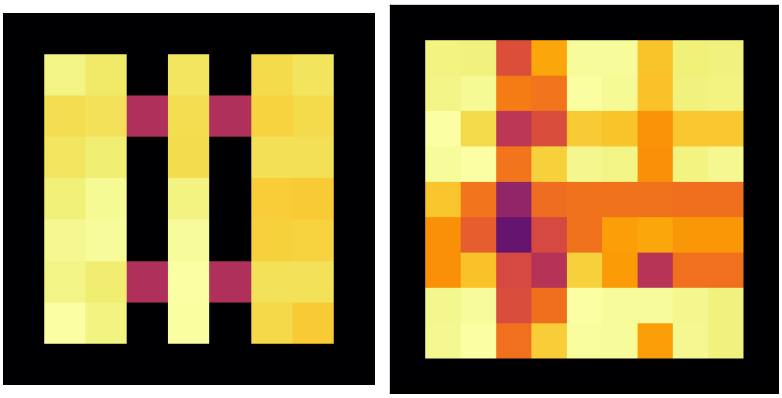
Environment (Train)	Method					
	Random	Vanilla-FB	HILP	Lap	**Belief-FB**	**Rotation-FB**
Randomized-Fourrooms	0.18 $\pm$0.02	0.25 $\pm$0.02	0.4 $\pm$0.02	0.2 $\pm$0.1	0.7 $\pm$0.02	0.85 $\pm$0.02
Randomized-Pointmass	0.0 $\pm$0.05	0.2 $\pm$0.2	0.45 $\pm$0.1	0.15 $\pm$0.15	0.76 $\pm$0.18	0.88 $\pm$0.2
Ant-Wind	-190 $\pm$250	390 $\pm$120	410 $\pm$90	340 $\pm$150	680 $\pm$80	740 $\pm$70

E.1. Dataset Generation

For Randomized Four-Rooms, we produce four training datasets with the following parameters:

# Transitions	**# layouts**	**# episodes per layout**	**episode length**
1000000	10	1000	100
1000000	20	500	100
1000000	30	250	100
1000000	50	150	100

Table 3. Details for Randomized Four-Rooms datasets



(a) Randomized-Doors (b) Randomized Four-rooms

Figure 7. State occupancy measures visualizations for collected datasets for discrete-based environments.

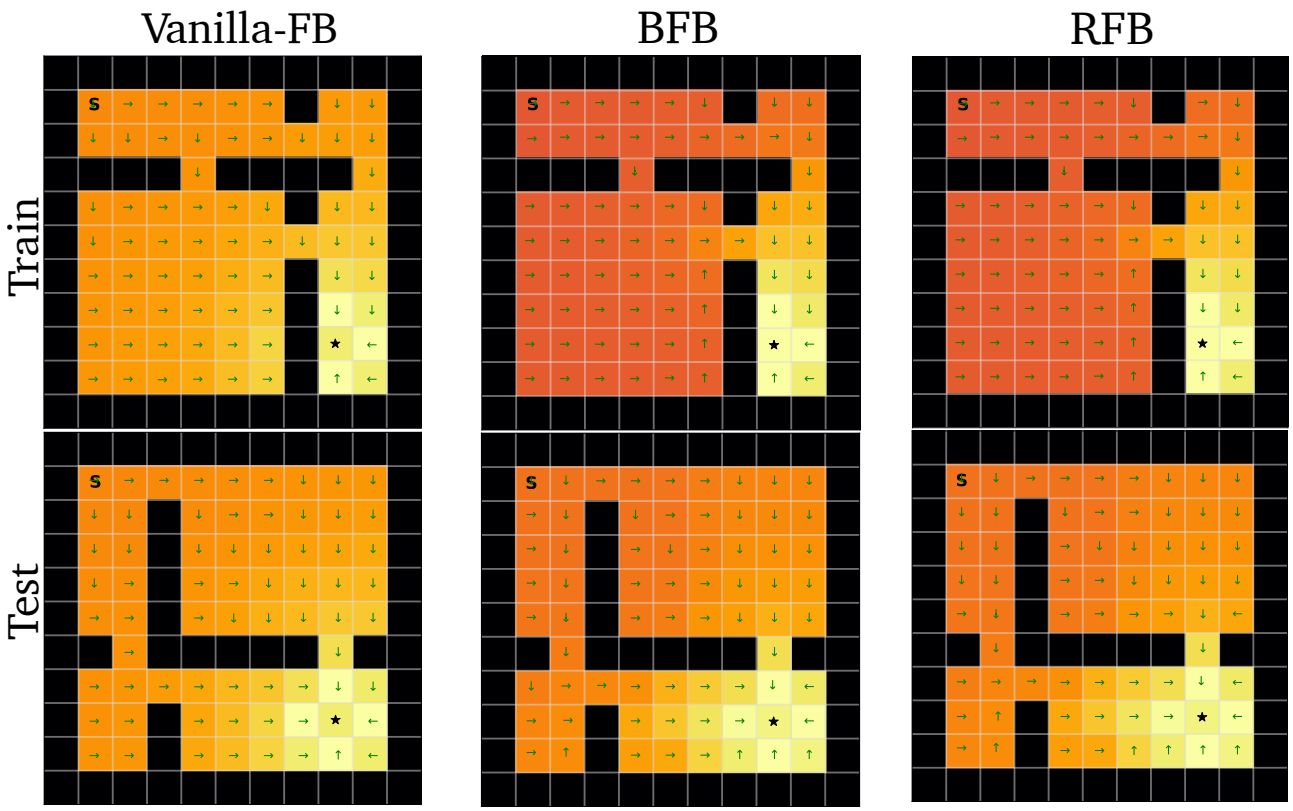


Figure 8. **Q-function and deterministic policy visualizations (Equation 3)** on Randomized Four-Rooms environment. Vanilla-FB ignores grid structure and resulting policy moves through obstacles. BFB and RFB do not have such issue.

Randomized Four-Rooms. For experiments on Randomized Four-Rooms during dataset collection we generate randomly grid layout, ensuring that each room is interconnected by exactly one door. For evalution we fix agent start position to $(1, 1)$ with the goal of reaching 3 other goals, specified at other rooms. Each episode terminates after 100 steps. The evaluation protocol is averaged success rate across 3 across 20 environments.

AntWind. For AntWind we first collect trajectories by varying wind direction d and training an expert-like SAC agent. After training, we collected evaluation trajectories from trained agent. This ensures that all directions are covered and explicitly sets dynamics context. As said in Experiments section, we train on 16 environments with wind directions corresponding to first 3 quadrants of circle, leaving other 4 (last quadrant) for hold out.

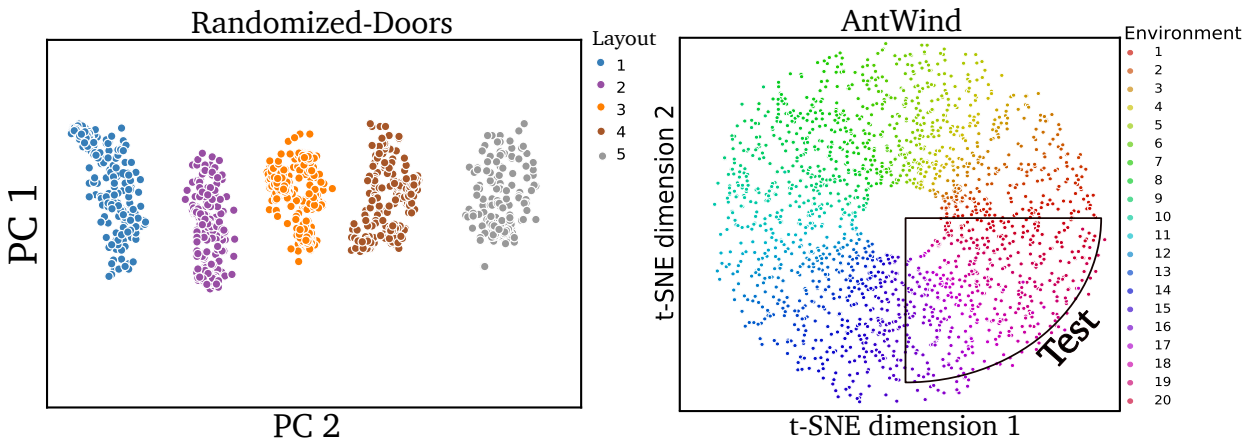


Figure 9. **2D projections of z_{dyn} inferred from different trajectories across number of different contexts (color), showing effective disentangling environments based on transition function or other mismatches. (*left*)** First two principal components are visualized for estimated z_{dyn} from five trajectories, each representing different layout type in Randomized-Doors. **(*right*)** Inferred context variables z_{dyn} recover hidden wind direction parameter in AntWind environment both for train and test, proving successful extrapolation properties.

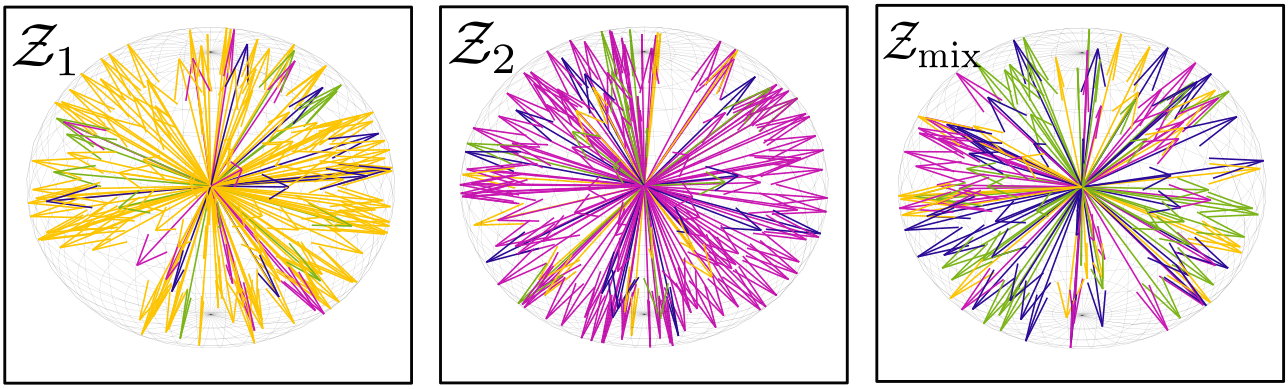


Figure 10. **Three different environment configurations from Figure 6 are visualized (yellow, purple and mixed trajectories).** For a fixed state s and same goal across configurations, arrows depict latent directions $z_{\text{FB}} \in \mathcal{Z}$ and colored by optimal action as $a_{color} = \arg\max_a F(s, a, z_{\text{FB}})^T z_{\text{FB}}$. (*left-middle*) When FB is trained on the two distinct configurations in separation, most of the latent directions agree on the optimal policy π_z. (*right*) When FB is trained on mix of CMDPs and at test time tasked with any particular configuration from train, obtained policy is ambiguous, since most policy-encoding directions do not agree on the action.

F. Theoretical Results

To formally study optimality guarantees of the zero-shot fast adaptation to new situations, we employ the following assumption commonly used for dynamics generalization (Eysenbach et al., 2021; Jeen & Cullen, 2024):

Assumption 1 (Coverage)**.** Let $\mathcal{P}^{\text{c}}(s_{t+1}|s_t, a_t)$ be a transition probability given small dataset of reward-free random interactions either from test or train context. Then, $\mathcal{P}^{c_{\text{test}}}(s_{t+1}|s_t, a_t) \Rightarrow \mathcal{P}^{c_{\text{train}}}(s_{t+1}|s_t, a_t)\, \forall s_t, s_{t+1} \in \mathcal{S}, a_t \in \mathcal{A}$.

F.1. Theorem 1

Let $\{M^{\pi_i}\}_{i=1}^k$ be a collection of successor measures corresponding to optimal policies $\{\pi_i\}_{i=1}^k$ for distinct CMDPs defined by hidden context configurations $c_i \in C$. Assume that ρ is the state-action distribution supported on the offline dataset used for FB training and $M^{\pi_i}(s, a, \cdot) \approx F(s, a, z_i)^T B(\cdot)$ is approximated via rank d factors. Define the worst-case approximation error ϵ_k over context-dependent k successor measures as follows:

$$\epsilon_k := \inf_{F,B} \max_{1 \le i \le k} ||M^{\pi_i} - F(\cdot, \cdot, z_i)^T B(\cdot)||_{L^2(\rho)}. \tag{5}$$

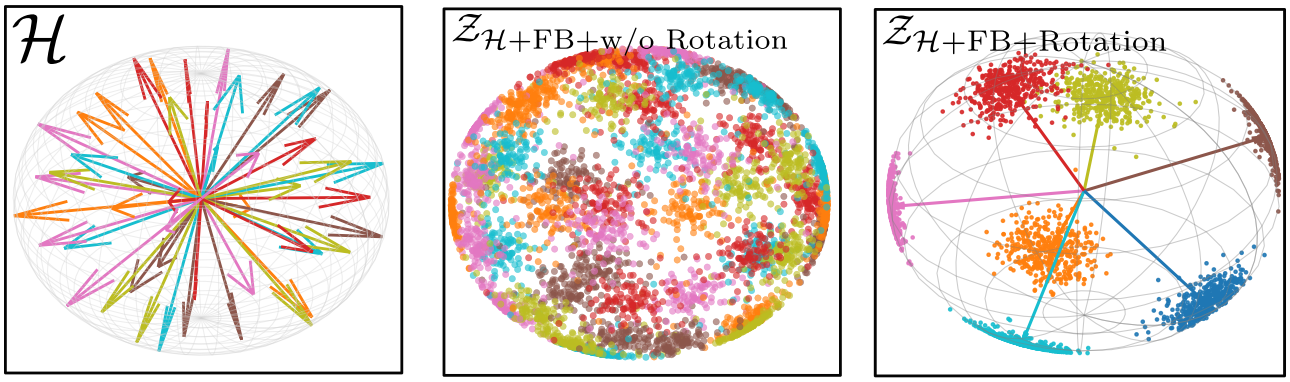


Figure 11. **Visualization of inferred contexts h from space of all possible contexts $\mathcal{H}$ (depicted as arrows) and task vectors z_{FB} (depicted as points on sphere boundary). Transitions from same CMDP colored the same. Concentration parameter κ defines spread of clusters.** (*left*) Untrained transformer f_{dyn} output for different transitions is unstructured and same transitions coming from same CMDP (identical colors) are not collinear. (*middle*) New sampling procedure aligns policy specific vectors z_{FB} with context specific h, but clusters overlap before training. (*right*) After training, h for transitions from the same context are aligned and policies z_{FB} do not interfere between different environment configurations.

Then, the extracted policy π_{z_i} for (s, a) satisfies:

Theorem 1 (Regret-bound for Multiple Dynamics)**.** *For any bounded reward $||r||_\infty \leq R$ and particular test-time CMDP,*

$$\mathbb{E}_{(s,a)\sim\rho_{test}}[Q_r^{\pi^*}(s,a) - Q_r^{\pi_{z_i}}(s,a)] \leq \frac{2\gamma\epsilon_k||r||_\infty}{(1-\gamma)^2}. \tag{6}$$

Because $\epsilon_{k+1} \geq \epsilon_k$ (monotonicity), the worst case regret per any CMDP at test time increases as more environments are included during training.

Intuitively, Theorem 1 tells that as number of environments k grows, **FB is forced to average over incompatible future dynamics**. However, this bound can be tightened, which we show in Section 3.3.

Lemma 1. *Theorems 8-9 from Touati & Ollivier (2021) prove this inequality for single instance of MDP, showing that if FB approximation error in $L^2(\rho)$ is at most ϵ then pointwise value gap is bounded by:*

$$(Q_r^* - Q_r^{\pi_{z_i}}) \leq \frac{\gamma}{1-\gamma}(P_{\pi^*} - P_{\pi_z})(I - \gamma P_{\pi^*})^{-1}E(z)r) \tag{7}$$

with $E(z)$ being a point-wise error matrix over state-actions as $E(z) = M^{\pi_z}(s,a,s') - F(s,a,z)^T B(s,a)$. Since

$$||(I-\gamma P)^{-1}||_\infty \leq \frac{1}{1-\gamma} \tag{8}$$

results in coefficient $2\gamma/(1-\gamma)^2$ in Equation 1.

Proof. Define a transition kernel P_i of CMDP at index i and M_{π_i} its successor measure. Let $E_i = M_{\pi_i} - F(s,a,z_i)^T B(\cdot) = M_{\pi_i} - \hat{M}_i$. Then, using $Q^* = (I - \gamma P_{\pi^*})^{-1} r$ value gap decomposes as

$$Q^* - Q^{\pi_{z_i}} = \gamma(I - \gamma P_{\pi^*})^{-1}(P_{\pi_*} - P_{\pi_{z_i}})(I - \gamma P_{\pi_{z_i}})^{-1} r \tag{9}$$

Since each of the resolvent factors (denote them as E_i)are at most $1/(1-\gamma)$ in L^∞, then from triangle inequality:

$$||Q^* - Q^{\pi_{z_i}}||_\infty \leq \frac{2\gamma}{(1-\gamma)^2}||E_i||_{L^2_\rho}||r||_\infty \tag{10}$$

From Assumption 1 on absolute continuity,

$$\mathbb{E}_{(s,a)\sim\rho_{\text{test}}}\{Q^* - Q^{\pi_{z_i}}\} \leq ||Q^* - Q^{\pi_{z_i}}||_\infty \tag{11}$$

Substituting this into Equation 10, gives desired inequality bound in Theorem 1. □

F.2. Theorem 2

Section 3.3 introduced a new sampling procedure of z_{FB}, which improves upon usual uniform sampling. This procedure can also be studied more formally.

Given an L possible contexual representations h of the environments coming from f_{dyn}, define a *cone* around each of the context axes $\{h^1, h^2 \dots h^L\} \in \mathbb{S}^{d-1}$, with the angle between any two latent vectors $\theta_{\max}$ set

$$\mathrm{C}_j = \{z_{\text{FB}} \in \mathbb{S}^{d-1} | \langle z_{\text{FB}}, h^j \rangle \geq \cos\theta_{\max}\} \tag{12}$$

Corresponding policy task vectors are defined for each cone $z^i_{\text{FB}} \in \mathrm{C}_{c(i)}$, with $c(i) \in \{1, \dots L\}$ being a classification function, mapping index i to one of the predifined context axes. For functions F, B define per environment error as:

$$\mathcal{E}_i(F, B) := ||M^{\pi_i} - F(\cdot, \cdot, z^i_{FB})^T B(\cdot)||_{L^2(\rho)} \tag{13}$$

With following optimization tasks:

$$\epsilon_k := \inf_{F,B} \max_{1 \leq i \leq k} \mathcal{E}_i(F, B), \quad \epsilon_j := \inf_{F,B} \max_{i \in \mathcal{S}_j} \mathcal{E}_i(F, B) \tag{14}$$

with $\mathcal{S}_j = \{i | c(i) = j\}$ being a set of task vectors (z_{FB}) indices that fall into the j-th cone of the latent space partition.

Theorem (Regret-bound under latent space partitioning)**.** *Under assumptions above, the Gram matrix of the directions $\{z_{FB}\}_{i=1}^k$ is block diagonal w.r.t. partition $\{S_j\}$ and*

$$\epsilon_k = \max_{j \leq L} \epsilon_j, \quad \epsilon_k \leq \epsilon_{k_{max}} \tag{15}$$

with $k_{max} := \max_j |S_j|$ being the size of a largest cone block.

In order to prove this theorem, assume that collection of contexual embeddings $\{h_j\}_{i=1}^L$ obtained from L environments are almost orthogonal.

Proof. Define a $k \times k$ Gram matrix as $G = \langle z^i_{\text{FB}}, z^j_{\text{FB}} \rangle$ with i, j corresponding to cone partition. Because cones, corresponding to different contexual embeddings h, are disjoint and lie in a span$\{h_i\}$, the resulting Gram matrix is block diagonal $G = \text{diag}(G^{(1)}, G^{(2)}, .., G^L)$. For a fixed rank d of F, B, the worst case approximation error is

$$\epsilon_k(F, B) = \max_{1 \leq i \leq k} ||M_{\pi_i} - \hat{M}_{\pi_i}||_{L^2(\rho)} = \max_{j \leq L} \max_{i \in S_j} ||M_{\pi_i} - \hat{M}_{\pi_i}||_{L^2(\rho)} \tag{16}$$

Since matrix G is block-diagonal, optimization of F, B decouples over blocks of G. Namely, minimizer on the full set is obtained by minimizing each block separately, hence:

$$\epsilon_k = \inf_{F,B} \epsilon_k(F, B) = \max_{j \leq L} \epsilon_j \tag{17}$$

By taking $k_{\max} = \max_j |S_j|$ and $\epsilon_k \leq \epsilon_{k_{\max}}$ for each block, we obtain desired inequality. □

Notably, such orthogonal cone partitioning eliminates interference. Once each cone has its own slice of the latent space, adding more cones does not enlarge the worst-case error bound, and with representation capacity of F and B being $d \geq k_{\max}$ the FB model can reach zero approximation error in principle.

Intuitively, Theorem F.2 states that after the partitioning procedure of the latent space into non-overlapping clusters based on context representations h, the global worst-case FB approximation error $\epsilon_k = \max_{j \leq L} \epsilon_j$ is determined only by the cluster whose error ϵ_j is largest. Importantly, this bound *does not depend on number of training environments* k. We provide a more formal treatment and a full proof in Appendix F.

G. Implementation Details

G.1. Forward-Backward Representations

G.1.1. GPUs

We run each experiment on 4 Nvidia 4090.

Table 4. **Hyperparameters for FB** The additional hyperparameters for Belief-FB and Rotation-FB are highlighted in

Hyperparameter	**Value**
Latent dimension d	150 (100 for discrete)
F / ψ dimensions	(1024, 1024)
B / φ dimensions	(256, 256, 256)
Preprocessor dimensions	(1024, 1024)
Std. deviation for policy smoothing σ	0.2
Truncation level for policy smoothing	0.3
Learning steps	1,000,000
Batch size	1024
Optimiser	Adam
Learning rate	0.0001
Learning rate of f_{dyn}	0.0001
Discount γ	0.99
Activations (unless otherwise stated)	GeLU
Target network Polyak smoothing coefficient	0.05
z-inference labels	10,000
z mixing ratio	0.5
κ	50, 100 for Pointmass
Contexual representation h dimension	150 (100 for discrete)
Next state predictor g_{pred}	(256, 256, 256)

G.1.2. ARCHITECTURE

The forward-backward architecture described below mostly follows the implementation by (Touati et al., 2022). All other additional hyperparameters are reported in Table 4.

Forward Representation $F(s, a, z)$. The input to the forward representation F is always preprocessed. State-action pairs (s, a) and state-task pairs (s, z) have their own preprocessors which are feedforward MLPs that embed their inputs into a 512-dimensional space. These embeddings are concatenated and passed through a third feedforward MLP F which outputs a d-dimensional embedding vector. Note: the forward representation F is identical to ψ used by USF so their implementations are identical (see Table 4).

Backward Representation $B(s)$**.** The backward representation B is a feedforward MLP that takes a state as input and outputs a d-dimensional embedding vector.

Actor $\pi(s, z)$**.** Like the forward representation, the inputs to the policy network are similarly preprocessed. State-action pairs (s, a) and state-task pairs (s, z) have their own preprocessors which feedforward MLPs that embed their inputs into a 512-dimensional space. These embeddings are concatenated and passed through a third feedforward MLP which outputs a a-dimensional vector, where a is the action-space dimensionality. A `Tanh` activation is used on the last layer to normalise their scale. Note the actors used by FB and USFs are identical (see Table 4).

Misc. Layer normalisation and `Tanh` activations are used in the first layer of all MLPs to standardise the inputs as recommended in original paper for both discrete and continuous becnhmarks.

G.2. Task Sampling Distribution $\mathcal{Z}$

Vanilla-FB. FB representations require a method for sampling the task vector z at each learning step. (Touati et al., 2022) employ a mix of two methods, which we replicate:

1. Uniform sampling of z on the hypersphere surface of radius $\sqrt{d}$ around the origin of $\mathbb{R}^d$,
2. Biased sampling of z by passing states $s \sim \mathcal{D}$ through the backward representation $z = B(s)$. This also yields vectors on the hypersphere surface due to the $L2$ normalization described above, but the distribution is non-uniform.

We sample z 50:50 from these methods at each learning step as in original work by (Touati & Ollivier, 2021).

Rotation-FB. After transformer f_{dyn} pretraining stage, RFB at each gradient step chooses task-conditioning vector z_{FB} based on **i)** context representation h acting as axes coming from f_{dyn} and **ii)** drawing task encoding vectors z_{FB} around this axes. We also perform normalization as in Vanilla-FB by projecting resulting vector on a surface of hypersphere of radius $\sqrt{d}$.

Stage ii) is implemented as drawing samples as $z_{\text{FB}} \sim \text{vMF}(\mu = h, \kappa)$. In order to remove high computational costs, we implement this sampling procedure through Householder reflection around context axes, by first drawing z from one of the basis vectors (*e.g.*, north pole) and then performing rotation. This is depicted Pseudocode section Section 1:

G.3. Pseudocode

Algorithm 1 Belief-FB Training

Input: offline diverse dataset $\mathcal{D}$ consisting of transitions based on hidden configuration variable c_i
Initialize transformer encoder f_{dyn_θ}, F_η, B_ω, number of gradient steps for transformer pre-training K, context length T, Polyak coefficient, β, batch size B learning rates $\lambda_f, \lambda_F, \lambda_B$
while update steps < K **do**
  sample batch of B trajectories of length T $\{(s_{i,t}, a_{i,t}, s_{i,t+1})\}_{i=1,\dots B, t=1,\dots,T} \sim \mathcal{D}$
  $(\boldsymbol{\mu}_i;\ \log \boldsymbol{\sigma}_i), = f_{\text{dyn}_\theta}\big(\{s_{i,t}, a_{i,t}, s_{i,t+1}\}_{t=1}^{M}\big), i = 1, \dots, B,$
  $\boldsymbol{z_i} = \boldsymbol{\mu}_i + \boldsymbol{\epsilon}_i \odot \exp\big(\log \boldsymbol{\sigma}_i\big),$
  $\mathbf{Z}_{i,t} = \boldsymbol{z}_{\text{dyn}_i},\ t = 1, \dots, T$ # Representation z_{dyn} is shared across each sequence
  $\hat{s}_{i,t+1} = g_{\text{pred}}(s_{i,t}, a_{i,t}, \mathbf{Z}_{i,t}) \qquad t = 1, \dots, T,\ i = 1, \dots, B$
  $\mathcal{L}_{\text{context}} = \frac{1}{BT}\sum_{i=1}^{B}\sum_{t=1}^{T}\big\|\hat{s}_{i,t+1} - s_{i,t+1}\big\|_2^2$
  $\theta_{f_{\text{dyn}}} \leftarrow \theta_{f_{\text{dyn}}} - \lambda_f \nabla_\theta \mathcal{L}_{\text{context}}(\theta)$
end while
while not converged **do**
  $\eta_F \leftarrow \eta_F - \lambda_F \nabla_{\eta_F} J_{(F,B)}(\eta_F)$ # FB training, Equation 4
  $\omega_B \leftarrow \omega_B - \lambda_B \nabla_{\omega_B} J_{(F,B)}(\omega_B)$
end while

Algorithm 2 Sampling z_{FB} for RFB

input B (batch size), d (latent dimension), anchor matrix $\mathbf{H} \in \mathbb{R}^{B\times d}$, κ (concentration)
output $\mathbf{Z} \in \mathbb{R}^{B\times d}$

Normalize anchors: $\mathbf{u}_i \leftarrow \mathbf{H}_i / (\|\mathbf{H}_i\|_2 + \varepsilon)$ {for $i = 1, \dots, B$}
$\mathbf{S} \leftarrow$ VMF_SAMPLE_NORTHPOLE(B, d, κ) {draw B VMF samples}
for $i \leftarrow 1$ **to** B **do**
  $\mathbf{R}_i \leftarrow$ HOUSEHOLDER_ROTATION$(\mathbf{u}_i)$
  $\mathbf{z}_i \leftarrow \mathbf{R}_i\, \mathbf{S}_i$
end for
$\mathbf{Z} \leftarrow$ PROJECT_TO_SPHERE$\big(\{\mathbf{z}_i\}_{i=1}^{B}\big)$
return $\mathbf{Z}$