Global–Local Decomposition of Contextual Representations in Meta-Reinforcement Learning

Nelson Ma, Junyu Xuan, *Senior Member, IEEE*, Guangquan Zhang, and Jie Lu, *Fellow, IEEE*

***Abstract*—Meta-reinforcement learning (meta-RL) algorithms extract task information from experienced context in order to reason about new tasks, and facilitate rapid adaptation. The quality of these contextual representations (or embeddings) is therefore crucial for a meta-RL agent to make effective decisions in unknown environments. Current methods predominantly assume the existence of a single underlying task, but using a single contextual embedding may not be expressive enough to fully capture the broader distribution of task variations that an agent might encounter. Decomposing that information into different representations can allow them to capture more relevant features in context space while applying additional structure that aids downstream exploitation. In this article, we develop global–local embeddings for contextual meta-RL (GLOBEX), an off-policy contextual meta-RL algorithm that decomposes the contextual representation into separate global and local embeddings. The learning process maximizes information retained by the embeddings and utilizes a mutual information constraint to encourage decoupling. Illustrative examples show that our method effectively adapts by identifying global task dynamics and exploiting temporally local signals. In addition, GLOBEX outperforms existing state-of-the-art meta-RL algorithms on standard MuJoCo benchmarks.**

***Index Terms*—Context, contextual meta-reinforcement learning (meta-RL), decomposition, meta learning, reinforcement learning, representation.**

I. Introduction

META-REINFORCEMENT learning (meta-RL) is a subfield of reinforcement learning that has attracted significant interest in recent years as a way to address common pitfalls in traditional RL methods, such as their prohibitive sample cost in training and their inability to generalize to similar environments [1], [2]. By leveraging knowledge and experience gained from previous tasks, meta-RL enables an agent to quickly adapt to new tasks or environments—in other words, aiming to *learn to learn faster*. This has seen success in transferring some skills across traditional research benchmarks, different Atari games [3] and a wide variety of robotics tasks [4], [5], [6]. One prominent approach to embed memory of past experiences into an RL agent is with *contextual meta-RL*, which explicitly infers both task distribution and the task at hand by embedding past experience into a contextual representation, then learns optimal behavior to either quickly learn more about the task, or exploit knowledge of the task itself [7]. This method affords a variety of advantages; if the task can be inferred successfully, then the burden of task inference can be removed from the agent policy, allowing for more efficient adaptation with fewer environment samples, while also providing a principled framework to approach the exploration-exploitation dilemma.

A core component of contextual meta-RL is the contextual encoder, which extracts information from current and previous RL transitions into a probabilistic latent variable that represents the inferred task. As the contextual embedding shoulders the burden of extracting essential task features for the RL agent, its quality is crucial for effective adaptation at test-time. To date, most prominent contextual meta-RL methods assume that there is a single underlying task representation. However, this approach might not sufficiently express all potential sources of task variation, leading to multiple tradeoffs in task representation design. For example, a single task embedding could be fixed during a full trajectory [8], which may not have the expressiveness to reliably adjust to temporally recent information that may be useful to an agent, such as remembering a map seen briefly at the beginning of a maze [9]. Alternatively, an agent using a time-variable task embedding may continually update its task prediction and may not behave in a manner consistent with a single task hypothesis [10]. Therefore, it is of interest to investigate new approaches in extracting task information that may enjoy the best of both worlds.

To address this challenge, we propose utilizing global–local decomposition to separately embed a global latent variable focused on static features of task inference, and a local variable that summarises and exploits task information from more temporally recent transitions. Decomposing context in this manner provides two primary advantages; first, having two separate contextual encoders, if trained effectively, will allow the combined global and local embeddings to capture more relevant features in the context space than an individual contextual embedding. Second, the added structure to the contextual representations provides an inductive bias that, when passed on to an RL policy, enables principled task

Received 29 August 2024; revised 24 November 2024 and 8 January 2025; accepted 14 January 2025. Date of publication 10 February 2025; date of current version 7 March 2025. This work was supported in part by the Australian Research Council under Australian Laureate Fellowships under Grant FL190100149, and in part by the Discovery Early Career Researcher under Award DE200100245. This article was recommended by Associate Editor H. M. Schwartz. *(Corresponding author: Jie Lu.)*

The authors are with the Australian Artificial Intelligence Institute (AAAI), University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: nelson.y.ma@student.uts.edu.au; junyu.xuan@uts.edu.au; guangquan.zhang@uts.edu.au; jie.lu@uts.edu.au).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2025.3533035.

Digital Object Identifier 10.1109/TCYB.2025.3533035

adaptation at test-time. This technique has been effectively applied in areas, such as time series analysis [11] and computer vision [12], [13], [14], and provides a clear framework to decouple gathered task information into global and local components.

The primary contributions of this article are as follows.

1) We detail the novel application of global–local decomposition of probabilistic task embeddings in contextual meta-RL, which extracts more information at meta-test time while enjoying the inductive benefits of global–local structure. To achieve this, we develop unique training procedures for each embedding, while utilizing a mutual information (MI) constraint novel to meta-RL to ensure both embeddings remain expressive and relevant.
2) In doing so, we introduce our off-policy contextual meta-RL algorithm, global–local embeddings for contextual meta-RL (GLOBEX). We demonstrate that GLOBEX outperforms peer meta-RL algorithms on classic research benchmarks, such as the MuJoCo physics suite [15], and that GLOBEX-augmented MetaCURE [16] substantially improves performance in the complex MetaWorld benchmark [5].

In this article we will first define the meta-RL problem and highlight key notation before discussing related work, particularly within contextual meta-RL. Section IV will introduce global–local decomposition of context alongside the GLOBEX algorithm in detail, while also providing specific implementation details. Finally, we showcase the performance of GLOBEX and the impact of effective global–local decomposition of context in Section V across both standard and complex benchmarks. A public implementation of GLOBEX can be found on Github: https://github.com/nelma-7/globex.

II. Problem Statement

In this section, we formalize the meta-RL problem and align on notation for the rest of this article. Let there be a collection of K Markov decision processes (MDPs) $M_i = \{S, A, P^i, R^i\}$ sampled from a broader distribution $M_i \sim p(M)$. We will refer to these MDPs colloquially as tasks, as they describe a specific environment and reward function that our meta-RL agent must adapt to. Each task M_i contains a state space S, action space A, transition matrix $P^i = p_i(s'|s, a)$, and reward function $R^i = r_i(s, a)$. It is typically assumed that there is some common structure among the different tasks M_i, which must be discovered and exploited by the meta-RL algorithm. In addition, the variation across tasks is assumed to be characterized solely by the reward function R^i, and the transition matrix P^i. Here, we will limit our considerations to sets of tasks that fulfill these conditions.

The meta-RL objective is then to find an adaptation procedure that learns a policy $\pi_{\theta_i}(a|s, c^{M_i})$ that maximizes cumulative expected reward over the whole distribution of tasks $\mathbb{E}_{M_i \sim p(M)} \mathbb{E}_{\pi_{\theta_i}}[r_i(\tau)]$. In contextual meta-RL, this policy is conditioned on a history of past transitions c^{M_i} that we denote as context, with a single transition $c_n^{M_i} = (s_n, a_n, r_n, s'_n)$, and a batch of N transitions $c^{M_i} = \{c_n^{M_i}\}_{1:N}$. To simplify notation we will drop the task subscript M_i unless necessary, and denote this parameterized policy as π_θ for the remainder of this article. Finally, we will use the subscript $n \in \{0, \ldots, N\}$ when discussing training batches that sample random individual transitions, and $t \in \{0, \ldots, T\}$ when discussing roll-out trajectories at meta-test time.

III. Related Work

Meta-RL: Meta-RL techniques can be characterized by how they embed prior knowledge into the algorithm in order to accelerate the learning process. Adopting the terminology used by Beck et al. (2023) [7] parameterized policy gradient (PPG) methods optimize policy parameters and hyperparameters in an outer optimization loop to maximize performance across the set of meta-training tasks [17], [18], [19], [20], [21], [22], and retain flexibility in the algorithm used to learn individual RL tasks. However, while PPG meta-RL enjoys attractive theoretical properties when it comes to consistency and asymptotic performance, it faces challenges when a limited number of policy parameter updates prove insufficient for adaptation [23], [24]. Recurrent methods aim to encapsulate the entire meta-learning process within a recurrent neural network (RNN) or similar structure [25], [26], [27], [28], [29], and allow the network to learn optimal exploration and exploitation behavior in a closed-box approach. Recurrent agents have shown excellent performance post-adaptation when fully trained, but are known to be difficult and inconsistent to train, and generalize poorly to out-of-distribution tasks [30].

Contextual Meta-RL: As previously mentioned, our work lies in the realm of contextual meta-RL, which utilizes a probabilistic latent variable to extract useful information from past learning experience. By shifting the burden of experiential adaptation away from the policy, contextual methods maintain the capability for rapid adaptation without needing to wait for gradient updates at test-time, while applying additional structure to the agent to simplify the learning problem [7]. The latent embedding in contextual meta-RL can be trained in either a *generative* or *model-free* manner. Methods that take a generative approach [9], [31], [32] aim to uniquely identify the MDP representation of an RL task by reconstructing the reward and transition functions of all tasks in the meta-RL training set, conditioned on the latent task variable. Alternatively, it is also possible to train the task variable in a model-free manner by either modeling the task-conditional Q-functions, or maximizing policy returns [8], [33]. In our method, we chose to take the latter approach by using gradients from the actor-critic Q-function loss, as we found empirically that it improved training stability.

Another key design choice centers around how often algorithms sample from (or update) the estimated task distribution. Methods that set a fixed latent variable for a full roll-out [16], [34], [35], [36] ensure that the agent moves according to a consistent task belief, but require dedicated exploration episodes before adaptation. Other methods choose to make the latent task variable time-dependent, and update the task distribution multiple times during an episode [10], [37], [38]. This enables the agents to exhibit more flexible exploratory

behavior and facilitates mid-episode adaptation, sacrificing behavioral consistency for adaptation speed. Our method seeks to bring together the best of both worlds, utilizing a fixed global variable, that is, trained to extract static features of task information, and a local variable updated during roll-out to embed temporal factors and enable further adaptation after seeing more recent observations.

Some recent methods seek to augment the exploratory capabilities of base algorithms by utilizing optimized exploration policies [9], [16], considering specific exploratory or curiosity-based terms [39], [40], [41], or when best to explore [42], [43]. There has also been significant attention on *offline* meta-RL, which aims to learn adaptive policies only utilizing previously collected data, and utilizes a variety of strategies to minimis the resulting distributional mismatch between data seen during training and experienced during testing [44], [45], [46], [47], [48]. Although neither exploratory incentives nor offline training are a primary focus of this article, we note in Section V-C that a GLOBEX-style global–local decomposition of contextual embeddings also improves the performance of algorithms like MetaCURE [16]. Further adaptation of this method to specifically leverage learned local representations for improved exploration, or to enable offline training can be considered to be future work on this particular algorithm.

Finally, we note that some recent works have also considered the decomposition of RL context into multiple, often decoupled representations. This has been explored through a variety of means; contrastive learning [35], [49], [50], parameterizing them as hidden states of an RNN encoder [51], learning successor features [36], and utilizing mixtures to model task distributions [32], [52], [53], [54], [55]. Parameterizing the latent representation as a Gaussian mixture can provide a hierarchical approach toward embedding tasks (or task clusters) and subtasks, but is sensitive to factors that may not be clear without prior information, such as the number of mixture components. Our method differs from those focused on contrastive learning as it explicitly minimizes the MI between two separate contextual embeddings, instead of between the contextual embeddings of two separate tasks. Furthermore, the induced global–local structure does not require additional tuning of the number of task clusters as required by GMM latent representations.

IV. Global–Local Decomposition of Contextual Embeddings

In this section, we present the framework of our algorithm GLOBEX, which trains a paired global–local encoder network to effectively learn contextual embeddings that are then exploited during meta-testing. We describe the components of the algorithm and its training in Section IV-A, and meta-test time specifics in Section IV-B. Pseudocode for GLOBEX during meta-training and meta-testing is detailed in Algorithms 1 and 2, respectively.

Algorithm 1 GLOBEX Meta-Train

Input: Batch of training tasks $\{M_i\}_{i=1,\dots,T}$ from $p(M)$, Sampling scheme S_c
Initialise replay buffers B^i for each training task
Initialise policy π_θ, encoders q_{ϕ_g} and q_{ϕ_l}, decoder d^T_ψ, and vCLUB network g_γ
while not done **do**
  # Gather data for all tasks
  for each M_i **do**
    Initialise context $c^{M_i} = \{\}$
    Sample $z^g \sim p_(z^g)$ and $z^l_0 \sim p_(z^l_t)$ from priors
    for ep in collection_episodes **do**
      Rollout policy $\pi_\theta(a_t|s_t, z^g, z^l_t)$, add steps to B^i
      Sample $c^{M_i} = \{(s_j, a_j, s'_j, r_j)\}_{j\in 0:N} \sim S_c(B^i)$
      Update $z^g \sim q_{\phi_g}(z^g|c^{M_i})$
    end for
  end for
  # Perform training
  for step in training_steps **do**
    for each M_i **do**
      Sample context $c^{M_i} \sim S_c(B^i)$, RL batch $b^i \sim B^i$
      Sample $z^g \sim q_{\phi_g}(z^g|c^{M_i})$
      Sample $z^l_n \sim q_{\phi_l}(z^l_n|c^{M_i}_n)$ for all n in $0, \dots, N$
      Calculate $\mathcal{L}^i_{transition}$ according to (1)
      Calculate $\mathcal{L}^i_{ELBO}$ according to (2)
      Calculate $\mathcal{L}^i_{MI}$ according to (3)
      Calculate SAC losses $\mathcal{L}^i_Q(b^i, z^g, z^l)$, $\mathcal{L}^i_\pi(b^i, z^g, z^l)$, $\mathcal{L}^i_V(b^i, z^g, z^l)$ according to Equations (5), (6), (7)
    end for
    Train decoder parameter ψ by minimising $\mathcal{L}_{transition}$
    Train global-local encoder parameters ϕ_g, ϕ_l by maximising $(\mathcal{L}_{\text{ELBO}} - \mathcal{L}_{MI})$
    Train vCLUB estimator parameter γ by maximising $\mathcal{L}_{\text{vCLUB}}$ according to (4)
    Train SAC policy parameters by minimising $\mathcal{L}_Q, \mathcal{L}_\pi, \mathcal{L}_V$
  end for
end while

A. Global–Local Embeddings

The primary feature of our method is the introduction and training of two distinct latent probabilistic random variables, a global embedding z^g, that is, trained to capture task information, and a local embedding z^l_n that captures local behavioral nuances, and is updated after each step during roll-out. The policy $\pi_\theta(a_n|s_n, z^g, z^l_n)$ is conditioned on both embeddings for adaptation to new tasks and environments. Fig. 1 outlines the structure of GLOBEX and illustrates the training process, which is described in this section.

Global Embedding: In GLOBEX, the global embedding z^g remains fixed for a whole episode and represents static, global features learned from the task and environment. Given a context training batch that we denote as *global context*, $c^g = \{c_n\}_{1:N}$, the global encoder learns to extract relevant features while simultaneously optimizing the policy to either gather

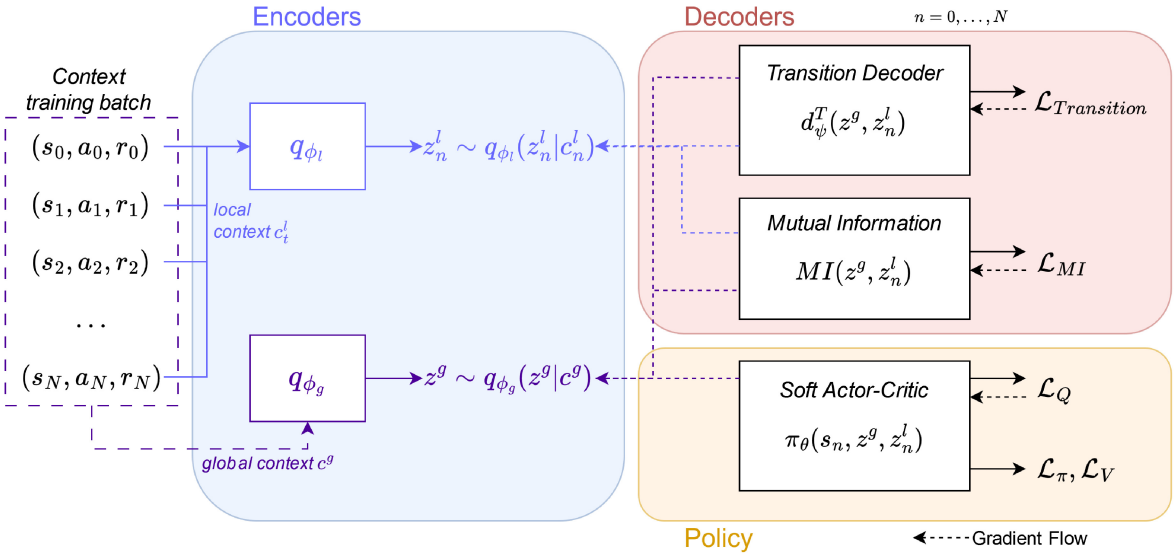

Fig. 1. *Architecture.* During meta-training, the global and local encoders, parameterized by q_{ϕ_g} and q_{ϕ_l}, respectively, are trained to decompose the context batch into separate embeddings. The posterior is trained using a transition decoder d_ψ^T for reconstruction error, along with gradients from training the RL policy π_θ. An MI constraint is included to reduce risk of posterior collapse. The dashed lines show gradient flows from each loss equation. Notably, only the global encoder takes gradients from the SAC policy.

more information prior to adaptation, or adapt to the task given the posterior. Architecturally, we design the global encoder network $q_{\phi_g}(z^g|c^g)$ with heavy inspiration from the encoder used in PEARL [8], which utilizes a permutation-invariant inference network, that is, the product of independent Gaussian factors: $q_{\phi_g}(z^g|c^g) = \prod_n \Psi_{\phi_g}(z|c_n^g)$, where $\Psi_{\phi_g}(z|c_n^g) \sim \mathcal{N}(f_\phi^{g,\mu}(c_n^g), f_\phi^{g,\sigma^2}(c_n^g))$ is calculated using a neural network f_ϕ^g that outputs parameters of the global Gaussian factor. The permutation-invariant nature of the network improves training stability by enabling the sampling of independent transitions during meta training, which breaks any trajectory-related dependencies. We also note that in our own experiments this outperformed a RNN global encoder, which aligns with the results found by Imagawa et al. [40]. Our global encoder learns task information implicitly by recreating the (soft) state-action value function under a maximum entropy framework $Q_\pi(s, a|z^g, z^l) = \log \pi(a|s, z^g, z^l) + V_\pi(s|z^g, z^l)$, as we found that training in this manner was more stable than recreating the reward or transition functions in a generative fashion. At test time, the global embedding is initially sampled from a $\mathcal{N}(0, I)$ unit Gaussian prior, which is used for exploration.

Local Embedding: Under the global–local decomposition framework, the probabilistic local embedding z^l should be able to extract auxiliary or temporal characteristics that a global embedding fixed for a trajectory may lack the expressiveness to sufficiently capture. In addition, the global and local embeddings in aggregate must have the information necessary to recreate the input *local context* at any episode step, $c_n^l = c_n = (s_n, a_n, r_n, s'_n)$. In GLOBEX we design our local encoder $q_{\phi_l}(z_n^l|c_n^l)$ as a simple multilayer perceptron (MLP) that, given the same context batch as used for global training $c^g = \{c_n^l\}_{1:N}$, learns to extract useful features from individual transitions. While we considered alternative encoder structures in Section V-D, we found that the MLP local encoder's stability and ease of training outweighed any potential benefits from a more complex architecture.

Training: In order to train the local (and global) encoders to maximize information retained, we also introduce a transition decoder $d_\psi^T(z^g, z_n^l) = \hat{c}_n^l$ that reconstructs the transition in the context batch $c_n^l = (s_n, a_n, r_n, s'_n)$ from the global and local embedding. Minimising the ensuing reconstruction loss performs this purpose, while the additional gradient flow from the policy ensures task information is learned

$$\mathcal{L}_{\text{transition}} = \sum_n^N \left(c_n^l - d_\psi^T(z^g, z_n^l)\right)^2. \tag{1}$$

During meta-training, the local encoder learns to extract z^l from the same context batch as z^g so that no useful information is lost from the embedding process, and optimizes the policy based on that embedding. However, at meta-test time z^l is generated from the *current roll-out*, unlike z^g which is conditioned on previous roll-outs. When implemented as such, we find that in practice z^l functions as a short-term memory component that enables the agent to exploit useful signals from recent transitions. As highlighted in Section V-A, making a time-variable z_n^l available to the policy in addition to the global z^g allows the agent to adjust to signals mid-roll-out, and augment the global embedding when it is not sufficient for optimal exploitative behavior.

When deciding on the prior for z^l, we would like to take advantage of the temporal nature of z^l and account for our current position in the trajectory. Therefore, in contrast to the static global prior we set our local prior $p(z^l)$ to be the *distribution of the previous step's local embedding*, $p(z_t^l) = q_{\phi_l}(z_{t-1}^l|c_{t-1}^l)$, where we use the subscript t to denote the temporal order of steps in a specific trajectory τ (as opposed to sampling from a random batch of N RL transitions).

Here, the local embedding becomes similar to a Bayesian filtering update [10], adding a minor temporal dependence on the previous embedding. Intuitively, constraining the local embedding at each time step toward its previous distribution provides stability and continuity to z_t^l over time, which leads to more stable roll-out paths.

As is standard in contextual meta-RL, we utilize variational inference to infer both global z^g and local latent z^l embeddings using the aforementioned networks $q_{\phi_g} = q_{\phi_g}(z^g|c^g)$ and $q_{\phi_l} = q_{\phi_l}(z^l|c^l)$. Using maximum likelihood interpretations of the Q-function loss ($\mathcal{L}_{Q_\pi} \approx -\log p(r|s,a,z^g,z^l)$) and transition reconstruction loss ($-\log p(\tau|z^g,z^l)$), we can formulate an objective function akin to a log-likelihood $F(M) = F(M|z^g,z^l) = \log p(r|s,a,z^g,z^l) + \log p(\tau|z^g,z^l)$ for task M, and assuming that z^g and z^l are sufficiently decoupled such that $p(\tau,z^g,z^l) \approx p(\tau|z^g,z^l)p(z^g)p(z^l)$, the ELBO can then be estimated as

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{p(M)}\Big[\mathbb{E}_{\substack{z^g\sim q_{\phi_g},\\ z^l\sim q_{\phi_l}}}\Big[&\log p\Big(r|s,a,z^g,z^l\Big)\\ &+\log p\Big(\tau|z^g,z^l\Big)\Big]\\ &-\beta_1 D_{KL}\Big(q_{\phi_g}(z^g|c^g)||p(z^g)\Big)\\ &-\beta_2 D_{KL}\Big(q_{\phi_l}(z^l|c^l)||p(z^l)\Big)\Big]\end{aligned} \tag{2}$$

where $\mathbb{E}_{p(M)}$ is the expectation over our distribution of tasks $M \sim p(M)$, which we calculate via Monte-Carlo sampling. Here, we use the notation $\bar{z}^l$ to indicate that gradients are not computed through the local embedding when calculating actor-critic or policy loss. Instead, it is trained purely via transition reconstruction and the Kullback–Leibler (KL) divergence. In our experiments, we found that only training the global embedding to recover state-action value improved training efficiency and stability, while the local embedding can still recover transition-specific variations.

MI Constraint: Once the global and local embeddings have been learned, it is typical in literature to utilize them directly or derive a weighted combination for downstream usage [56]. However, given the stochastic nature of these embeddings it is crucial to ensure that they are properly decoupled. This ensures that z^g and z^l both capture different aspects of task information, while also guarding against the risk of posterior collapse in either latent random variable. In order to achieve this, we utilize a regularizer that estimates and penalizes the MI between z^g and z^l. Finding the MI between two random variables is not trivial, and exact bounds can only be determined if either conditional density $p(z^g|z^l)$ or $p(z^l|z^g)$ is known, which cannot be calculated in this case. Instead, we utilize the vCLUB estimator [57], which consists of a network $g_\gamma(z^g|z^l)$, that is, trained to estimate the unknown conditional density $p(z^g|z^l)$ in parallel with the algorithm. This network can then be used to bound the MI between global–local embeddings by the equation

$$\mathcal{L}_{MI} = \mathbb{E}_{z^g,z^l}\Big[\log g_\gamma\Big(z^g,z^l\Big)\Big] - \mathbb{E}_{z^g}\mathbb{E}_{z^l}\Big[\log g_\gamma\Big(z^g,z^l\Big)\Big]. \tag{3}$$

In order to train g_γ, the network is updated by taking the log-likelihood of the context batch at each training step

$$\mathcal{L}_{\text{vCLUB}} = \sum_n^N \log g_\gamma\Big(z^g,z_n^l\Big) \tag{4}$$

which is then maximized via gradient descent. Apart from using the same context batch for convenience, the training of the vCLUB network is completely decoupled from the rest of the training process.

B. Implementation

In order to take advantage of the benefits of off-policy RL, we use soft actor critic (SAC) [58] as the inner RL algorithm. This allows the algorithm to take many gradient steps before needing to gather more data, resulting in a large increase in sample efficiency relative to methods that utilize on-policy inner-loop RL algorithms, such as PPO [59], [60]. Given an RL training batch sampled from the replay buffer $(s,a,r,s') \sim \mathcal{B}$ and global–local random variables $z^g \sim q_{\phi_g}(z^g|c^g)$ and $z^l \sim q_{\phi_l}(z^l|c^l)$, the loss functions for training SAC with global—local contextual embeddings are given below

$$\mathcal{L}_Q = \mathbb{E}\Big[\Big(Q_\theta(s,a,z^g,\bar{z}^l) - r - \overline{V}(s',\bar{z}^g,\bar{z}^l)\Big)^2\Big] \tag{5}$$

$$\mathcal{L}_\pi = \mathbb{E}\Big[D_{\text{KL}}\Big(\pi_\theta\Big(a|s,\bar{z}^g,\bar{z}^l\Big)\Big\|\frac{e^{Q_\theta(s,a,\bar{z}^g,\bar{z}^l)}}{\mathcal{Z}_\theta(s)}\Big)\Big] \tag{6}$$

$$\begin{aligned}\mathcal{L}_V = \mathbb{E}\Big[\Big(&V_\theta(s,\bar{z}^g,\bar{z}^l) - Q_\theta(s,a,\bar{z}^g,\bar{z}^l)\\ &+\log\pi_\theta(a|s,\bar{z}^g,\bar{z}^l)\Big)^2\Big]\end{aligned} \tag{7}$$

where $\overline{V}$ is a target function, $\mathcal{Z}_\theta$ is a normalization function that does not affect gradients, and we again use $\bar{z}^g,\bar{z}^l$ to denote when gradients are not being passed through during training.

In order to offset concerns with distributional mismatch when training the contextual encoders and RL policy in an off-policy manner, we utilize the same dual sampling scheme as PEARL [8]. This involves sampling encoder training data from a separate buffer that retains only recently collected data (e.g., from the most recent training epoch), while the RL batch is allowed to sample from a larger replay buffer to build the actor-critic value function.

During roll-out, it is not possible to calculate $z^g \sim q_{\phi_g}(z^g|c^g)$ and $z_t^l \sim q_{\phi_l}(z_t^l|c_t^l)$ simultaneously, as is done during training. Instead, during roll-out and meta-test adaptation global context is collected from previous episodes (including initial exploratory episodes), and local context uses information collected from the current episode. Fig. 2 illustrates how the global and local embeddings are used in this process, and pseudocode for test-time adaptation can be found in Algorithm 2.

V. Experiments

This section describes experiments that aim to answer the following questions: 1) What exploratory and exploitative behavior does the inclusion of a local embedding in addition to the global embedding drive? 2) Does GLOBEX outperform

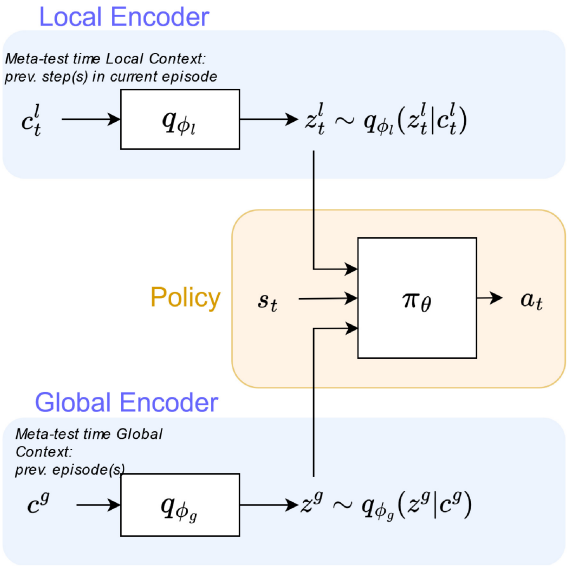

Fig. 2. *Roll-out.* The local and global encoders use different contextual inputs during roll-out and when gathering data, which are then utilized by the RL policy. The subscript t is utilized to clearly denote the relationship between local context and the action determined by the RL policy.

Algorithm 2 GLOBEX Meta-Test

Input: Test task M from $p(M)$
Initialise context $c = \{\}$
Sample $z^g \sim p(z^g)$ from prior
for ep in adaptation_episodes **do**
  Sample $z_0^l \sim p(z_t^l)$ from prior
  for $t = 0, \ldots, T$ **do**
    Choose action from $\pi_\theta(a_t|s_t, z^g, z_t^l)$ and step
    Add collected context $c_{t+1} = (s_t, a_t, r_t, s_{t+1})$ to c
    Update and sample $z_{t+1}^l \sim q_{\phi_l}(z_{t+1}^l|c_{t+1})$
  end for
  Update and sample $z^g \sim q_{\phi_g}(z^g|c)$
end for

peer methods that utilize a single encoder on standard benchmarks? 3) Can a global–local embedding be applied to new methodologies with many complex components to improve performance in difficult environments? and 4) What is the impact of certain design choices of GLOBEX?

A. Behavioral Impact of Local Embedding

Experiment Setup: To address the first question, we begin with a visual demonstration of the performance of GLOBEX in the simple PointRobot environment, which features randomly sampled goal points on a 2-D unit circle around the agent, with well-shaped rewards. We consider this to be a simple environment, and are primarily interested in how effectively the algorithm exploits basic environmental signals and maintains adaptation consistency. We compare GLOBEX to PEARL [8], which features a very similar global encoder structure without the local component. In the experiment, we first run two exploratory episodes (with $z^g \sim N(0, I)$), then display 10 adaptation trajectories per task in order to understand the distribution of trajectories given z sampled from the posterior.

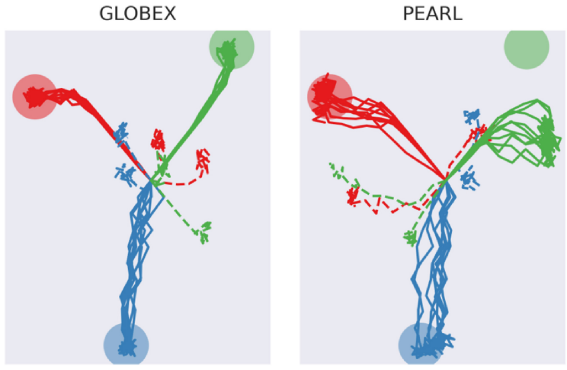

Fig. 3. *2-D Navigation.* Dashed lines indicate exploratory trajectories using a task variable sampled from the prior, while full lines indicate adaptation trajectories using the task posterior. Our approach, GLOBEX, adapts more accurately and with lower variance in path distribution.

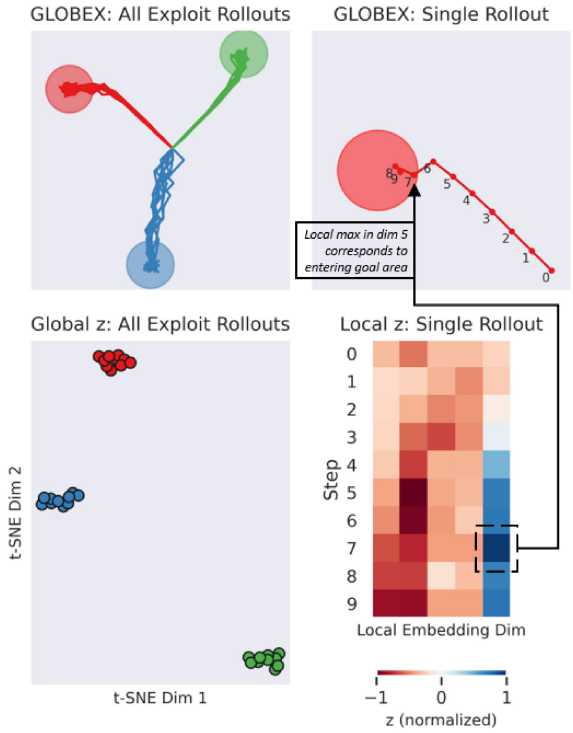

Fig. 4. *Global–local embeddings.* For a group of GLOBEX rollouts (top left), the t-SNE visualization of the global z values that generated them is provided (bottom left). Then, focusing on a single rollout (top right), the local z embeddings at each timestep are provided (bottom right).

Results: In Fig. 3, we see that with just a couple of exploratory episodes as context, GLOBEX clearly finds the goal with a high level of accuracy and consistency in path trajectory. In comparison, PEARL is less accurate (notably missing the top-right green goal) and has a wider range of potential paths, which means that adaptation speed and accuracy is more dependent on the randomly sampled z. We hypothesize that the local embedding of GLOBEX has learned to pick up on the clear signal that the well-shaped reward provides, while PEARL's adaptation procedure lacks the temporal expressiveness needed to accurately exploit the same signal, which highlights the effectiveness of a well-trained local embedding.

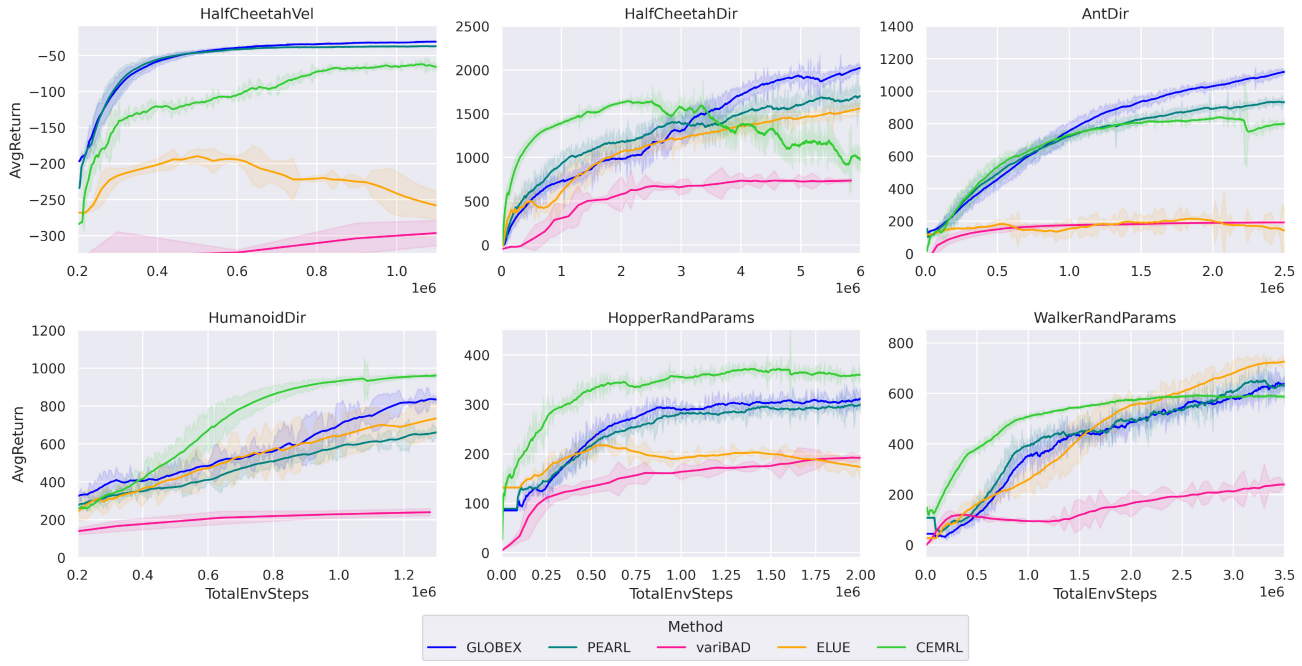

Fig. 5. *MuJoCo continuous control.* Test-task performance versus samples collected during meta-training. GLOBEX is competitive with peer off-policy algorithms across multiple environments, due to the policy being able to utilize more information from its global and local embeddings. The on-policy variBAD comparison clearly is less sample efficient than off-policy methods.

Further analysis and visualization of the learned global–local embeddings can be found in Fig. 4. First, we see that the global posterior after adaptation ($z^g \sim q_{\phi_g}(z^g|c)$) is clearly clustered and distinct for different tasks, indicating that it is effectively reasoning about the true task identity. Then, when considering the values of the local embedding for each timestep z_t^l we can see that the last dimension takes increasingly high values as the agent gets closer to the goal, suggesting that it also strongly impacts behavior and may have embedded some knowledge of where the goal is. Additional analysis, available in the supplementary document, also indicates that the first dimension fulfils a complementary purpose.

B. Performance and Sample Efficiency

Experiment Setup: To evaluate the performance of GLOBEX on more complex meta-RL environment tasks, we consider 6 continuous control environments from the MuJoCo suite [15]. These environments have well-shaped rewards, but are challenging in that there are either major variations in the goal and reward function (*HalfCheetahVel*, *HalfCheetahDir*, *AntDir*, and *HumanoidDir*), or there are major variations in environment and transition dynamics (*WalkerRandParams*, *HopperRandParams*). We compare GLOBEX with various peer and classical meta-RL techniques: PEARL, variBAD [10], which is an on-policy contextual meta-RL algorithm, ELUE [40] which utilizes belief embeddings like variBAD but in an off-policy fashion, and CEMRL [32], which utilizes a GMM-based encoder to achieve a hierarchical embedding structure. We use the garage package [61] implementations for PEARL and MAML, and original source code for variBAD, ELUE and CEMRL. In all instances, we used two exploratory episodes before measuring the performance of a single adaptation episode and average performance across five seeds.

Results: Results are shown in Fig. 5. In general, we see that GLOBEX achieves a greater final performance compared to peer algorithms in half of the environments tested (*HalfCheetahVel*, *HalfCheetahDir*, and *AntDir*). CEMRL (*HumanoidDir* and *HopperRandParams*) and ELUE (*WalkerRandParams*) both outperform GLOBEX in certain environments but show some performance instability in others: CEMRL trains remarkably quickly but often reaches failure states, possibly due to difficulties in fitting task clusters. On the other hand, ELUE fails to solve some tasks well (e.g., *HalfCheetahVel*). In Section V-D, we show that utilizing a belief-based approach for GLOBEX can decrease performance; we suspect that belief-conditioned policies [10], [53] are more suited toward on-policy training rather than off-policy meta-RL.

When compared to PEARL, which can be considered to be relatively equivalent to GLOBEX without a local embedding, GLOBEX outperforms in all environments with the exception of *WalkerRandParams*. In addition, one-sided paired t-tests between GLOBEX and PEARL results show $p < 0.001$ for all environments except for *WalkerRandParams*, where p indicates no significant difference in performance between the two methods. Finally, all off-policy algorithms exhibit greater sample efficiency compared to the on-policy variBAD - while this may not necessarily translate to a significant difference in wall-clock training time for simulated environments, it is a clear advantage when considering real-world applications that may not have the luxury of being able to gather hundreds of millions of samples.

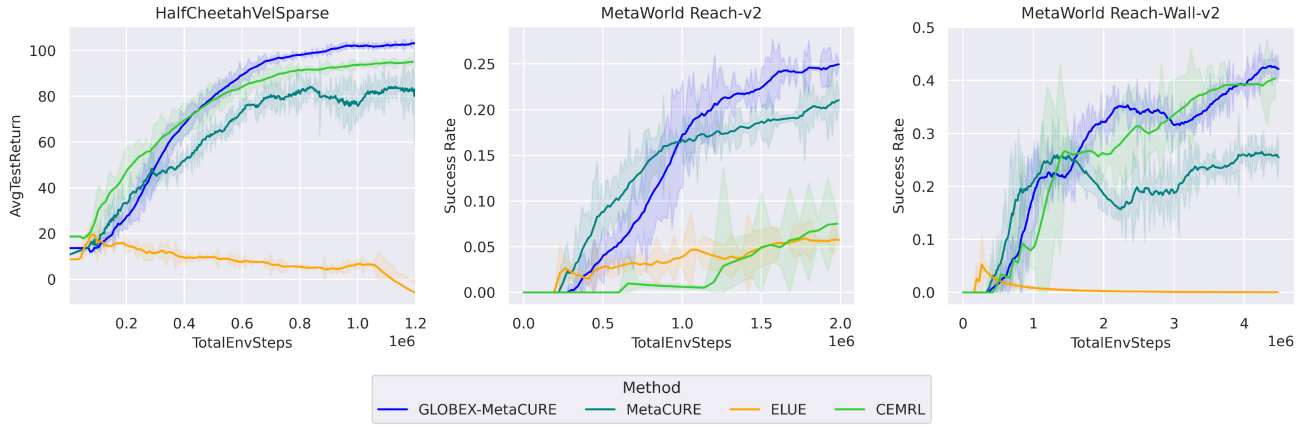

Fig. 6. *GLOBEX-MetaCURE Results:* Enhancing other contextual meta-RL methods with global–local decomposition of context enables the solution of difficult tasks across sparse MuJoCo and complex MetaWorld benchmarks. Comparisons to peer off-policy methods show how difficult the tasks are to consistently solve.

C. Global–Local Decomposition With Exploration

Experiment Setup: MetaCURE utilizes an additional exploratory policy π_e to maximize information gained from initial task exploration, in order to facilitate improved adaptation. As the contextual encoder used by MetaCURE takes prior gathered context as an input, and outputs a task variable $z \sim q_\phi(z|c)$, it can be considered to be in the same class of contextual meta-RL methods as PEARL and GLOBEX, and potentially stands to benefit from the global–local decomposition of context used in the latter. In this experiment, we implement a global–local encoder structure based on GLOBEX within the MetaCURE algorithm and denote this augmented MetaCURE with global–local decomposition as "GLOBEX-MetaCURE." We then compare the two methods on a sparse MuJoCo environment (*HalfCheetahVelSparse*) and two MetaWorld environments (*Reach-v2* and *Reach-Wall-v2*) [5], which are difficult continuous manipulation tasks that requires the agent to control a simulated Sawyer arm. In these experiments, we utilized experiment parameters from the public implementation of MetaCURE, and provide comparisons with the same peer off-policy meta-RL methods ELUE and CEMRL.

Results: Results are shown in Fig. 6. It can be seen that the global–local decomposition of context also improves the embeddings used by MetaCURE, with the augmented GLOBEX-MetaCURE demonstrating improved results across complex environments, while other methods either struggle to learn a consistent policy to achieve success (CEMRL), or often fail to solve the environment entirely (ELUE). This indicates that the ideas shared in GLOBEX and this article are generally applicable to any contextual meta-RL algorithm that learns a single task representation.

D. Ablations

In this section, we run ablations that target key features of GLOBEX and demonstrate the impact of our design choices. These ablations will focus primarily on choices revolving around the local embedding z^l and its joint-optimization via MI, as it is the primary contribution of GLOBEX.

Impact of MI Constraint: While the reasoning behind the inclusion of an estimated MI loss module was discussed earlier, Fig. 7 analyses the sensitivity of the MI loss constraint on overall performance in *HalfCheetahDir*, where we see that the optimal MI coefficient was 0.05. Although some degree of parameter tuning per environment may be useful to maximize performance, we found that typically any coefficient of 0.1 or lower resulted in good and stable results. We note that in other environments the unconstrained (*MI0*) treatment may outperform some higher-MI coefficients, likely as the learned global–local decomposition is naturally decoupled.

Local Encoder Network Architecture: In our next experiment, we will evaluate our choice of network architecture for $q_{\phi_l}(z_t^l|c_t^l)$. In Section IV-A, we argued that a simple MLP was sufficient for the local encoder, as time-invariant global information would flow instead to the global embedding. Results are shown in Fig. 8, where we see that a simple MLP encoder ("MLP (ours)") outperforms a more complex RNN-based encoder ("RNN"), despite the RNN having the capacity to take in and remember more past transitions from the current episode. In order to implement the RNN encoder, we sample context batches as full trajectories rather than independent transitions, which also impacts training. We believe the difference in performance seen likely stems from the ability to break path dependencies with an encoder suited to training on independent transitions, as opposed to requiring full trajectories.

Choice of Local Encoder Prior: Finally, we investigate the impact of different choices of prior for z_t^l. In Fig. 9, we see that our choice of prior ("Previous step") outperforms a fixed $N(0, I)$ prior both in terms of training efficiency and final performance. We believe the variable prior offers greater flexibility in learning an effective local embedding, primarily impacting training stability and speed. In addition, we found in many instances with a fixed prior the local encoder was guided too heavily to regress toward the prior, ending up with a completely uninformative local embedding that did not assist the agent in the adaptation process.

Embedding Sampling: GLOBEX samples the values of its global latent variable z^g and local latent variable z^l when determining the action to take during adaptation. However,

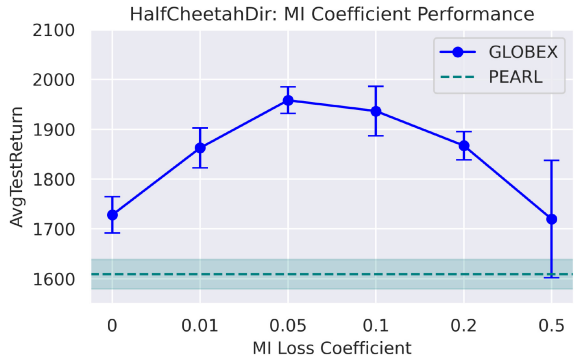

Fig. 7. *MI ablation.* GLOBEX uses an MI regularizer to ensure against posterior collapse in global and local embeddings, which can improve overall performance.

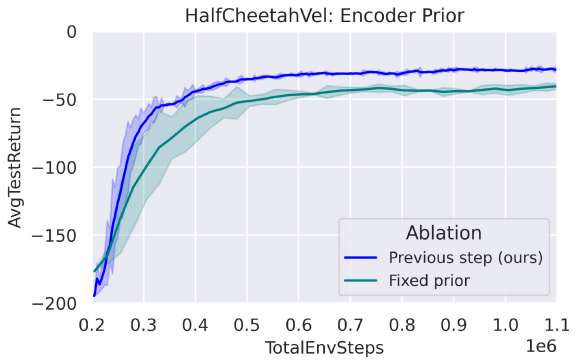

Fig. 9. *Local encoder prior ablation.* GLOBEX sets the prior of the local latent variable to be its value in the previous step. This outperforms a basic $N(0, I)$ prior, likely due to the additional flexibility provided.

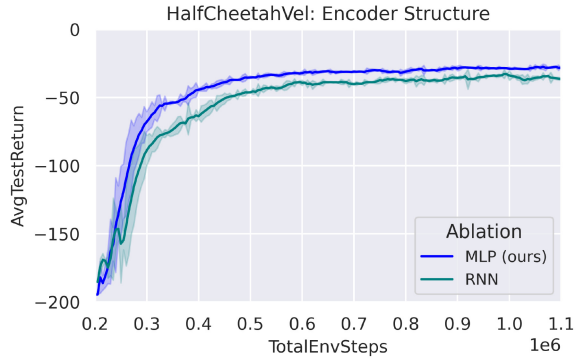

Fig. 8. *Local Encoder structure.* Using a simple MLP for the local embedding enables a sampling method that samples independent tuples from the replay buffer, instead of training over whole trajectories.

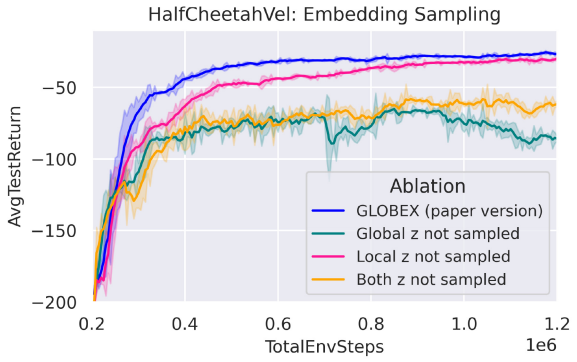

Fig. 10. *Embedding sampling.* Sampling embeddings at runtime outperforms relative to providing the distribution to the agent, as the embeddings are trained to recover maximal information from the input context.

it is also possible to instead pass distributional parameters to the RL policy instead [10]. When done effectively, this may allow the agent to more effectively reason about task uncertainty, as the variance of the latent distribution is known by the policy. However, GLOBEX is uniquely designed to utilize the actual sampled latent variables due to the presence of transition reconstruction loss, $\mathcal{L}_{\text{transition}}$. Minimising that loss term ensures that a sampled z^g and z^l maximizes task information retained, a property that may not be present when utilizing distribution parameters. In Fig. 10, we compare different versions of GLOBEX.

1) Sampling both z^g and z^l ("GLOBEX").
2) Sampling z^l only, while passing the mean and variance of z^g to the policy ("Global z not sampled").
3) Sampling z^g only, while passing the mean and variance of z^l to the policy ("Local z not sampled").
4) Sampling neither ("Both z not sampled").

It is clear that not sampling z^g degrades performance, as sampling a fixed z^g provides a consistent task hypothesis, that is, used for effective exploitation at test-time, which may not be present when only inputting distribution parameters. In addition, not sampling z^l leads to slower training efficiency, as the sampled z^l is directly optimized to maximize information retention during training, and is therefore more useful to the policy when adapting to unknown environments.

VI. Conclusion and Future Work

In this article, we introduce GLOBEX, an off-policy contextual meta-RL algorithm that utilizes a probabilistic global–local decomposition of context gathered in a new environment to decouple task information into static global components, and local temporal variations. We develop separate training approaches for each embedding, leveraging both shared and embedding-specific losses guided by an MI constraint newly tailored to meta-RL. Finally, we identify that GLOBEX and its derivatives outperform peer methods in terms of training efficiency and few-shot adaptation across a wide variety of simple and complex environments.

There is also opportunity for further work on this topic. First, GLOBEX primarily relies on the principled yet simple adaptation strategy of Thompson sampling for exploration. Since Section V-C shows that global–local decomposition aids exploration, there is room to devise embedding-specific exploratory incentives. This may also enable policies that adapt to reward and dynamics changes in real-time, enabling continual meta-RL applications, as discussed in [32] and [38].

In addition, it may be of interest to investigate how a global–local structure can be adapted toward works that explore alternative approaches toward representation learning within meta-RL, such as attention-based encoders [62], [63], diffusion models [64], or offline learning in general [44], [45]. Finally, GLOBEX's performance degrades when test-time

tasks differ significantly from those encountered during training (that is, out-of-distribution tasks), which is a common challenge with existing meta-RL. Each point mentioned provides promising directions for future work.

Ultimately, we believe that the techniques used in GLOBEX show that there is great potential in applying additional structure to the contextual embedding so important to contextual meta-RL, and that further enriching the quality of these representations will lead to improved task inference and performance moving forward.

REFERENCES

[1] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman, "Quantifying generalization in reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1282–1289.

[2] S. Mohanty et al., "Measuring sample efficiency and generalization in reinforcement learning benchmarks: NeurIPS 2020 Procgen benchmark," in *Proc. Compet. Demonstr. Track (NeurIPS)*, 2021, pp. 361–395.

[3] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.

[4] A. Nagabandi et al., "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.

[5] T. Yu et al., "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Proc. Conf. Robot Learn.*, 2020, pp. 1094–1100.

[6] S. Belkhale, R. Li, G. Kahn, R. McAllister, R. Calandra, and S. Levine, "Model-based meta-reinforcement learning for flight with suspended payloads," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1471–1478, Apr. 2021.

[7] J. Beck et al., "A survey of meta-reinforcement learning," 2024, *arXiv:2301.08028*.

[8] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen, "Efficient off-policy meta-reinforcement learning via probabilistic context variables," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 5331–5340.

[9] E. Z. Liu, A. Raghunathan, P. Liang, and C. Finn, "Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 6925–6935.

[10] L. Zintgraf et al., "VariBAD: Variational Bayes-adaptive deep RL via Meta-learning," *J. Mach. Learn. Res.*, vol. 22, no. 289, pp. 1–39, 2021.

[11] S. Tonekaboni, C.-L. Li, S. O. Arik, A. Goldenberg, and T. Pfister, "Decoupling local and global representations of time series," in *Proc. 25th Int. Conf. Artif. Intell. Statist.*, 2022, pp. 8700–8714.

[12] Q. Wang, W. Huang, X. Zhang, and X. Li, "GLCM: Global–local captioning model for remote sensing image captioning," *IEEE Trans. Cybern.*, vol. 53, no. 11, pp. 6910–6922, Nov. 2023.

[13] H. Liu, W. Zhang, F. Liu, H. Wu, and L. Shen, "Fingerprint presentation attack detector using global–local model," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12315–12328, Nov. 2022.

[14] X. Ma, X. Kong, S. Zhang, and E. H. Hovy, "Decoupling global and local representations via invertible generative flows," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–23.

[15] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.

[16] J. Zhang et al., "MetaCURE: Meta reinforcement learning with empowerment-driven exploration," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 12600–12610.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic Meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[18] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*.

[19] Z. Xu, H. P. van Hasselt, and D. Silver, "Meta-gradient reinforcement learning," in *Proc. 32nd Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–12.

[20] J. Rothfuss, D. Lee, I. Clavera, T. Asfour, and P. Abbeel, "ProMP: Proximal meta-policy search," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–25.

[21] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine, "Meta-learning with implicit gradients," in *Proc. 33rd Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–12.

[22] Z. Xu, X. Chen, and L. Cao, "Fast task adaptation based on the combination of model-based and gradient-based meta learning," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 5209–5218, Jun. 2022.

[23] C. Finn and S. Levine, "Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–20.

[24] Z. Xiong, L. Zintgraf, J. Beck, R. Vuorio, and S. Whiteson, "On the practical consistency of meta-reinforcement learning algorithms," 2021, *arXiv:2112.00478*.

[25] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "RL2: Fast reinforcement learning via slow reinforcement learning," 2016, *arXiv:1611.02779*.

[26] J. X. Wang et al., "Learning to reinforcement learn," in *Proc. CogSci*, 2017, pp. 1–17.

[27] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–17.

[28] B. C. Stadie et al., "Some considerations on learning to explore via meta-reinforcement learning," 2019, *arXiv:1803.01118*.

[29] E. Parisotto, S. Ghosh, S. B. Yalamanchi, V. Chinnaobireddy, Y. Wu, and R. Salakhutdinov, "Concurrent meta reinforcement learning," 2019, *arXiv:1903.02710*.

[30] Z. Mandi, P. Abbeel, and S. James, "On the effectiveness of fine-tuning versus Meta-reinforcement learning," in *Proc. 36th Adv. Neural Inf. Process. Syst.*, 2022, pp. 26519–26531.

[31] T. Z. Zhao, A. Nagabandi, K. Rakelly, C. Finn, and S. Levine, "MELD: Meta-reinforcement learning from images via latent state models," in *Proc. Conf. Robot Learn.*, 2021, pp. 1246–1261.

[32] Z. Bing, D. Lerch, K. Huang, and A. Knoll, "Meta-reinforcement learning in non-stationary and dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3476–3491, Mar. 2023.

[33] H. Wang, Z. Liu, Z. Han, Y. Wu, and D. Liu, "Rapid adaptation for active pantograph control in high-speed railway via deep meta reinforcement learning," *IEEE Trans. Cybern.*, vol. 54, no. 5, pp. 2811–2823, May 2024.

[34] A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine, "Meta-reinforcement learning of structured exploration strategies," in *Proc. 32nd Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–10.

[35] H. Fu et al., "Towards effective context for Meta-reinforcement learning: An approach based on contrastive learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 7457–7465.

[36] M. Wang, X. Li, L. Zhang, and M. Wang, "MetaCARD: Meta-reinforcement learning with task uncertainty feedback via decoupled context-aware reward and dynamics components," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 15555–15562.

[37] J. Humplik, A. Galashov, L. Hasenclever, P. A. Ortega, Y. W. Teh, and N. Heess, "Meta reinforcement learning as task inference," 2019, *arXiv:1905.06424*.

[38] F.-M. Luo, S. Jiang, Y. Yu, Z. Zhang, and Y.-F. Zhang, "Adapt to environment sudden changes by learning a context sensitive policy," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7637–7646.

[39] L. Zintgraf et al., "Exploration in approximate hyper-state space for meta reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12991–13001.

[40] T. Imagawa, T. Hiraoka, and Y. Tsuruoka, "Off-policy meta-reinforcement learning with belief-based task inference," *IEEE Access*, vol. 10, pp. 49494–49507, 2022.

[41] A. Ajay, A. Gupta, D. Ghosh, S. Levine, and P. Agrawal, "Distributionally adaptive meta reinforcement learning," in *Proc. 36th Adv. Neural Inf. Process. Syst.*, 2022, pp. 25856–25869.

[42] M. Pislar, D. Szepesvari, G. Ostrovski, D. Borsa, and T. Schaul, "When should agents explore?" in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–23.

[43] J. Kim, J. Xuan, C. Liang, and F. Hussain, "An autonomous non-monolithic agent with multi-mode exploration based on options framework," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2023, pp. 1–8.

[44] L. Li, R. Yang, and D. Luo, "FOCAL: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–23.

[45] V. H. Pong, A. V. Nair, L. M. Smith, C. Huang, and S. Levine, "Offline meta-reinforcement learning with online self-supervision," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 17811–17829.

[46] H. Yuan and Z. Lu, "Robust task representations for offline meta-reinforcement learning via contrastive learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 1–13.

[47] Y. Gao et al., "Context shift reduction for offline meta-reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 1–12.
[48] H. Yang, K. Lin, T. Yang, and G. Sun, "SCORE: Simple contrastive representation and reset-ensemble for offline meta-reinforcement learning," *Knowl.-Based Syst.*, vol. 309, Jan. 2025, Art. no. 112767.
[49] Y. Mu et al., "DOMINO: Decomposed mutual information optimization for generalized context in Meta-reinforcement learning," in *Proc. 36th Adv. Neural Inf. Process. Syst.*, 2022, pp. 27563–27575.
[50] L. Wen, E. H. Tseng, H. Peng, and S. Zhang, "Dream to adapt: Meta reinforcement learning by latent context imagination and MDP imagination," *IEEE Robot. Autom. Lett.*, vol. 9, no. 11, pp. 9701–9708, Nov. 2024.
[51] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Estimating disentangled belief about hidden state and hidden task for meta-reinforcement learning," in *Proc. 3rd Conf. Learn. Dyn. Control*, 2021, pp. 73–86.
[52] M. Wang et al., "Meta-reinforcement learning based on self-supervised task representation learning," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 10157–10165.
[53] S. Lee, M. Cho, and Y. Sung, "Parameterizing non-parametric meta-reinforcement learning tasks via subtask decomposition," in *Proc. 37th Adv. Neural Inf. Process. Syst.*, 2023, pp. 1–28.
[54] Z. Chu, R. Cai, and H. Wang, "meta-reinforcement learning via exploratory task clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 11633–11641.
[55] Z. Bing, Y. Yun, K. Huang, and A. Knoll, "Context-based Meta-reinforcement learning with Bayesian nonparametric models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 10, pp. 6948–6965, Oct. 2024.
[56] S. Yang, K. Yu, F. Cao, H. Wang, and X. Wu, "Dual-representation-based autoencoder for domain adaptation," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7464–7477, Aug. 2022.
[57] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, "Club: A contrastive log-ratio upper bound of mutual information," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1779–1788.
[58] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.
[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[60] H. Xu, Z. Yan, J. Xuan, G. Zhang, and J. Lu, "Improving proximal policy optimization with alpha divergence," *Neurocomputing*, vol. 534, pp. 94–105, May 2023.
[61] "Garage: A toolkit for reproducible reinforcement learning research." 2019. [Online]. Available: https://github.com/rlworkgroup/garage
[62] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked convolutional denoising auto-encoders for feature representation," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1017–1027, Apr. 2017.
[63] M. Xu et al., "Prompting decision transformer for few-shot policy generalization," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 1–15.
[64] F. Ni et al., "MetaDiffuser: Diffusion model as conditional planner for offline meta-RL," in *Proc. 40th Int. Conf. Mach. Learn.*, 2023, pp. 26087–26105.

Nelson Ma received the B.S. degree [with Hons. Class I (Statistics)] in advanced mathematics from the University of Sydney, Camperdown, NSW, Australia, in 2018. He is currently pursuing the Ph.D. degree with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia.

His research interests include machine learning, representation learning, and reinforcement learning.

Junyu Xuan (Senior Member, IEEE) received the Dual Doctoral degrees from the University of Technology Sydney, Ultimo, NSW, Australia, and Shanghai University, Shanghai, China, in 2016.

He is currently a Senior Lecturer with the Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW, Australia. He has authored or co-authored more than 40 articles in high-quality journals and conferences, including *Artificial Intelligence*, *Machine Learning*, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the *ACM Computing Surveys*, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *ACM Transactions on Information Systems*, and IEEE TRANSACTIONS ON CYBERNETICS. His main research interests include Bayesian nonparametric learning, text mining, and Web mining.

Guangquan Zhang received the Ph.D. degree in applied mathematics from Curtin University, Bentley, WA, Australia, in 2001.

He is an Australian Research Council (ARC) QEII Fellow, an Associate Professor, and the Director of the Decision Systems and e-Service Intelligent Research Laboratory, Australian Artificial Intelligence Institute, University of Technology Sydney, Ultimo, NSW, Australia. From 1993 to 1997, he was a Full Professor with the Department of Mathematics, Hebei University, Baoding, China. He has published six authored monographs and over 500 papers. His research has been widely applied in industries.

Dr. Zhang has won ten very competitive ARC Discovery grants and many other research projects.

Jie Lu (Fellow, IEEE) received the Ph.D. degree from Curtin University, Bentley, WA, Australia, in 2000.

She is an Australian Laureate Fellow, an IFSA Fellow, an ACS Fellow, a Distinguished Professor, and the Director of Australian Artificial Intelligence Institute, University of Technology Sydney, Ultimo, NSW, Australia. She has published over 500 papers in IEEE TRANSACTIONS and other leading journals and conferences. Many of her research results have been successfully transferred to industry. Her main research expertise is in transfer learning, concept drift, fuzzy systems, decision support systems, and recommender systems.

Dr. Lu has been awarded 12 Australian Research Council discovery and linkage grants and led 20 industry projects. She is the recipient of two IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Awards (2019 and 2022), the NeurIPS2022 Outstanding Paper Award, the Australia's Most Innovative Engineer Award (2019), the Australasian Artificial Intelligence Distinguished Research Contribution Award (2022), the Australian NSW Premier's Prize on Excellence in Engineering or Information and Communication Technology (2023), and the Officer of the Order of Australia (AO) 2023. She also serves as the Editor-in-Chief for *Knowledge-Based Systems*.