Improved Context-Based Offline Meta-RL with Attention and Contrastive Learning

Lanqing Li [*1] **Yuanhao Huang** [*2] **Dijun Luo** [1]

Abstract

Meta-learning for offline reinforcement learning (OMRL) is an understudied problem with tremendous potential impact by enabling RL algorithms in many real-world applications. A popular solution to the problem is to infer task identity as augmented state using a context-based encoder, for which efficient learning of task representations remains an open challenge. In this work, we improve upon one of the SOTA OMRL algorithms, FOCAL, by incorporating intra-task attention mechanism and inter-task contrastive learning objectives for more effective task inference and learning of control. Theoretical analysis and experiments are presented to demonstrate the superior performance, efficiency and robustness of our end-to-end and model-free method compared to prior algorithms across multiple meta-RL benchmarks.[1]

1. Introduction

Deep reinforcement learning (RL) has achieved many successes with human- or superhuman-level performance across a wide range of complex domains (Mnih et al., 2015; Silver et al., 2017; Vinyals et al., 2019; Ye et al., 2020). However, all these major breakthroughs focus on finding the best-performing strategy by trial-and-error interactions with a single environment, which poses severe concern for scenarios such as healthcare (Gottesman et al., 2019) and autonomous driving (Shalev-Shwartz et al., 2016) where safety is paramount. Moreover, these RL algorithms require tremendous explorations and training samples, and also tend to overfit to the target task (Whiteson et al., 2011; Song et al., 2019), resulting in poor generalization and robustness. To make RL truly practical in many real-world applications, a new paradigm with better safety, sample efficiency and generalization is in need.

Offline meta-RL, as a marriage between offline/batch RL and meta-RL, has emerged as a promising candidate to the aforementioned challenges. Like supervised learning, offline RL restricts the agent to solely learn from fixed data, circumventing potentially risky explorations. Additionally, offline algorithms are by nature off-policy, which by incorporating and reusing diverse prior experience, have proven to achieve far better sample efficiency than on-policy counterparts (Haarnoja et al., 2018). However, such desired properties come at a cost. In order to learn without trial-and-error feedback, offline RL requires careful constraints on the learner to prevent divergence due to out-of-distribution state-actions, known as bootstrapping errors (Kumar et al., 2019; Levine et al., 2020).

Meta-RL on the other hand, trains the agent across a distribution of tasks. It is desired that such an agent also performs well on out-of-distribution data and achieve fast adaptation and generalization to unseen tasks from a similar distribution. This can be realized by learning a single universal policy conditioned on a latent task representation, known as context-based method (Rakelly et al., 2019; Khetarpal et al., 2020). Alternatively, the shared skills can be learned with a meta-controller (Oh et al., 2017).

In this work we restrict our attention on context-based offline meta-RL, which remains an understudied framework with a few existing algorithms (Li et al., 2019a; 2020; Dorfman & Tamar, 2020), for tasks that differ in reward or transition functions. One major challenge associated with this particular scenario is termed Markov Decision Process (MDP) ambiguity or mis-identification, namely the task-conditioned policies spuriously correlate with state-action pairs due to biased distribution of fixed datasets. This phenomenon can be interpreted as a special form of memorization problem in classical meta-learning (Yin et al., 2019), where the value and policy functions overfit the training set distributions without capturing the task identity from reward and transition functions, often leading to degenerate latent representations (Li et al., 2020) and poor generalization. To alleviate such overfitting, Li et al. (2020) proposes

[*]Equal contribution [1]Tencent AI Lab, Shenzhen, China [2]Department of Computer Science, ETH Zürich, Zürich, Switzerland. Correspondence to: Lanqing Li <lanqingli@tencent.com>, Dijun Luo <dijunluo@tencent.com>.

[1]Link to the source code and details of experiments will be covered in Appendix and supplmentary materials.

a framework which decouples the learning of task representations from control by using self-supervised distance metric learning. *However, under extreme scenarios such as sparse reward, where a considerable portion of aggregated experience provides little information regarding task identity, efficient and effective learning of task representation is still challenging.*

To address the aforementioned problem, in this paper we propose intra-task attention mechanism and inter-task contrastive learning objectives to achieve more effective task inference. More specifically, for each task, we apply batch-wise channel attention to assign inhomogeneous weights to transition samples, and use sequence-wise self-attention (Vaswani et al., 2017) to better capture the correlation within transition dimensions. Additionally, we implemented a matrix-form variant of the SOTA Momentum Contrast (MoCo) (He et al., 2020) on task-level representation learning, by replacing its dictionary queue with a meta-batch sampled on-the-fly. We evaluate our method on a set of continuous control tasks (with varying rewards/dynamics) in MuJoCo (Todorov et al., 2012) as well as a variety of 2D-navigation problems introduced by Rakelly et al. (2019). Experiments show that the specific design choices of attention and contrastive learning mechanisms not only boost the performance and efficiency of task inference in SOTA context-based OMRL algorithms like FOCAL (Li et al., 2020), but also significantly improve its robustness against sparse reward and distribution shift. We name our new method FOCAL++ in this respect.

2. Related Work

Attention in RL Although attention mechanism has proven a powerful tool across of a broad spectrum of problems (Mnih et al., 2014; Wang & Shen, 2017; Veličković et al., 2017; Devlin et al., 2018), to our best knowledge, its applications to RL domain remain relatively understudied. Mishra et al. (2017) experimented with a combination of temporal convolution and soft attention in meta-learner architectures for RL, but found that simple application of SOTA attention modules such as transformers is hard to optimize and could not solve even simple bandit tasks and tabular MDPs. A success in training stability of attention in RL was later reported by reordering of the layer normalization coupled with the addition of gating mechanism to key points in the submodules of the transformer (Parisotto et al., 2020), which is further improved with adaptive attention span (Sukhbaatar et al., 2019) in (Kumar et al., 2020). However, all these works focus on applying attention along the temporal dimension, in order to capture the time-dependent correlation in MDPs or POMDPs. So far, we found no related work with clear motivation to use attention mechanism for learning better representations in classical RL, not to mention the mutli-task setting.

The closest work we found by far (Barati & Chen, 2019) employs attention in multi-view/multi-agent RL, to learn different weights on various workers or agents, aggregated by a global network to form a centralized policy. Analogous to our proposal, such architecture has the advantage of accounting for different importance of each input in the decision making process, and makes the global agent robust to noise and partial observability.

Contrastive Learning Contrastive learning (Chopra et al., 2005; Hadsell et al., 2006) has emerged as a powerful framework for self-supervised and unsupervised representation learning. In essence, it aims to capture feature similarity by learning to distinguish between semantically similar and dissimilar samples with theoretical guarantees (Arora et al., 2019). Recent progress in contrastive learning focuses mostly on learning visual representations as 'pretext' tasks (He et al., 2020). MoCo (He et al., 2020) formulates contrastive learning as dictionary look-up, and builds a dynamic dictionary with a queue and a moving-averaged encoder. SimCLR (Chen et al., 2020) further pushes the SOTA benchmark with careful composition of data augmentations as well as introducing a learnable nonlinear transformation between the representation and the contrastive loss. However, all these algorithms concentrate primarily on the generation of pseudo-labels and contrastive pairs, whereas in COMRL scenario, we are naturally given the task labels and transitions. We therefore re-invent the query-key structure of MoCo on inter-task meta-batches, and demonstrate superior performance and efficiency compared to other methods such as distance metric learning used in FOCAL.

Context-Based Offline Meta-RL Context-based offline meta-RL (COMRL) employs models with memory such as recurrent (Duan et al., 2016; Wang et al., 2016; Fakoor et al., 2019), recursive (Mishra et al., 2017) or probabilistic (Rakelly et al., 2019) structures to achieve fast adaptation by aggregating experience into a latent representation on which the policy is conditioned. A number of existing COMRL algorithms build upon the SOTA off-policy meta-RL method PEARL (Rakelly et al., 2019), which introduces a probabilistic permutation-invariant context encoder, along with a design that disentangles task inference and control by distinct sampling strategies. To address the bootstrapping error problem (Kumar et al., 2019) for offline learning, framework like FOCAL enforces behavior regularization (Wu et al., 2019), which constrains the distribution mismatch between the behavior and learning policies in actor-critic objectives. In this paper, we follow the same paradigm.

Most similar to our work are those of Li et al. (2020) and Raileanu et al. (2020). Li et al. (2020) proposed an end-to-end and model-free framework of offline context-based meta-RL and trained the task inference module with self-supervised distance metric loss. They made the assumption that task identity can be efficiently inferred from very few randomly sampled transitions (s_t, a_t, s'_t, r_t), which severely breaks down when reward and transition functions are non-distinguishable across tasks over a large portion of the state-action support, such as in sparse reward scenarios. To achieve more effective task inference, our method introduces attention mechanism, particularly batch-wise attention to attend to transition samples that contain more information of task identity. We also apply attention modules along the transition sequence, which we term sequence-wise attention in context encoder, to better capture the correlation within transition dimensions.

Raileanu et al. (2020) on the other hand, incorporates attention in form of transformer encoders, to generate dynamics and policy embeddings for fast adaptation of value function to new tasks. Their framework involves several learning phases, and may potentially be generalized to offline meta-RL problem by training a task/context encoder instead of dynamic encoder on complete transition tuples (s_t, a_t, s'_t, r_t) from logged data. However, the attention module here is naively used as a feature extractor without ablation study of the design. In contrast, our paper not only directly addresses the OMRL problem in an end-to-end fashion, but also introduces carefully-designed attention networks and contrastive learning mechanism as key pieces for provably more efficient task inference and learning of control.

3. Method

In this paper we tackle the COMRL problem and follow the procedure described in FOCAL (Li et al., 2020), by first learning an effective representation of tasks on latent space $\mathcal{Z}$, on which a single universal policy is conditioned and trained with behavior-regularized actor-critic method (Wu et al., 2019). As an improved version over FOCAL, our main contribution is twofold:

1. To our best knowledge, we are the first to apply attention mechanism in meta/multi-task RL setting, for the sole purpose of learning better task representations. Our carefully-designed transformer architectures, particularly the batch-wise channel attention, has proven to significant robustify prior COMRL methods against MDP ambiguity and sparse reward.
2. We propose a novel matrix-form Momentum Contrast (He et al., 2020) for task presentation learning in COMRL, with provably better performance.

3.1. Problem Setup

Consider a family of stationary MDPs defined by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ where $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ are the corresponding state space, action space, transition function, reward function and discount factor. A task $\mathcal{T}$ is defined as an instance of $\mathcal{M}$, which is associated with a time-invariant transition and reward function, $\mathcal{P}(s'|s, a) \in \mathcal{P}$ and $\mathcal{R}(s, a) \in \mathcal{R}$, respectively. For the scope of this work, we focus on a set of tasks which share the same state and action space. Consequently, a task distribution can be modeled as a joint distribution of $\mathcal{P}$ and $\mathcal{R}$, usually can be factorized:

$$p(\mathcal{T}) = p(\mathcal{P}, \mathcal{R}) = p(\mathcal{P})p(\mathcal{R}) \tag{1}$$

In offline setting, each task $\mathcal{T}_i$ (i being the task label) is associated with a static dataset of transitions $\mathcal{D}_i = \{(s_i, a_i, s'_i, \mathcal{R}_i(s_i, a_i))\}$. A *meta-batch* $\mathcal{B}$ is a set of mini-batches $\mathcal{B}_i$ drawn from each $\mathcal{D}_i$. Given a meta-optimization objective such as the behavior-regularized actor-critic loss (Li et al., 2020)

$$\begin{aligned}\mathcal{L}(\theta, \psi) &= \mathbb{E}_{\mathcal{D}_i \sim p(\mathcal{D})}[\mathcal{L}_{\text{actor}}(\mathcal{D}_i; \theta) + \mathcal{L}_{\text{critic}}(\mathcal{D}_i; \psi)] \quad (2)\\ &= \mathbb{E}_{\mathcal{D}_i \sim p(\mathcal{D})}[\mathcal{L}_{\mathcal{D}_i}(\theta, \psi)] \quad (3)\end{aligned}$$

where $\mathcal{L}_{\mathcal{D}_i}(\theta, \psi)$ is the objective evaluated on transition samples drawn from $\mathcal{D}_i$, parameterized by θ and ψ. Assuming a common uniform distribution for a set of n tasks, the meta-training procedure turns into minimizing the average losses across all training tasks

$$\hat{\theta}_{\text{meta}}, \hat{\psi}_{\text{meta}} = \arg\min_{\theta, \psi} \frac{1}{n} \sum_{k=1}^{n} \mathbb{E}\left[\mathcal{L}_{\mathcal{D}_k}(\theta, \psi)\right] \tag{4}$$

For COMRL problem, a task distribution corresponds to a family of MDPs on which a single universal policy is supposed to perform well. Since the MDP family is considered partially observed if no task identity information is given, a task inference module $E_\phi(\boldsymbol{z}|\boldsymbol{c})$ is required to map context information $\boldsymbol{c} \sim \mathcal{D}$ to a latent task representation $\boldsymbol{z} \in \mathcal{Z}$ to form an augmented state, i.e.,

$$\mathcal{S} \leftarrow \mathcal{S} \times \mathcal{Z}, \quad \boldsymbol{s} \leftarrow \text{concat}(\boldsymbol{s}, \boldsymbol{z}) \tag{5}$$

Such an MDP family is formalized as a Task-Augmented MDP (TA-MDP) in (Li et al., 2020). Additionally, Li et al. (2020) provided arguments to show that a good task representation $\boldsymbol{z}$ is beneficial for optimization of the meta-objective in Eqn 4, which is the prime focus of this paper. We now show how we address the issue with the proposed attention architectures and contrastive learning framework.

3.2. Attention Architectures

We employ two forms of attention in the context encoder $E_\phi(\boldsymbol{z}|\boldsymbol{c})$: **residual batch-wise channel attention**

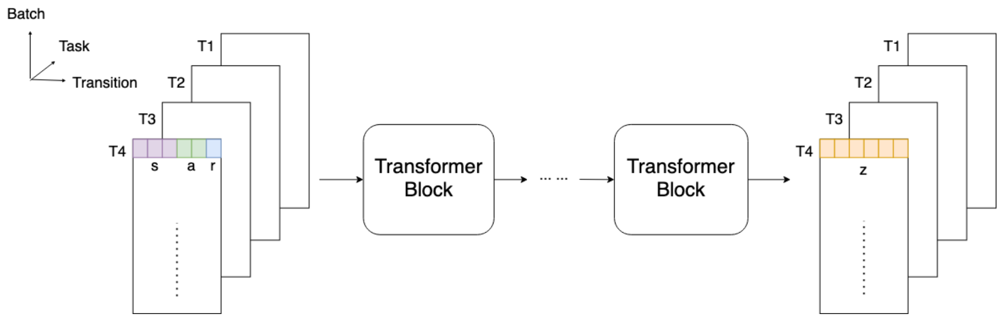


Figure 1. **Context encoder as a stack of transformer blocks.**

and **residual sequence-wise self-attention**, for the sole purpose of learning better task representations. The architectures are shown in Figure 2.

Residual Batch-Wise Channel Attention When performing task inference, transitions inside the same batch may contribute differently to learning of the networks, especially in sparse reward situations. Intuitively, transitions with non-zero rewards contain more information regarding the task identity. Therefore, we utilize channel attention along the batch dimension to account for this inhomogeneity by computing a scalar multiplier for every sample.

Residual Sequence-Wise Self-Attention Sequence attention focuses on capturing the correlation within the transition dimensions. Consider the input from previous layer $X^{(l-1)} \in \mathbb{R}^{T\times B\times C}$ where T, B, C are the meta-batch, batch and transition dimensions respectively. Our proposed sequence attention first reshape the input tensor as $(TB) \times 1 \times C$, taking the transition dimension as the feature dimension and apply a linear map $f^{(l-1)}$ to obtain its embedding, followed by (multi-head) self-attention (MHA) and layer normalization (LN) (Ba et al., 2016). The dimensions of the output $Y^{(l)}$ are then restored as $T \times B \times Z$, with Z being the latent dimension:

$$Y^{(l)} = \mathrm{LN}(X^{(l-1)} + \mathrm{MHA}(f^{(l-1)}(X^{(l-1)}))) \quad (6)$$

To stack attention blocks as in Figure 1, every pair of blocks can be connected either *in series* or *in parallel*, both we investigated in our experiments. It's worth noting that we also experimented with various attention mechanism in the downstream actor-critic networks in our algorithm, but found none converged. This is consistent with the observations made by Mishra et al. (2017) and Parisotto et al. (2020) that classical attention modules like transformers in RL agents are extremely unstable to optimize. We leave the exploration of stabilized attention architectures of COMRL-learners in future work.

3.3. The Contrastive Learning Framework

Similar to MoCo (He et al., 2020), we formulate contrastive learning as dictionary look-up with momentum. Consider

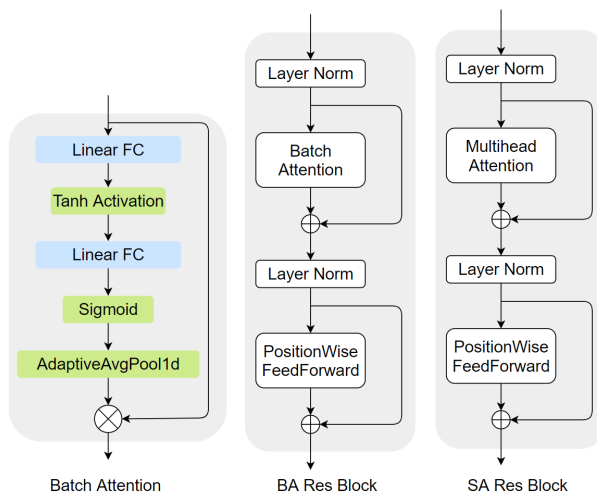


Figure 2. **Attention blocks for task inference.**

an encoded latent query vector $\boldsymbol{z}^q$ and a set of K encoded latent key vectors $\{\boldsymbol{z}_0^k, \boldsymbol{z}_1^k, \boldsymbol{z}_2^k, ...\}$. Suppose one of the keys $\boldsymbol{z}_+^k$ is the only match to $\boldsymbol{z}^q$, the building block of our contrastive learning framework is the InfoNCE (Oord et al., 2018) objective:

$$\mathcal{L}_z = -\log \frac{\exp(\boldsymbol{z}^q \cdot \boldsymbol{z}_+^k/\tau)}{\sum_{i=0}^K \exp(\boldsymbol{z}^q \cdot \boldsymbol{z}_i^k/\tau)} \quad (7)$$

where τ is a temperature hyper-parameter (Wu et al., 2018). However, unlike the SOTA contrastive learning techniques studied extensively in computer vision field (He et al., 2020; Chen et al., 2020) which require construction of positive pairs via data augmentation, our task representation learning problem in COMRL is naturally endowed with similarity structures by task labels. To ensure maximum sample efficiency, for each pair of meta-batches $\mathcal{B} = \{\mathcal{B}_i \sim \mathcal{D}_i | i = 1, ..., T\}$ where T is the meta-batch size, one can construct T ($2T$ if we switch the inputs to the query and key encoders) infoNCE objectives by taking the average latent vector of each task as the query. Namely, given a of meta-batch of encoded queries $\{\boldsymbol{z}_i^q\} = \{E^q(\mathcal{B}_i; \phi_q)\}$ and keys $\{z_i^k\} = \{E^k(\mathcal{B}_i; \phi_k)\}$, our proposed contrastive loss is

$$\mathcal{L}_z = -\sum_{i=0}^{T} \log \frac{\exp(z_i^q \cdot z_i^k/\tau)}{\sum_{j=0}^{T} \exp(z_i^q \cdot z_j^k/\tau)} \tag{8}$$

which can be written in a matrix-form

$$\mathcal{L}_z = -\operatorname{Tr}(M), \quad M_{ij} = \log \frac{\exp(z_i^q \cdot z_j^k/\tau)}{\sum_{j=0}^{T} \exp(z_i^q \cdot z_j^k/\tau)} \tag{9}$$

Intuitively, it is the log loss of a $(T+1)$-way softmax-based classifier trying to classify each z_i^q as z_i^k. With this interpretation, we now provide a formal analysis of the matrix-form momentum contrast objective above. Assuming a finite cardinality N of the task set $\mathcal{T}$, a multi-class classifier is a function $g : \mathcal{C} \to \mathbb{R}^N$ whose output coordinates are indexed by the task label, where $\mathcal{C}$ is the space of raw transitions $(s, a, s', \mathcal{R}(s, a))$. Without loss of generality, we consider a convex logistic loss $l(\boldsymbol{v}) = \log(1 + \sum_i \exp(-v_i))$ for $\boldsymbol{v} \in \mathbb{R}^T$, as a simplified form of Eqn. 7. We proceed with the following definitions adopted from (Arora et al., 2019):

Definition 3.1 (Supervised Contrastive Loss)

$$L_{sup}(\mathcal{T}, g) := \mathbb{E}_{\substack{\mathcal{T}_i \sim p(\mathcal{T}) \\ c \sim \mathcal{D}}}[\ell(\{g(c)_i - g(c)_{i'}\}_{i' \neq i})] \tag{10}$$

To use the task encoder $E_\phi(\boldsymbol{z}|\boldsymbol{c})$ with a classifier, a matrix $W \in \mathbb{R}^{N \times Z}$, such that $g(c) = WE(c)$, is trained to minimize $L_{sup}(\mathcal{T}, WE)$. Z is the dimension of the task latent space $\mathcal{Z}$. Hence the supervised loss of E on $\mathcal{T}$ is defined as

$$L_{sup}(\mathcal{T}, E) = \inf_{W \in \mathbb{R}^{N \times Z}} L_{sup}(\mathcal{T}, WE) \tag{11}$$

In practice, we estimate the mean task representation of z^q and z^k using its batch-wise mean

$$\mu_i^{q,k} := \mathbb{E}_{c \sim \mathcal{D}_i}[E^{q,k}(c)] \approx \mathbb{E}_{c \sim \mathcal{B}_i}[E^{q,k}(c)] \tag{12}$$

which induces the following definitions:

Definition 3.2 (Mean Task Classifier) *For an encoder function E and a task set $\mathcal{T}$ of cardinality N, the mean task classifier is W^μ whose i^{th} row is the mean μ_i of representations of inputs with task label i. We use as a shorthand for its loss $L_{sup}^\mu(\mathcal{T}, E) := L_{sup}(\mathcal{T}, W^\mu E)$*

Definition 3.3 (Averaged Supervised Contrastive Loss) *Average loss for an encoder function E on (T+1)-way classification of task representation is defined as*

$$L_{sup}(E) := \mathop{\mathbb{E}}_{\{\mathcal{T}_i\}_{i=0}^T \sim p(\mathcal{T})} \left[L_{sup}(\{\mathcal{T}_i\}_{i=0}^T, E)\right] \tag{13}$$

The average supervised loss of its mean classifier is

$$L_{sup}^\mu(E) := \mathop{\mathbb{E}}_{\{\mathcal{T}_i\}_{i=0}^T \sim p(\mathcal{T})} \left[L_{sup}^\mu(\{\mathcal{T}_i\}_{i=0}^T, E)\right] \tag{14}$$

which we prove in Appendix that

Theorem 3.1 *The matrix-form momentum contrast objective $\mathcal{L}_z$ (Eqn 9) is equivalent to the average supervised loss of its mean classifier L_{sup}^μ (Eqn 14) if $E = E^q(\boldsymbol{z}|\boldsymbol{c})$ is the query encoder and the mean task classifier W^μ whose i^{th} row is the mean of latent key vectors with task label i.*

If we compare our proposed loss function with classical unsupervised contrastive loss

Definition 3.4 (Unsupervised Loss)

$$L_{un}(E) := \mathbb{E}\left[\ell(\{E(c)^T(E(x^+) - E(x_i^-))\}_{i=1}^k)\right] \tag{15}$$

By **Lemma 4.3** in (Arora et al., 2019), using convexity of ℓ and Jensen's inequality, assuming no repeated task labels in each meta-batch, we have

Theorem 3.2 *For all E in $\mathcal{E}$*

$$L_{sup}(E) \leq L_{sup}^\mu(E) \leq L_{un}(E) \tag{16}$$

combined with Theorem 3.1, it shows that our proposed matrix-form momentum contrast is a better surrogate for L_{sup} than ordinary unsupervised contrastive losses.

The training scheme of our proposed inter-task matrix-form momentum contrast is illustrated in Fig 3. It can be implemented as a cross-entropy loss with the target probability distribution in form of a $T \times T$ diagonal matrix. As in (He et al., 2020), we hypothesize that desirable dictionaries for contrastive learning should include consistent keys as they evolve during training. Therefore denoting the parameters of E^k as θ_k and those of E^q as θ_q, we soft-update θ_k by:

$$\theta_k \leftarrow m\theta_k + (1 - m)\theta_q \tag{17}$$

where $m \in [0, 1)$ is a momentum hyper-parameter.

3.4. Variance of FOCAL++

In experiments, we found that our proposed algorithm, FOCAL++, which combines attention mechanism and matrix-form momentum contrast, exhibit significant less variance

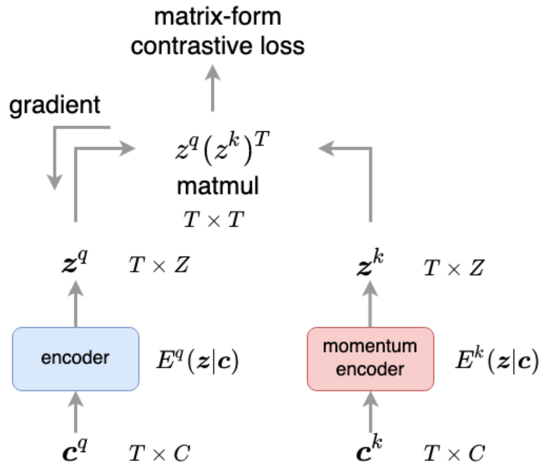


Figure 3. **Inter-task matrix-form momentum contrast.** Given two meta-batches of transitions $\{\boldsymbol{c}^q\}$ and $\{\boldsymbol{c}^k\}$, a quickly progressing query encoder and a slowly progressing key encoder compute the corresponding batch-wise mean task representations in latent space $\mathcal{Z}$. A matrix multiplication is performed between the set of query and key vectors to produce the supervised contrastive loss in Eqn 9. T, C, Z are the meta-batch, transition and latent space dimensions respectively.

compared to the baselines on tasks with sparse reward (Table 1). We provide a proof of this observation for a simplied version of FOCAL++, by only considering the batch channel attention along with contrastive learning objective defined in Eqn 9 and 14, *in presence of sparse reward. Assuming all tasks differ only in reward function*, we begin with the following definition

Definition 3.5 (Task with Sparse Reward) *For a dataset $\mathcal{D}_i = \{(s_i, a_i, s_i', \mathcal{R}_i(s_i, a_i))\}$ sampled from any task $\mathcal{T}_i$ with sparse reward, it can be decomposed as a disjoint union of two sets of transitions*

$$\mathcal{D}_i = \{(s_i, a_i, s_i', \mathcal{R}_i(s_i, a_i))\} \cup \{(s_i, a_i, s_i', 0)\} \tag{18}$$
$$= \{c_i\} \cup \{c_s\} \tag{19}$$

where $\{c_i\}$ is the set of non-sparse reward transitions, which are unique to task $\mathcal{T}_i$. $\{c_s\}$ is the set of sparse-reward transitions, which are shared cross all tasks. By definition of the reward funciton, the state-action support of the two disjoint sets are also disjoint, meaning random variables $c_i \sim \{c_i\}$, $c_s \sim \{c_s\}$ are independent.

Definition 3.6 (Batch-Wise Channel Attention) *A Batch-wise channel attention A assigns an inhomogeneous weight to the batch-wise estimation of the mean task representation of $\boldsymbol{z}^q$ and $\boldsymbol{z}^k$ in Eqn 12*

$$\mu_i^{q,k}(A) := \mathbb{E}_{c \sim \mathcal{D}_i}[A(c)E^{q,k}(c)] \tag{20}$$
$$= p_i \mathbb{E}_{c_i \sim \{c_i\}}[A(c_i)E^{q,k}(c_i)] + p_s \mathbb{E}_{c_s \sim \{c_s\}}[A(c_s)E^{q,k}(c_s)] \tag{21}$$

where $p_i = \{c \in \{c_i\} | c \sim \mathcal{D}_i\}$, $p_s = \{c \in \{c_s\} | c \sim \mathcal{D}_i\}$ and A is normalized such that $\mathbb{E}_{c \sim \mathcal{D}_i}[A(c)] = 1$. One can easily see that $p_i + p_s = 1$ by Definition 3.5.

Theorem 3.3 *Given a learned batch-wise channel attention and context encoder $\widehat{A}, \widehat{E} \in arg\ min_{A \in \mathcal{A}, E \in \mathcal{E}} L^{\mu}_{sup}(A, E)$ that minimize the contrastive learning objective, we have*

$$Var(\mu_i^{q,k}(\widehat{A})) \leq Var(\mu_i^{q,k}) \tag{22}$$

We prove Theorem 3.3 in Appendix. Intuitively, lower variance of the mean task representations would lead to lower variance of the conditioned policy and its performance, assuming the policy and value functions are continuous. Table 1 empirically reflects such intuition. We leave a more rigorous proof of this statement to future work.

4. Experiments

In the following experiments, all trials are averaged over three random seeds. The offline training data are generated in accordance with the protocol in (Li et al., 2020) by training stochastic SAC (Haarnoja et al., 2018) models for every single task and roll out policies saved at every checkpoint to collect trajectories. The offline training datasets can be collected as a selection of the saved trajectories, which facilitates tuning of the performance level and state-action distributions of the datasets for each task (Figure 6). A complete description of hyperparameters used and the details of the experiments is included in Appendix.

4.1. Environmental Setup

Sparse-Point-Robot is a 2D-navigation task with sparse reward, introduced in (Rakelly et al., 2019). Tasks **differ only in reward function**. Each task is associated with a goal uniformly distributed on a unit semicircle. Starting from the origin, the agent must navigate to a previously unseen goal with non-zero reward given only when inside the goal radius. The agent is trained to navigate to a training set of goals, then tested on a distinct set of unseen test goals. We use radius of 0.2 in this paper.

Point-Robot-Wind is a variant of Sparse-Point-Robot. Task **differ only in transition function**. Each task is associated with the same reward but a distinct "wind" sampled uniformly from $[-l, l]^2$. Every time the agent takes a step, it drifts by the wind vector. We use $l = 0.05$ in this paper.

Sparse-Cheetah-Vel, Sparse-Ant-Dir are sparse-reward variants of the popular meta-RL benchmarks Half-Cheetah-Vel and Sparse-Ant-Dir based on MuJoCo environments, introduced by (Finn et al., 2017) and (Rothfuss et al., 2018). Due to time constraint, we leave the introduction and experiments of these two environments in Appendix.

4.2. Sample Efficiency and Performance

There are three key pieces of FOCAL++: contrastive learning objective, batch-wise attention and sequence-wise attention. As ablations, we add one piece each time and compare the three variants with FOCAL and FOCAL++ (all pieces added). An illustration of the training curves is given in Figure 8, in which it is observed consistently that variants of FOCAL++ achieve faster convergence and higher asymptotic performance. The full statistics are summarized in Table 1. One can clearly observe the evidence for statement in Section 3.4 that on tasks with sparse reward, FOCAL++ exhibits low variance in performance by virtue of its design.

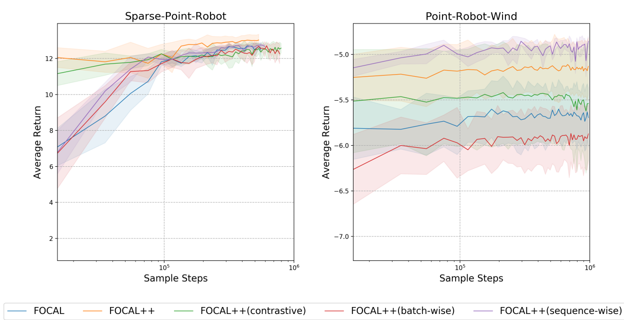


Figure 4. Average episodic testing return of variants of FOCAL++ vs. FOCAL on two meta-environments. Results with more baselines such as MBML (Li et al., 2019b) are given in Appendix.

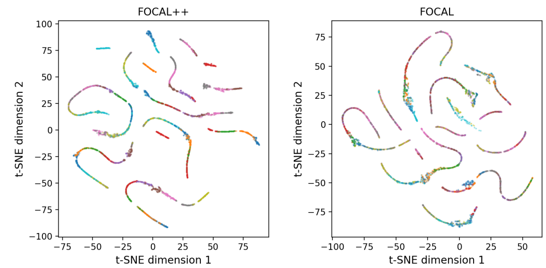
Algorithm	Sparse-Point-Robot	Point-Robot-Wind
FOCAL	$12.25_{(0.47)}$	$-5.67_{(0.30)}$
FOCAL++ (contrastive)	$12.53_{(0.31)}$	$-5.55_{(0.63)}$
FOCAL++ (batch-wise)	$12.54_{(0.23)}$	$-5.89_{(0.34)}$
FOCAL++ (sequence-wise)	$12.64_{(0.14)}$	$\mathbf{-4.90}_{(0.04)}$
FOCAL++ (parallel)	$12.90_{(0.02)}$	$-5.77_{(0.19)}$
FOCAL++ (series)	$\mathbf{13.02}_{(0.30)}$	$-5.49_{(0.47)}$

Table 1. Average testing return (standard deviation in parenthesis) of FOCAL and variants of FOCAL++.

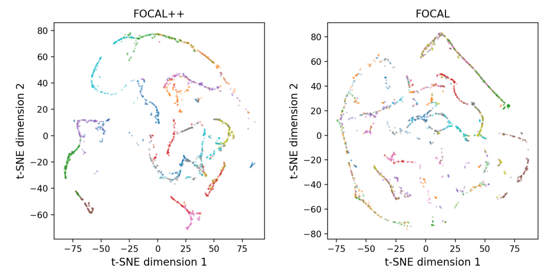
4.3. Analysis of Learned Representations

Since the prime focus of our proposed attention and contrastive learning framework is to learn better task representations, it's crucial to evaluate the quality of the embeddings with proper metrics. A visualization of the 2D projection of the latent space by t-SNE (Van der Maaten & Hinton, 2008) is illustrated in Figure 5. Compared to baseline FOCAL, it's visually evident that our method obtains higher clustering quality of task embeddings on both environments.

For quantitative assessment, we compute the root mean square (RMS) and *effective separation rate (ESR)* introduced by Li et al. (2020). ESR estimates the the percentage of embedding vector pairs of different tasks whose distance on latent space $\mathcal{Z}$ is larger than the expectation of randomly distributed pairs, i.e., $\sqrt{2l/3}$ on $(-1,1)^{\mathcal{Z}}$. Table 2 demonstrates a clear improvement on both metrics of FOCAL++.



(a) Sparse-Point-Robot



(b) Point-Robot-Wind

Figure 5. t-SNE visualization of the learned task embeddings $\boldsymbol{z}^q$ on Sparse-Point-Robot and Point-Robot-Wind. Each point represents a query vector which is color-coded according to its task label.

Environment	Algorithm	RMS	ESR
Sparse-Point-Robot	FOCAL	1.283	0.226
	FOCAL++	**1.288**	**0.281**
Point-Robot-Wind	FOCAL	0.892	0.077
	FOCAL++	**1.065**	**0.100**

Table 2. Embedding Statistics

4.4. Robustness to MDP Ambiguity and Sparse Reward

As discussed in Section 1, the key challenges that FOCAL++ tries to address with attention mechanism are MDP ambiguity and sparse reward. Take Sparse-Point-Robot for example, for a task with a goal on the semicircle, the state distribution exhibits specific pattern associated with the quality of the training data (Figure 6). Moreover, sparse reward makes a considerable portion of transitions (states outside the target circle) useless for task inference. Without an effective task representation, the policy/value function could easily overfit to the state-action distribution of the data, which is sensitive to distribution shift (MDP ambiguity). Attention mechanism, especially the batch-wise channel attention, helps the context encoder attend to the informative portion of the input transitions, and therefore significantly improve the robustness of the learned policies. To verify this hypothesis, we tested a variant of FOCAL++ with only batch-wise channel attention on combinations of 3 datasets: expert, medium and random. We adopted the result of baseline FOCAL directly from (Li et al., 2020). Shown in Table 3, we observe that overall the performance drop due to distribution shift is significantly less when channel attention is applied. Moreover, FOCAL++ consistently achieve expert-level performance on all combinations of training and testing sets, even when the data is generated by close-to-random behavior policies, which are all decisive evidence for the robustness of our method.

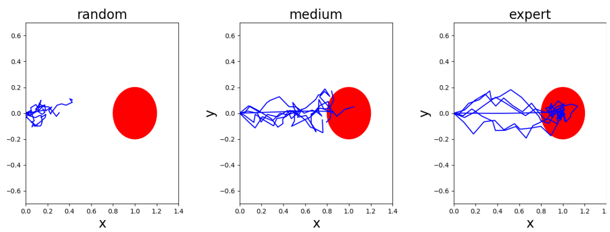


Figure 6. Distribution of rollout trajectories of trained SAC policies of three performance levels: random, medium and expert. Since reward is sparse, only states that lie in the red circle are given non-zero rewards, making meta-learning more challenging and sensitive to data distributions. Adopted from (Li et al., 2020).

Training	Testing	FOCAL$_{\text{(drop)}}$	FOCAL++$_{\text{(drop)}}$
expert	expert	$8.16_{(-)}$	$12.60_{(-)}$
medium	medium	$8.44_{(-)}$	$12.54_{(-)}$
random	random	$2.34_{(-)}$	$10.81_{(-)}$
expert	medium	$7.12_{(1.04)}$	$12.47_{(0.13)}$
expert	random	$4.43_{(3.73)}$	$10.17_{(2.43)}$
medium	expert	$8.25_{(0.19)}$	$12.44_{(0.10)}$
medium	random	$6.76_{(1.68)}$	$10.49_{(2.05)}$
random	expert	–	$10.35_{(0.46)}$
random	medium	–	$10.04_{(0.77)}$

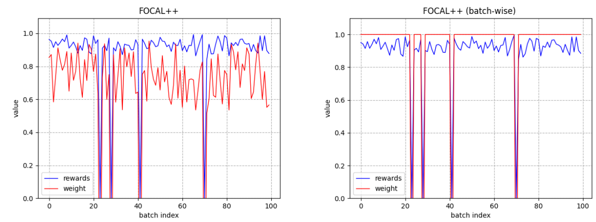
Table 3. Average testing return of FOCAL and FOCAL++ with batch-wise channel attention on Sparse-Point-Robot tasks with different qualities/distributions of training/testing sets. The numbers in parethesis are the performance drop due to distribution shift (compared to the scenario where the testing distribution equals the training distribution).

4.5. More Ablations

In this section, we carry out more ablation studies to examine the specific design choices of FOCAL++.

4.5.1. Attention on Sample Selection

The attention mechanism employed in FOCAL++, particularly the batch-wise channel attention is designed to capture the inhomogeneous contribution of each transition sample to inferred task representations. Intuitively, on sparse-reward environment, a desired attentive encoder should assign significantly larger weights to transitions with non-sparse reward. To verify this design choice, on Sparse-Point-Robot, we draw a random batch and visualize the output weights given by the encoder along the batch dimension in Figure 7. In contrast to the baseline which treats each sample equally, we show that the learned weight distribution of both FOCAL++ and FOCAL++ with only channel attention perfectly match the sparse reward distribution of the samples.



(a) FOCAL++ (b) FOCAL++ (batch-wise)

Figure 7. Reward and attention weight distributions along batch dimension.

4.5.2. Momentum in Contrastive Learning

As hypothesized in (He et al., 2020), desirable dictionaries for contrastive learning should include consistent keys as they evolve during training, for which they implemented a soft-update mechanism for the parameters of the key encoder (Eqn 17). To verify this intuition, we compare a soft-update strategy with $m = 0.9$ to a hard-update (query and key encoders are identical) for FOCAL++ with various attention architectures. Results are conclusive that the soft-update mechanism in contrastive learning is a key factor for performance boost.

Attention	Soft-update	Hard-update
None	$\mathbf{12.78 \pm 0.09}$	11.53 ± 1.42
Batch-wise	$\mathbf{13.15 \pm 0.06}$	11.41 ± 0.48
Sequence-wise	$\mathbf{12.83 \pm 0.08}$	10.10 ± 2.26
Parallel	$\mathbf{12.83 \pm 0.08}$	8.78 ± 0.92
Serialize	$\mathbf{12.20 \pm 0.22}$	7.07 ± 0.42

Table 4. Average testing return of FOCAL++ with soft/hard update and vaious attention architectures on Sparse-Point-Robot tasks. FOCAL++ with soft update significantly outperforms its counterpart in all trials.

5. Conclusion

In this work, we address the understudied COMRL problem and improve upon the existing SOTA baselines such as FOCAL, by focusing on more effective and robust learning of task representations. Key to our framework is the combination of intra-task attention mechanism and inter-task contrastive learning, for which we provide theoretical grounding and experimental evidence on the superiority of our design.

References

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Barati, E. and Chen, X. An actor-critic-attention mechanism for deep reinforcement learning in multi-view environments. *arXiv preprint arXiv:1907.09466*, 2019.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 539–546. IEEE, 2005.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dorfman, R. and Tamar, A. Offline meta reinforcement learning. *arXiv preprint arXiv:2008.02598*, 2020.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. Rl ^{2}: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Fakoor, R., Chaudhari, P., Soatto, S., and Smola, A. J. Meta-q-learning. *arXiv preprint arXiv:1910.00125*, 2019.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Gottesman, O., Johansson, F., Komorowski, M., Faisal, A., Sontag, D., Doshi-Velez, F., and Celi, L. A. Guidelines for reinforcement learning in healthcare. *Nat Med*, 25(1): 16–18, 2019.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Khetarpal, K., Riemer, M., Rish, I., and Precup, D. Towards continual reinforcement learning: A review and perspectives. *arXiv preprint arXiv:2012.13490*, 2020.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, pp. 11784–11794, 2019.

Kumar, S., Parker, J., and Naderian, P. Adaptive transformers in rl. *arXiv preprint arXiv:2004.03761*, 2020.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, H., Eigen, D., Dodge, S., Zeiler, M., and Wang, X. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–10, 2019a.

Li, J., Vuong, Q., Liu, S., Liu, M., Ciosek, K., Iskov Christensen, H., and Su, H. Multi-task batch reinforcement learning with metric learning. *arXiv*, pp. arXiv–1909, 2019b.

Li, L., Yang, R., and Luo, D. Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. *arXiv preprint arXiv:2010.01112*, 2020.

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. Recurrent models of visual attention. *arXiv preprint arXiv:1406.6247*, 2014.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Oh, J., Singh, S., Lee, H., and Kohli, P. Zero-shot task generalization with multi-task deep reinforcement learning. In *International Conference on Machine Learning*, pp. 2661–2670. PMLR, 2017.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., Jayakumar, S., Jaderberg, M., Kaufman, R. L., Clark, A., Noury, S., et al. Stabilizing transformers for reinforcement learning. In *International Conference on Machine Learning*, pp. 7487–7498. PMLR, 2020.

Raileanu, R., Goldstein, M., Szlam, A., and Fergus, R. Fast adaptation to new environments via policy-dynamics value functions. In *International Conference on Machine Learning*, pp. 7920–7931. PMLR, 2020.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pp. 5331–5340, 2019.

Rothfuss, J., Lee, D., Clavera, I., Asfour, T., and Abbeel, P. Promp: Proximal meta-policy search. *arXiv preprint arXiv:1810.06784*, 2018.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Song, X., Jiang, Y., Tu, S., Du, Y., and Neyshabur, B. Observational overfitting in reinforcement learning. *arXiv preprint arXiv:1912.02975*, 2019.

Sukhbaatar, S., Grave, E., Bojanowski, P., and Joulin, A. Adaptive attention span in transformers. *arXiv preprint arXiv:1905.07799*, 2019.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

Wang, W. and Shen, J. Deep visual attention prediction. *IEEE Transactions on Image Processing*, 27(5):2368–2378, 2017.

Whiteson, S., Tanner, B., Taylor, M. E., and Stone, P. Protecting against evaluation overfitting in empirical reinforcement learning. In *2011 IEEE symposium on adaptive dynamic programming and reinforcement learning (ADPRL)*, pp. 120–127. IEEE, 2011.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

Ye, D., Liu, Z., Sun, M., Shi, B., Zhao, P., Wu, H., Yu, H., Yang, S., Wu, X., Guo, Q., et al. Mastering complex control in moba games with deep reinforcement learning. In *AAAI*, pp. 6672–6679, 2020.

Yin, M., Tucker, G., Zhou, M., Levine, S., and Finn, C. Meta-learning without memorization. *arXiv preprint arXiv:1912.03820*, 2019.

Appendices

A. Pseudo-code

Algorithm 1 FOCAL++ Meta-training

Given:

- Pre-collected batch $\mathcal{D}_i = \{(s_j, a_j, s'_j, r_j)\}_{j:1,\dots,N}$ of a set of training tasks $\{\mathcal{T}_i\}_{i=1,\dots,n}$ drawn from $p(\mathcal{T})$
- Learning rates $\alpha_1, \alpha_2, \alpha_3$, temperature τ, momentum m

Initialize context replay buffer $\mathcal{C}_i$ for each task $\mathcal{T}_i$
Initialize attentive task encoders $E_\phi^{q,k}(z|c)$, learning policy $\pi_\theta(a|s,z)$ and Q-network $Q_\psi(s,z,a)$ with parameters ϕ_q, ϕ_k, θ and ψ
while not done **do**
  for each $\mathcal{T}_i$ **do**
    for t = $0, T-1$ **do**
      Sample mini-batches of B transitions $\{(s_{i,t}, a_{i,t}, s'_{i,t}, r_{i,t})\}_{t:1,\dots,B} \sim \mathcal{D}_i$ and update $\mathcal{C}_i$
    end
  end
  Sample a pair of query-key meta-batches of T tasks $\sim p(\mathcal{T})$
  for step in training steps **do**
    for each $\mathcal{T}_i$ **do**
      Sample mini-batches c_i and $b_i \sim \mathcal{C}_i$ for E_ϕ^q and policy training (b_i, c_i are identical by default)
      Compute $\boldsymbol{z}_i^q = E_\phi^q(c_i)$
      for each $\mathcal{T}_j$ **do**
        Sample mini-batches c_j from $\mathcal{C}_j$ and compute $\boldsymbol{z}_j^k = E_\phi^k(c_j)$
        $\mathcal{M}_{ij} = \mathcal{M}_z(\boldsymbol{z}_i^q, \boldsymbol{z}_j^k)$ ▷ **matrix-form momentum contrast**
      end
      $\mathcal{L}_{actor}^i = \mathcal{L}_{actor}(b_i, E_\phi^q(c_i))$
      $\mathcal{L}_{critic}^i = \mathcal{L}_{critic}(b_i, E_\phi^q(c_i))$
    end
    $\mathcal{L}_z = \mathrm{Tr}(M)$
    $\phi_q \leftarrow \phi_q - \alpha_1 \nabla_{\phi_q} \mathcal{L}_z$
    $\phi_k \leftarrow m\phi_k + (1-m)\phi_q$ ▷ **momentum update**
    $\theta \leftarrow \theta - \alpha_2 \nabla_\theta \sum_i \mathcal{L}_{actor}^i$
    $\psi \leftarrow \psi - \alpha_3 \nabla_\psi \sum_i \mathcal{L}_{critic}^i$
  end
end

B. Definitions and Proofs

B.1. Proof of Theorem 3.1

Consider a task set $\mathcal{T} = \{\mathcal{T}_0, ..., \mathcal{T}_T\}$ drawn uniformly from $p(\mathcal{T})$. In Definition 3.1, the loss incurred by g on point $(c, \mathcal{T}_i) \in \mathcal{C}$[2] $\times \mathcal{T}$ is defined as $\ell(\{g(c)_i - g(c)_{i'}\}_{i' \neq i})$, which is a function of a T-dimensional vector of differences in the coordinates. Given the definition of the mean task classifier W^μ that $g(c) = WE(c)$ and $\ell(\boldsymbol{v}) = \log(1 + \sum_i \exp(-v_i))$, the supervised contrastive loss defined in Eqn 10 can be rewritten as

[2] $\mathcal{C} = \{(s_i, a_i, s'_i, \mathcal{R}_i(s_i, a_i))\}$ is the context space

Algorithm 2 FOCAL++ Meta-testing

Given:

- Pre-collected batch $\mathcal{D}_{i'} = \{(s_{j'}, a_{j'}, s'_{j'}, r_{j'})\}_{j':1,...,M}$ of a set of testing tasks $\{\mathcal{T}_{i'}\}_{i'=1...m}$ drawn from $p(\mathcal{T})$

Initialize context replay buffer $\mathcal{C}_{i'}$ for each task $\mathcal{T}_i$
  for each $\mathcal{T}_{i'}$ **do**
    for t = $0, T-1$ **do**
      Sample mini-batches of B transitions $c_{i'} = \{(s_{i',t}, a_{i',t}, s'_{i',t}, r_{i',t})\}_{t:1,...,B} \sim \mathcal{D}_{i'}$ and update $\mathcal{C}_{i'}$
      Compute $z^q_{i'} = E^q_\phi(c_{i'})$
      Roll out policy $\pi_\theta(a|s, z^q_{i'})$ for evaluation
    end
end

$$L_{sup}(\mathcal{T}, g) := \mathbb{E}_{\substack{\mathcal{T}_i \sim p(\mathcal{T}) \\ c_i \sim \mathcal{D}}} \left[\log \left(1 + \sum_{i' \neq i} \exp \left(\sum_j \left(W_{i'j} E(c_i)_j - W_{ij} E(c_i)_j \right) \right) \right) \right] \tag{23}$$

Since the i^{th} row of W is the mean of latent key vectors with task label i, and $E = E^q(\boldsymbol{z}|\boldsymbol{c})$ is the query encoder, Eqn 23 turns into

$$L_{sup}(\mathcal{T}, g) := \mathbb{E}_{\mathcal{T}_i \sim p(\mathcal{T})} \left[\log \left(1 + \sum_{i' \neq i} \exp \left(\boldsymbol{z}^k_{i'} \cdot \boldsymbol{z}^q_i - \boldsymbol{z}^k_i \cdot \boldsymbol{z}^q_i \right) \right) \right] \tag{24}$$

In practice, we estimate the latent vectors $\boldsymbol{z}^{q,k}_i$ using batch-wise mean to approximate the the mean task representation $\mu^{q,k}_i$. Therefore L_{sup} in Eqn 24 is equivalent to the mean task classifier L^μ_{sup} defined in Definition 3.2. One step further, assuming uniform distribution of the task set $\mathcal{T}$[3], the averaged supervised contrastive loss by Definition 3.3 is

$$\begin{aligned} L^\mu_{sup}(E) &:= \mathop{\mathbb{E}}_{\{\mathcal{T}_i\}^T_{i=0} \sim p(\mathcal{T})} \left[L^\mu_{sup}(\{\mathcal{T}_i\}^T_{i=0}, E) \right] \\ &= \frac{1}{T} \sum_{i=0}^{T} \left[\log \left(1 + \sum_{i' \neq i} \exp \left(\boldsymbol{z}^k_{i'} \cdot \boldsymbol{z}^q_i - \boldsymbol{z}^k_i \cdot \boldsymbol{z}^q_i \right) \right) \right] && (25) \\ &= -\frac{1}{T} \sum_{i=0}^{T} \log \frac{\exp \left(\boldsymbol{z}^q_i \cdot \boldsymbol{z}^k_i \right)}{\sum_{j=0}^T \exp \left(\boldsymbol{z}^q_i \cdot \boldsymbol{z}^k_j \right)} && (26) \end{aligned}$$

which is precisely the matrix-form momentum contrast objective (Eqn 8,9) if one rescales W by a factor of τ.

B.2. Proof of Theorem 3.3

With Definition 3.5 and 3.6, we hereby provide an informal proof by assuming a constant $A(c)$ on the non-sparse set $\{c_i\}$ and the sparse set $\{c_s\}$ respectively, then we have

$$\mu^{q,k}_i(A) = p_i A(c_i) \mathbb{E}_{c_i \sim \{c_i\}}[E^{q,k}(c_i)] + p_s A(c_s) \mathbb{E}_{c_s \sim \{c_s\}}[E^{q,k}(c_s)] \tag{27}$$

where $\mathbb{E}_{c \sim \mathcal{D}_i}[A(c)]$ implies $p_i A(c_i) + p_s A(c_s) = 1$. Therefore, adding the batch-wise attention is effectively modulating p_i and p_s. Since $p_i + p_s = 1$, without loss of generality, we apply the following notations:

[3]**Note that the task set $\mathcal{T}$ discussed here is a *subset* of the whole tasks and does not necessarily cover the whole support of $p(\mathcal{T})$. It is sampled for the sole purpose of computing the contrastive loss.**

$$p_i = p, \quad p_i A(c_i) = p' \tag{28}$$

$$\mathbb{E}_{c_i \sim \{c_i\}}[E^{q,k}(c_i)] = x_i^{q,k}, \quad \mathbb{E}_{c_s \sim \{c_s\}}[E^{q,k}(c_s)] = x_s^{q,k} \tag{29}$$

Assuming i.i.d x_i and x_s, which yields

$$\mathrm{Var}(\mu_i^{q,k}(A)) = \mathrm{Var}(p' x_i^{q,k} + (1-p')x_s^{q,k}) = (p')^2 \mathrm{Var}(x_i^{q,k}) + (1-p')^2 \mathrm{Var}(x_s^{q,k}) \tag{30}$$

$$\mathrm{Var}(\mu_i^{q,k}) = \mathrm{Var}(p x_i^{q,k} + (1-p)x_s^{q,k}) = p^2 \mathrm{Var}(x_i^{q,k}) + (1-p)^2 \mathrm{Var}(x_s^{q,k}) \tag{31}$$

By B.1, the averaged supervised loss $L^{\mu}_{sup}(A, E)$ is equivalent to the matrix-form contrastive objective, which can be written as

$$\begin{aligned} L^{\mu}_{sup}(A, E) &= \frac{1}{T}\sum_{i=0}^{T}\left[\log\left(1 + \sum_{i' \neq i} \exp\left((\boldsymbol{\mu}_{i'}^k - \boldsymbol{\mu}_i^k)\cdot \boldsymbol{\mu}_i^q\right)\right)\right] \\ &= \frac{1}{T}\sum_{i=0}^{T}\left[\log\left(1 + \sum_{i' \neq i} \exp\left(p'(\boldsymbol{x}_{i'}^k - \boldsymbol{x}_i^k)\cdot \boldsymbol{\mu}_i^q\right)\right)\right] \end{aligned} \tag{32}$$

where we use the definition of $\boldsymbol{\mu}$ in Eqn 27 and the fact that $x_s^{q,k}$ is the same across all tasks. Since the learned $\widehat{A}, \widehat{E} \in \arg\min_{A \in \mathcal{A}, E \in \mathcal{E}} L^{\mu}_{sup}(A, E)$, and $p' \approx p$ by the identity map initialization of the *residual* attention module, we have, for learned $\widehat{p'}$, $\widehat{\boldsymbol{x}}$ and $\widehat{\boldsymbol{\mu}}$,

$$\widehat{p'} \geq p, \quad (\widehat{\boldsymbol{x}}_{i'}^k - \widehat{\boldsymbol{x}}_i^k)\cdot \widehat{\boldsymbol{\mu}}_i^q < 0 \tag{33}$$

Now subtract 30 by 31, we have

$$\begin{aligned} \mathrm{Var}(\mu_i^{q,k}(\widehat{A})) - \mathrm{Var}(\mu_i^{q,k}) &= [(\widehat{p'})^2 - p^2]\mathrm{Var}(x_i^{q,k}) + [(1-\widehat{p'})^2 - (1-p)^2]\mathrm{Var}(x_s^{q,k}) \\ &= (\widehat{p'} - p)\left[(\widehat{p'} + p)\mathrm{Var}(x_i^{q,k}) - (2 - p - \widehat{p'})\mathrm{Var}(x_s^{q,k})\right] \\ &\leq 0, \quad if \quad p \leq \widehat{p'} \leq \frac{(2-p)\mathrm{Var}(x_s^{q,k}) - p\mathrm{Var}(x_i^{q,k})}{\mathrm{Var}(x_i^{q,k}) + \mathrm{Var}(x_s^{q,k})} \end{aligned} \tag{34}$$

The left inequality automatically holds by Eqn 33, the RHS is satisfied when

$$p \leq \frac{\mathrm{Var}(x_s^{q,k})}{\mathrm{Var}(x_i^{q,k}) + \mathrm{Var}(x_s^{q,k})} \tag{35}$$

or equivalently,

$$p_s = (1-p) \geq \frac{\mathrm{Var}(x_i^{q,k})}{\mathrm{Var}(x_i^{q,k}) + \mathrm{Var}(x_s^{q,k})} \tag{36}$$

which means when the sparsity of reward exceeds the threshold, a learned batch attention module can reduce the variance of the mean task representation $\mu_i^{q,k}$.

C. Additional Experiments

In this section, we present experimental results of more baselines (MBML and Contextual BCQ[4]). We also add Sparse-Ant-Fwd-Back and Sparse-Cheetah-Vel, two meta-environments with sparse reward derived from MuJoCo and show that FOCAL++ excel on more complex tasks compared to point-robot variants.

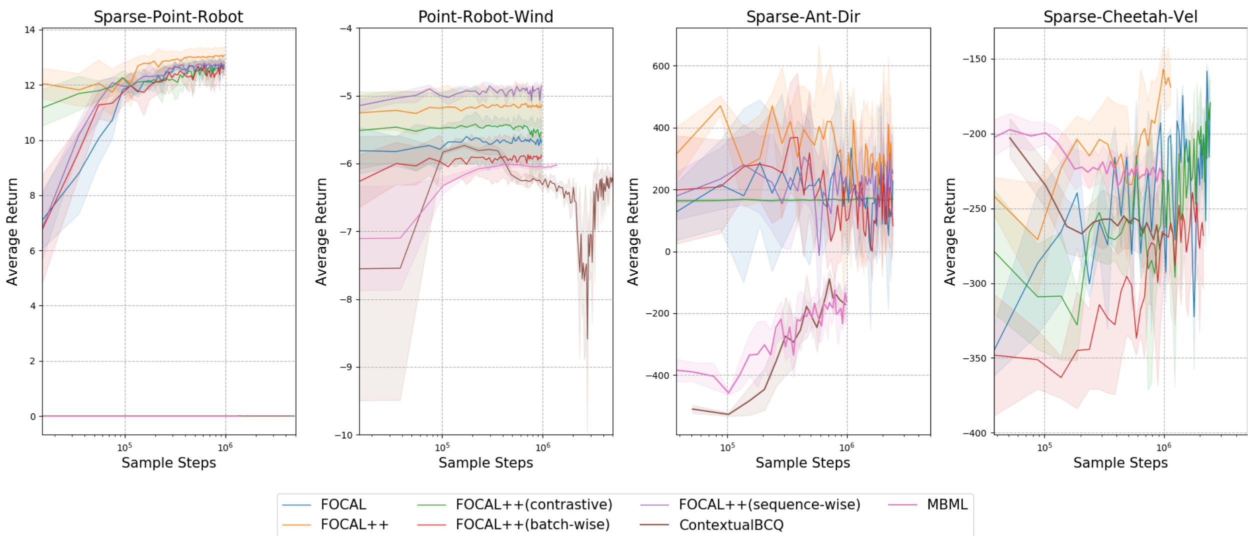


Figure 8. Extension of Figure 4 in the main text. Average episodic testing return of variants of FOCAL++ vs. baselines on four meta-environments. Note that on Sparse-Point-Robot, Contextual BCQ and MBML we implemented failed to adapt to testing environments because of sparse reward. For Ant and Cheetah, we adopted the baseline result directly from FOCAL, which were ran on non-sparse variants of the corresponding experiments. We expect their performance to be even worse if trained on sparse reward scenario (what we did for FOCAL++).

Algorithm	Sparse-Point-Robot	Point-Robot-Wind	Sparse-Cheetah-Vel	Sparse-Ant-Fwd-Back
FOCAL	$12.25_{(0.47)}$	$-5.67_{(0.30)}$	$-201.8_{(37.4)}$	$142.8_{(156.0)}$
FOCAL++ (contrastive)	$12.53_{(0.31)}$	$-5.55_{(0.63)}$	-	$166.7_{(4.9)}$
FOCAL++ (batch-wise)	$12.54_{(0.23)}$	$-5.89_{(0.34)}$	-	$167.8_{(122.2)}$
FOCAL++ (sequence-wise)	$12.64_{(0.14)}$	$\mathbf{-4.90}_{(0.04)}$	$-199.8_{(32.9)}$	$177.4_{(103.2)}$
FOCAL++ (parallel)	$12.90_{(0.02)}$	$-5.77_{(0.19)}$	$\mathbf{-166.7}_{(16.9)}$	$\mathbf{281.9}_{(182.1)}$
FOCAL++ (series)	$\mathbf{13.02}_{(0.30)}$	$-5.49_{(0.47)}$	-	$278.5_{(161.7)}$

Table 5. Extension of Table 1 in the main text. Average testing return (standard deviation in parenthesis) of FOCAL and variants of FOCAL++.

Environment	Algorithm	RMS	ESR
Sparse-Point-Robot	FOCAL	1.283	0.226
	FOCAL++	**1.288**	**0.281**
Point-Robot-Wind	FOCAL	0.907	0.203
	FOCAL++	**1.060**	**0.254**
Sparse-Ant-Fwd-Back	FOCAL	0.758	0.5
	FOCAL++	**0.763**	**0.5**

Table 6. Extension of Table 2 in the main text. Embedding Statistics

[4]Both offline context-based meta-RL methods, introduced in https://ui.adsabs.harvard.edu/abs/2019arXiv190911373L/abstract

D. Experimental Details

D.1. Description of the Meta Environments

- Sparse-Cheetah-Vel: Half-Cheetah-Vel with customized sparse reward. Control a Cheetah robot to achieve a target velocity running forward. For transitions utilized in context encoder training, the reward is non-zero only if the velocity is within a goal radius to the target velocity. The sparse reward as a continuous function is expressed as follows

$$\text{sparse reward} = \begin{cases} \frac{\text{reward} - \text{goal radius}}{|\text{goal radius}|}, & \text{if reward} > \text{goal radius} \\ 0, & \text{otherwise} \end{cases}$$

  For Sparse-Cheetah-Vel, we set goal radius to be -0.1 to control the ratio of non-sparse transitions to be around 50% when training context encoder.

- Sparse-Ant-Fwd-Back (Sparse-Ant-Dir): Control an Ant robot to move forward or backward with the same sparsification scheme as for Sparse-Cheetah-Vel. For Sparse-Ant-Fwd-Back, we set goal radius to be 3 so that non-sparse transitions are about 50% of all the transitions.

D.2. Hyperparameter Settings

Hyperparameters	Sparse-Point-Robot	Point-Robot-Wind	Sparse-Cheetah-Vel	Sparse-Ant-Fwd-Back
reward scale	100	100	5	5
DML loss weight(β)	1	1	1	1
contrastive T	0.5	0.5	0.5	0.5
contrastive m	0.9	0.9	0.9	0.9
behavior regularization strength(α)	0	0	500	1e6
buffer size (per task)	1e4	1e4	1e4	1e4
batch size (sac)	256	256	256	256
batch size (context encoder)	1024	1024	512	512
meta batch size	16	16	16	4
g_lr(f-divergence discriminator)	1e-4	1e-4	1e-4	1e-4
dml_lr(α_1)	1e-3	1e-3	3e-4	3e-4
actor_lr(α_2)	1e-3	1e-3	3e-4	3e-4
critic_lr(α_3)	1e-3	1e-3	3e-4	3e-4
discount factor	0.9	0.9	0.99	0.99
# training tasks	80	40	80	2
# testing tasks	20	10	20	2
goal radius	-0.06	N/A	-0.1	3
latent space dimension	5	5	20	20
transformer hidden size (context encoder)	128	128	128	128
# multihead (if enabled)	8	8	8	8
# reduction (batch attention)	16	16	16	16
transformer blocks (context encoder)	3	3	3	3
dropout (context encoder)	0.1	0.1	0.1	0.1
network width (others)	256	256	256	256
network depth (others)	3	3	3	3
maximum episode length	20	20	200	200
target divergence	N/A	N/A	0.05	0.1

Table 7. Hyperparameters used to produce Figure 8. Meta batch size refers to the number of tasks for computing the DML or contrastive loss at a time. Larger meta batch size leads to faster convergence but requires greater computational power.