Appendix

A Pseudocode of Saliency-Guided Features Decorrelation

In Algorithm 1, we introduce the training process of SGFD. At each step, we apply the policy model π_ϕ to interact with training environments and sample a batch of transitions along with environment labels, as described in lines 3-5. Before the sample reweighting, we assess the current classifier's accuracy. If the accuracy exceeds 0.9, we proceed with sample reweighting, as described in lines 6-9. In contrast, if the accuracy is lower and $t > 1\mathrm{e}5$, we update the classifier based on the cross-entropy loss, as described in lines 9-13. In experiments, the classifier could be quickly converged. Based on the converged classifier, SGFD uses Equation (7) to learn a set of sample weights, as described in lines 14-17. This process also converged quickly, as only 128 parameters (the batch size) needed to be updated. Finally, SGFD uses Equation (1) and Equation (8) to update state-action value function $\mathcal{Q}_\theta$ and policy model π_ϕ based on the weighted data, as described in lines 18-20.

Algorithm 1 Saliency-Guided Features Decorrelation

1: **Input:** Initialize a policy model π_ϕ, a state-action value function $\mathcal{Q}_\theta$, a classifier model f_ψ, and a set of training environments with environment labels $\{e^1, e^2, \ldots, e^K\}$, where e^k is a K-dimensional one-hot vector and the $k-$th component is 1
2: **Output:** The learnt policy π_{ϕ^*}
3: **for** $t = 1$ **to** T **do**
4: Applying the policy model π_ϕ to interact with training environments
5: Sampling a batch of transitions along with environment labels $(s_i, a_i, r_i, s_i', e_i^k) \sim \mathcal{D}$
6: **for** $iter = 1$ to 10 **do**
7: **if** $t < 1\mathrm{e}5$ **or** the accuracy of $f(s) > 0.9$ **then**
8: Break this cycle
9: **else**
10: Sampling a batch of transitions along with environment labels $(s_i, a_i, r_i, s_i', e_i^k) \sim \mathcal{D}$
11: Update the classifier model f_ψ by minimizing the following loss
$\mathcal{L}(\psi) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} e_{i,j}^k \log(f_\psi(s_i)_j)$,
where $e_{i,j}^k$ denotes the $j-$th component of vector e_i^k, $f_\psi(s_i)_j$ denotes the predicted probability of the $j-$th class
12: **end if**
13: **end for**
14: Initialize a set of sample weights that can be learned $\mathbf{w}$
15: **for** $iter = 1$ to 10 **do**
16: Update the sample weights by minimizing Equation (7)
17: **end for**
18: Update the state-action value function $\mathcal{Q}_\theta$ by minimizing Equation (1)
19: Update π_ϕ by minimizing Equation (8)
20: **end for**

B Implement Details

Experimental Setting Details. For task-irrelevant variations, we follow the prior work [28] to sample several images from Kinetics [21]. In training, we create four parallel environments with different image backgrounds. In testing, we create a new environment with an unseen image background. For task-relevant variations, we follow the prior work [48] to revise the robot configures. The controllable configuration contains 10 different settings for each task, e.g., the length of the robot torso. In training, we create four parallel environments with different robot configures, where the values of configures are sampled from a predefined distribution, i.e., a uniform distribution between 1 and 5. In testing, we exhibit evaluations under two setups: interpolation and extrapolation. In the interpolation setup, task-relevant features changed in a manner interpolated between those in the training environments, whereas in the extrapolation setup, changes exceeded the range of those in the training environments, i.e., the tenth setting with the farthest training distribution. The visualizations are illustrated in Figure 7.

Table 3: Architectures of SGFD.

	Hyperparameter	Value
Encoder architecture	Convolutional layers	4
	Latent representation dimension	50
	Image size	(84,84)
	Stacked frames	3
	kernel size	3×3
	Channels	3
	stride	2 for the first layer, 4 otherswise
	Activation	ReLU
Policy and value function	MLP layers	2
	Hidden dimension	1024
	Activation	ReLU
Classifier architecture	MLP layers	2
	Hidden dimension	128
	Activation	ReLU

Table 4: Hyperparameters of SGFD.

	Hyperparameter	Value
Training for policy	Optimizer	Adam
	Learning rate	1e-3
	Action repeat	2
	Replay buffer capacity	1000000
	Batch size	128
	Target soft-update rate (τ)	0.01
	Actor update frequency	2
Training for sample reweighting	Optimizer	SGD
	momentum	0.9
	Learning rate	1e-2
	weight decay	1e-4
	The number of RFFs m	5

For each experiment, we report the mean and standard deviation over 10 random seeds.

Compute. Experiments are carried out on NVIDIA GeForce RTX 3090 GPUs and NVIDIA A10 GPUs. Besides, the CPU type is Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz. Each run of the experiments spanned about 12-24 hours, depending on the algorithm and the complexity of the task.

Architectures. Following AMBS [10], we use the same encoder architecture, which consists of 4 convolutional layers. The encoder weights are shared between the policy and the state-action value function. Each convolutional layer has a 3×3 kernel size and 32 channels. The first layer has a stride of 2, and all other layer has a stride of 1. There is a ReLU activation between each convolutional layer. The convolutional layers are followed by a trunk network with a linear layer, layer normalization, and finally, a tanh activation. As we focus on the sample reweighting, the encoder is updated by referring to the AMBS method.

The policy π_ϕ and the state-action value function $\mathcal{Q}_\theta$ are both 2-layer MLPs with a hidden dimension of 1024. We apply ReLU activations after each layer except the last layer. The classifier is implemented with 2-layer MLPs with a hidden dimension of 128. We also apply ReLU activations after each layer except the last layer. Table 3 shows the values of the architectures for the encoder, the policy, the value function, and the classifier.

Hyperparameters. Table 4 shows the hyperparameters used in our SGFD model.

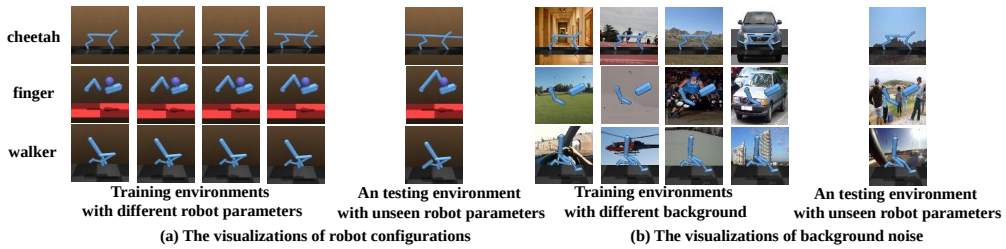

Figure 7: The visualization of training environments and testing environments.

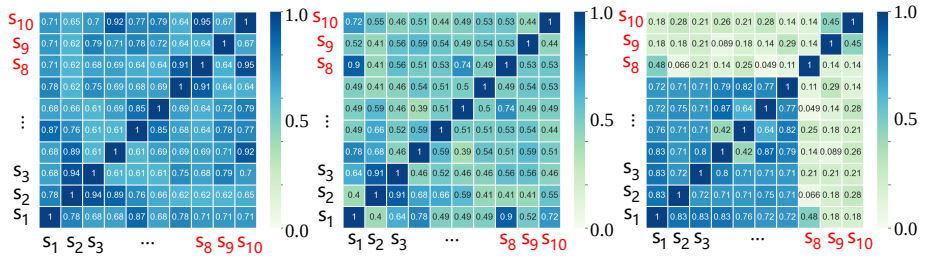

Figure 8: Pearson correlation coefficients among variables: a) on raw data, b) on weighted data, and c) on saliency-guided weighted data. The variables S_8, S_9, and S_{10} are the ones that have been changed among the environments.

C Additional Experimental Results

C.1 Additional Correlation Studies on the Weighted Samples

To further evaluate the decorrelation capability of our model, we implement the 'Pushing' task in Causal World and allow the model to access state features that have physical meaning. We record the Pearson correlation coefficients among features under three conditions: a) on raw data, b) on weighted data, and c) on saliency-guided weighted data (SGFD). Among these features, we designate the S_8, S_9, and S_{10}, which represent the length, width, and height, as the features of interest for environmental variations. Therefore, when the model is required to push an object with a new size, the correlation between size and other features needs to be carefully considered. As shown in Figure 8, due to limited samples and strong correlation between features, the effectiveness of directly decorrelating all features is greatly reduced. In contrast, SGFD is capable of effectively reducing the correlation between the size features (S_8, S_9, and S_{10}) and other features.

C.2 Additional Ablation Studies

We performed additional ablation experiments to test the effect of RFF and the saliency-guided model. As depicted in Figure 9 (a) and Figure 9 (b), when RFF is removed, the generalization performance drops significantly. In contrast, the global decorrelation can cause over-adjusting of samples weighting and impair the performance, as demonstrated by the reward curves in Figure 9 (c). Due to the need to eliminate correlations between variables using limited samples, the model may assign very low weights to certain training samples. However, this can potentially harm the efficiency of training. Thanks to the saliency-guided model, SGFD balances both efficiency and decorrelation.

C.3 Evaluation on the Classification Model

We recorded the converge curve of the classifier during training. The classifier starts to update when the replay buffer collects 1e5 steps. To avoid overfitting, we stopped to update the classifier when the accuracy exceeded 0.9. As shown in Figure 10, the classifier converged quickly and achieved the accuracy of 0.9, which means that the classifier does not take up many computing resources.

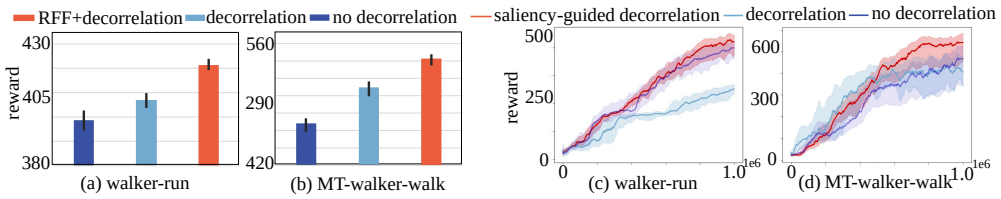

Figure 9: Ablation studies. (a)-(b) The performance difference of ablation studies on the RFF. (c)-(d) The performance difference of ablation studies on the saliency-guided model. The x-axis denotes the steps in the training environments. The y-axis denotes the cumulative reward recorded in the testing environment with background noises (walker-run) or varied robot parameters (MT-walker-walk).

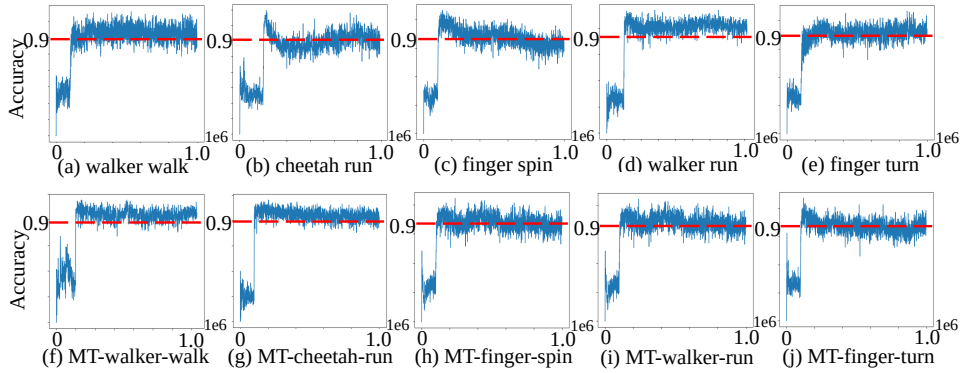

Figure 10: The accuracy of the classifier during training. The x-axis denotes the number of training steps taken in training environments, while the y-axis indicates the accuracy. The classifier starts to update when the replay buffer collects 1e5 steps. The red line corresponds to the 0.9 threshold, which indicates that the classifier stops updating if its accuracy exceeds 0.9.

C.4 Saliency Maps on the Policy Models

An intuitive approach to explaining visually-based models involves identifying pixels that have a significant impact on the final decision [1]. To determine whether the model focuses on the object or the background during decisions, we visualize the gradient of the action with respect to the input pixels. Experiments are conducted in environments with varying backgrounds and robot parameters, as depicted in Figure 11. Saliency maps of the baseline models reveal that various backgrounds draw considerable attention from the policy model while failing to make decisive contributions to our model. Moreover, SGFD can notice the changing aspects of the robot's body that other baseline models overlook. As a result, the saliency maps demonstrate that the decorrelation benefits the model's generalization in both task-relevant and task-irrelevant situations.

C.5 Additional Visualization of the Weighted Samples

In this experiment, we significantly expanded the dataset to include 300 state instances, which is substantially larger than the 20 states mentioned in Section 5.3. The identified states were also sorted into three groups in each environment, each consisting of multiple similar state instances. Moreover, we carefully analyzed the distribution of these states within each group, and the corresponding proportions are visually depicted in Figure 12. As illustrated in Figure 12 (a), the green dotted boxes indicate that 50% of the states in environment A and 35.4% of the states in environment B exhibited similarities. After the reweighting process, both environments A and B had approximately 42% similar states, effectively neutralizing the correlation between background and body posture. A similar outcome was observed in Figure 12 (b), wherein the reweighting process endeavored to balance the number of states in each group across environments with varied body lengths.

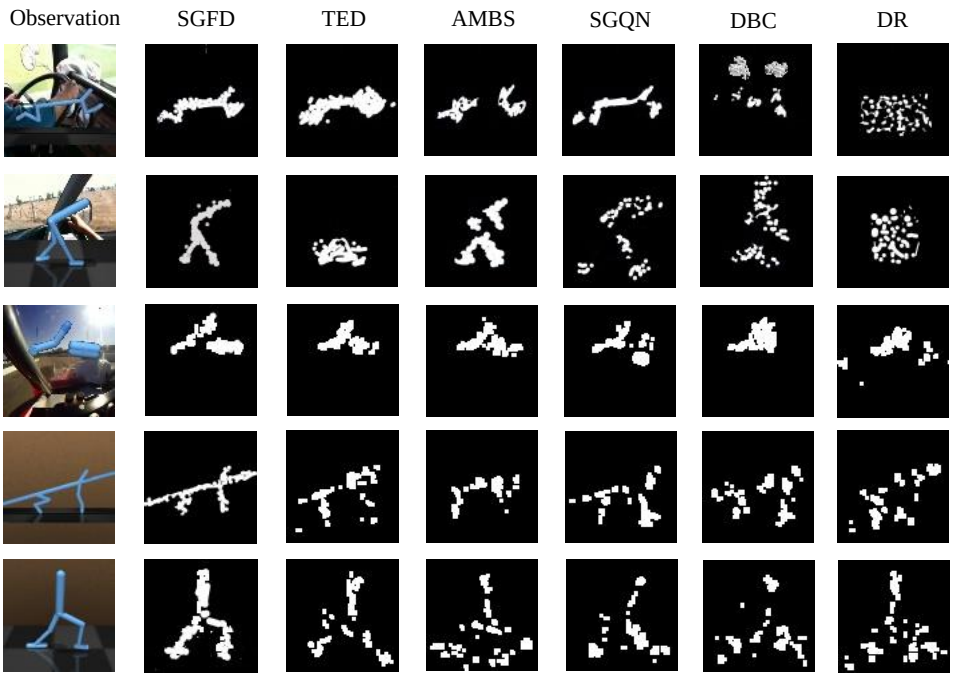

Figure 11: Visualization of saliency maps produced by the SGFD and other baselines when the robot with different backgrounds and robot parameters. The brighter the pixel is, the more contributions it makes to the decision.

C.6 Evaluation on the Generalization of SGFD

Evaluation on Changed Task-Irrelevant Features. We report the training curves of SGFD in environments with different task-irrelevant features, where the end points of the curves correspond to the results in Table 1. As illustrated in Figure 13, our model achieved superior performance compared to other baselines in previously unseen background environments. As the task becomes difficult (Figure 13 (b) and Figure 13 (d)), the advantage of SGFD achieves more significance. The results demonstrate that the decorrelation is suitable for RL tasks with different task-irrelevant features, and SGFD exhibits strong generalization capabilities to new values of task-irrelevant features.

Evaluation on Changed Task-Relevant Features. We also report the training curves of SGFD in environments with different task-relevant features, where the end points of the curves correspond to the results in Table 2. As shown in Figure 14, the reward curve exhibited a consistent positive correlation between the number of training steps and the average reward in the testing environment under both interpolation and extrapolation setups. Generally speaking, the extrapolation setting is more challenging for generalization because it requires the model to fully understand how changed features affect optimal decisions. Benefiting from decorrelation, our model achieved superior performance compared to other baselines, where the gaps were more significant under the extrapolation setting, as illustrated in Figure 14 (f). Our results demonstrate that the decorrelation facilitates our model's understanding of the true associations between changed features and decisions, leading to strong generalization on new values of task-relevant features.

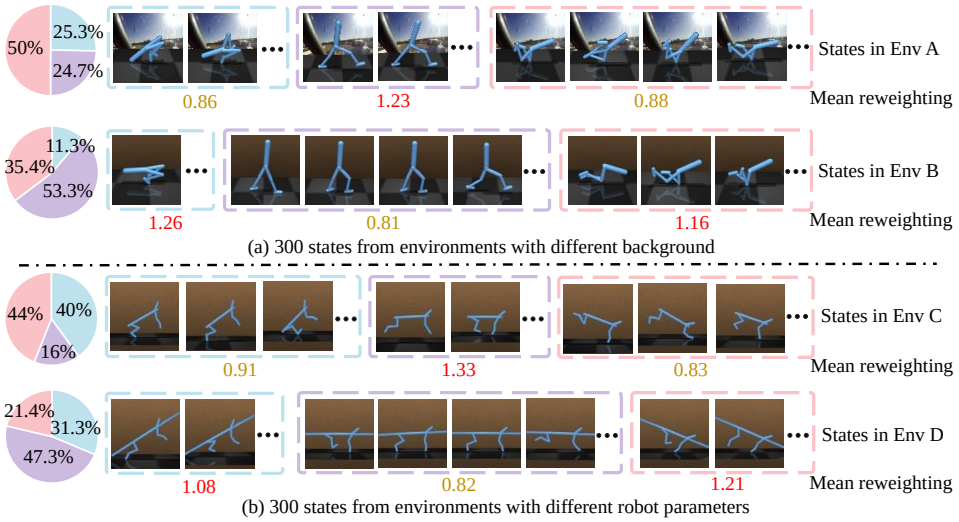

Figure 12: Visualization of weighted samples obtained from different environments with background noises and robot parameters. States that belong to the same dotted box with a specific color are considered semantically similar. SGFD balances the weights of samples to eliminate the correlations between changed features and other features. As shown in (a), the walker agents in Env A and Env B have different backgrounds, entangled with different distributions of the body posture. SGFD reduces the weighting of states with backgrounds and enhances the weighting of states without backgrounds, effectively neutralizing the correlation between background and body posture. A comparable result was observed in (b), where the reweighting process aimed to equalize the number of states in each group across environments with different body lengths.

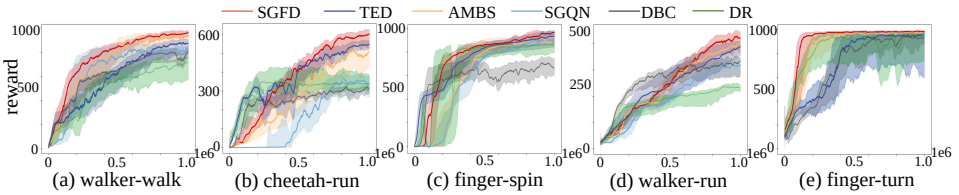

Figure 13: Generalization to changed task-irrelevant features. The x-axis denotes the number of training steps taken in training environments, while the y-axis indicates the average reward in the testing environment under different backgrounds. The reward curve demonstrates that our SGFD model generalizes well to new values of task-irrelevant features.

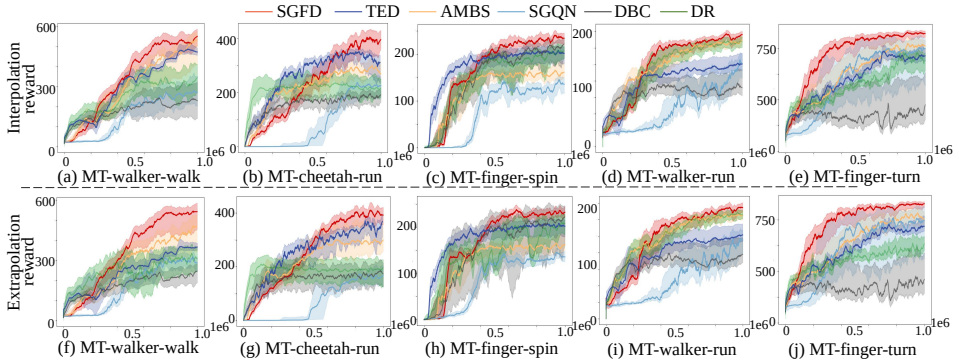

Figure 14: Generalization to changed task-relevant features. The x-axis denotes the number of training steps taken in training environments, while the y-axis represents the average reward in the testing environment with different robot parameters. We consider two setups for evaluation: an interpolation setup ((a)-(e)) and an extrapolation setup ((f)-(j)), where the changes in the task-relevant features are interpolations and extrapolations between the changes in the training environments, respectively. The reward curve shows that our SGFD generalizes well to the new values of task-relevant features.