PFRL: Pose-Free Reinforcement Learning for 6D Pose Estimation - Supplementary Material

1. Detailed Explanation of Pose Transformation

As stated in the main paper (Section 3.1), the key idea is to disentangle the rotation and translation, and further discretize each individual degree of freedom for the actions. At each step, for each of the rotation and translation, we provide 13 actions to rotate (or translate) the object along an axis in a directional small or large step, or stay still (See the Figure 1). The discretization converts the pose regression to a classification task, which tremendously reduces the training difficulty.

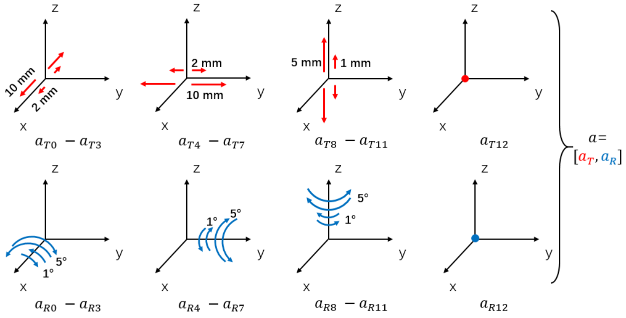

Figure 1. Illustration of pose transformations.

2. Additional Evaluation of the Refined Poses

Table 1 shows the additional evaluation of the refined poses with AAE initialization on the whole LINEMOD test set. The average depth error is 1.66cm, a bit larger than error on axis x and y. For rotation the error on elevation and in-plane tilt is smaller than on azimuth. We observe that the in-plane rotation exhibits more significant changes in 2D masks, which may explain the higher accuracy on this DoF.

3. Results with DPOD [5] as initial pose

Instead of AAE, we utilize PFRL to refine DPOD-syn's [5] initial poses provided by the author to make a fair comparison. The results of recall scores on ADD are shown in Table 2. When using the same initial poses, our method performs generally better than DPOD-syn.

		Translation			Rotation		
		x	y	z	Rx	Ry	Rz
	Mean(cm, °)	0.21	0.21	1.66	12.55	4.92	6.37
	Std(cm, °)	0.31	0.29	1.95	30.42	8.57	18.52
Acc (%)	2(cm, °)	99.53	99.67	72.72	33.06	41.64	56.66
	5(cm, °)	99.99	99.99	94.49	59.47	70.36	77.99
	10(cm, °)	100.00	100.00	99.25	77.80	88.35	88.00

Table 1. Accuracy of 6 degrees of freedom. Rx, Ry, Rz represent azimuth, elevation, and in-plane rotation respectively.

Metric	Recall scores (%) on ADD						
Class	ape	bv.	cam	can	cat	driller	duck
DPOD-refine	52.12	64.67	22.23	77.51	56.49	65.23	49.04
PFRL	69.26	78.68	27.77	77.16	64.52	79.90	48.24
Class	egg.	glue	hol.	iron	lamp	phone	**Mean**
DPOD-refine	62.21	38.94	25.55	98.43	58.35	33.79	54.20
PFRL	67.68	37.73	27.87	88.00	73.67	37.51	**59.85**

Table 2. DPOD-refine/PFRL with DPOD init.

4. Generalization Ability

To test our method's generalization ability, we evaluate the model on testing objects different from the training object on T-LESS dataset. Specifically, we trained 3 models on object 19-21 and test them on object 6-10 with AAE initialization separately. As shown in Table 3, the three models all improve the recall of VSD on 5 unseen objects for about 16%-18%, which shows our method can well generalize to unseen objects.

5. Class-Agnostic Training

The class-specific training setting in the original manuscript was adopted considering that the RL training is quite time-consuming and difficult to converge, especially in the case of multiple objects. We conducted class-agnostic training with the same network structure on LineMOD dataset, in which we just replaced the training data from

Metric	Recall scores (%) on VSD								
Test Obj		6			7			8	
Train Obj	19	20	21	19	20	21	19	20	21
AAE		52.3			36.6			22.1	
+PFRL	59.1	62.5	56.9	51.0	52.1	51.2	42.1	42.3	42.7
Test Obj		9			10			**Mean**	
Train Obj	19	20	21	19	20	21	19	20	21
AAE		46.5			14.3			34.3	
+PFRL	54.9	56.9	54.4	47.0	50.5	46.9	50.8	52.9	50.4

Table 3. Recall of VSD on T-LESS objects 6-11 for the model trained on object 19-21.

one object to all 13 objects. As shown in Table 4, although the class-agnostic training result can not compare with the class-specific training, it can still bring appreciable improvement to the initial poses.

Metric	ADD(%)						
Class	ape	bv.	cam	can	cat	driller	duck
AAE	3.96	20.92	30.47	35.87	17.90	23.99	4.86
Cls Agnos	22.00	56.26	14.02	53.84	44.21	43.41	32.68
Class	egg.	glue	hol.	iron	lamp	phone	**Mean**
AAE	81.01	45.49	17.60	32.03	60.47	33.79	31.41
Cls Agnos	87.51	63.22	25.78	55.87	91.17	38.43	48.34

Table 4. Class agnostic training results with AAE initialization.

6. Details of Optimization Rules

6.1. On Policy Part

We employ the proximal policy optimization algorithm [3] as the basic update rule. Let the 6D pose estimation procedure with one RGB image have K frames in total. At each time step k, let $\mathbf{s}_k$ denote the current state, then the relative SE(3) transformation $\mathbf{a}_k = [\mathbf{a}_R|\mathbf{a}_t]$ can be sampled from the network output distribution with input $\mathbf{s}_k$. π_θ denotes the current network output distribution, and $\pi_{\theta_{\text{old}}}$ denotes the network output distribution when $\mathbf{a}_k$ was sampled. $V(\mathbf{s}_k) = \mathbb{E}_{\mathbf{a}_k, \mathbf{s}_{k+1} \ldots}[\sum_{l=0}^{\infty} \gamma^l r_{k+l}]$ is the value function, which is meant to be the expected cumulative reward under current state $\mathbf{s}_k$. V_θ denotes V estimated by another network with input $\mathbf{s}_k$ that shares the same weights as the action network except for the last layer. The clipped surrogate objective can be written as:

$$L_c(\theta) = \hat{\mathbb{E}}[\min(\xi_k(\theta)\hat{A}_k, \text{clip}(\xi_k(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_k)], \tag{1}$$

where

$$\xi_k(\theta) = \frac{\pi_\theta(\mathbf{a}_k|\mathbf{s}_k)}{\pi_{\theta_{\text{old}}}(\mathbf{a}_k|\mathbf{s}_k)}. \tag{2}$$

The $\hat{A}_k$ in Eq.(1) is the advantage estimator defined as: $\hat{A}_k = -V(\mathbf{s}_k) + r_k + \gamma r_{k+1} + ... + \gamma^{K-k+1} r_{K-1} + \gamma^{K-k} V(\mathbf{s}_K)$. The value loss can be written as:

$$L_v(\theta) = (V_\theta(\mathbf{s}_k) - V_{\text{targ}})^2, \tag{3}$$

where

$$V_{\text{targ}} = r_k + \gamma r_{k+1} + ... + \gamma^{K-k} V_{\theta_{\text{old}}}(\mathbf{s}_K). \tag{4}$$

And the entropy regularization term to encourage adequate exploration can be written as:

$$L_e(\theta) = \pi_\theta \log \pi_\theta. \tag{5}$$

The on-policy update loss L_{on} can be written as:

$$L_{\text{on}} = L_c + \lambda_v L_v + \lambda_e L_e. \tag{6}$$

6.2. Off Policy Part

We introduce the V-trace target from [1] to use samples more efficiently with an off-policy update and give value function a more accurate estimation. The n-step V-trace target can be written as:

$$V_{\text{trace}} = V(\mathbf{s}_k) + \sum_{q=k}^{k+n-1} \gamma^{q-k} (\prod_{i=k}^{q-1} c_i) \delta_q V, \tag{7}$$

where

$$\begin{aligned} \delta_q V &= \rho_q(r_q + \gamma V(\mathbf{s}_{q+1}) - V(\mathbf{s}_q)), \\ \rho_q &= \min(\overline{\rho}, \frac{\pi_\theta(\mathbf{a}_q|\mathbf{s}_q)}{\pi_{\theta_{\text{old}}}(\mathbf{a}_q|\mathbf{s}_q)}), \\ c_i &= \min(\overline{c}, \frac{\pi_\theta(\mathbf{a}_i|\mathbf{s}_i)}{\pi_{\theta_{\text{old}}}(\mathbf{a}_i|\mathbf{s}_i)}). \end{aligned} \tag{8}$$

In Eq.(8), ρ_q and c_i are truncated importance sampling weights, and the truncated parameters $\overline{\rho} \geq \overline{c}$. The off-policy value function loss is:

$$L_{\text{off}}(\theta) = (V_\theta(\mathbf{s}_k) - V_{\text{trace}})^2. \tag{9}$$

7. Results on T-LESS

Fig. 2 shows some qualitative results of objects 19-23 on T-LESS dataset [2].

References

[1] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1406–1415, 2018.

[2] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017.