PoCo: Policy Composition from and for Heterogeneous Robot Learning Paper Review

Paper Review by Tyler Kim

Summary

In Lirui Wang et al's *PoCo*: *Policy Composition from and for Heterogenous Robot Learning*, the researchers describe an approach to train a robot to perform various actions using data that is derived from different sources and modalities. Current robot learning pipelines trained specialized models for a single robot for a single task without any behavior. Additionally, training on data that are inherently from different distributions and sensors proved difficult. Therefore, the authors propose a framework called Policy Composition (PoCo) which aims to compose data derived from different modalities, sources, and tasks. **The main contribution of the paper are PoCo, the framework that uses diffusion models for combine data from different domains and modalities, developing task-, behavior-, and domain-level composition for creating policies without retraining, and illustrating scene- and task-level generalization of PoCo across simulation and real-world settings.**

Policy Composition (PoCo) uses a diffusion model to represent a policy where a trajectory is generated given a history of observation denoted as $\pi(\tau|o)$ using $L_{MSE} = \left|\left|\epsilon^t - \epsilon_\theta(\tau_0 + \epsilon^t, t \mid o)\right|\right|^2$ as the loss function. The general process of PoCo to input the heterogenous data has multiple steps. Modalities are defined as $M \in \{M_{tactile}, M_{pointcloud}, M_{image}, M_{proprioceptive}\}$ where each modality is defined as information from a particular sensor. Next, data domains D, which could be simulations, real-world robots, and human demonstrations, are considered as long as they share the same action space. Then, constraints are considered on a desired behavior via a cost function which in the paper are $c_{smoothness}(\tau)$ and $c_{safety}(\tau)$. Finally, a robot task T is specified via a natural language command. A separate probabilistic model is learned on tuple (M, D, T, c).

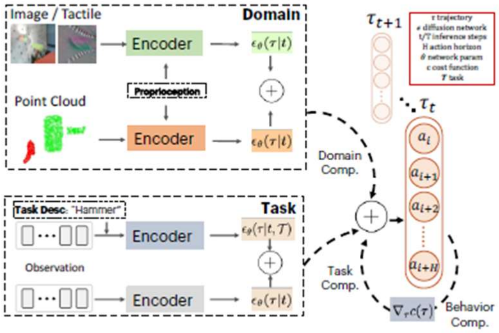
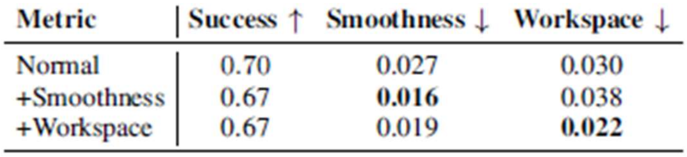


Fig. 3: **Illustration of Policy Composition.** Policies are composed at test time by combining gradient predictions. This can apply to *domain composition* by combining policies that train with different modalities such as image, point cloud, and tactile images. It can also be used with different tasks in *task composition* and additional cost functions for desired behavior with *behavior composition*. The only assumption is that the diffused outputs for each model need to be in the same space, i.e. the action dimension and action horizon. We denote the composition operator using a "plus" sign.

The paper proposes to sample from the product of different distributions in hopes to combine the score predictions from diffusion policies at inference time. The product of the composition will yield a trajectory that will most likely accomplish the task under multiple distributions. However, assumptions must be made: mutual dependence of tasks T and costs c and conditional independence of tasks T and costs c given a trajectory τ. The implementation is represented as energy-based models.

The experiments were divided into two types: simulated and real-world. During the simulation experiments, the researchers reported gains in smoothness and safety constraints adding prior information when composing behaviors. They also report that task composition performs the best in multitask policy evaluation. Finally, they found that their approach can generalize across multiple distractors.



Metric	Success ↑	Smoothness ↓	Workspace ↓
Normal	0.70	0.027	0.030
+Smoothness	0.67	**0.016**	0.038
+Workspace	0.67	0.019	**0.022**

TABLE I: **Effect of Behavior Composition.** By combining costs probabilistically, we can optimize metrics from each cost objective.

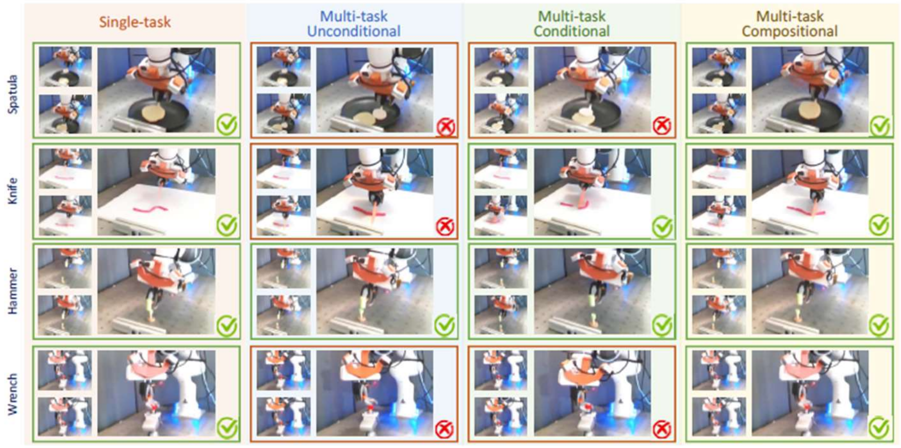


Fig. 6: **Task-level Qualitative Results.** By composing unconditional and task-conditional models, our composed policy can accomplish a diverse set of tasks and outperform unconditioned multitask policies.

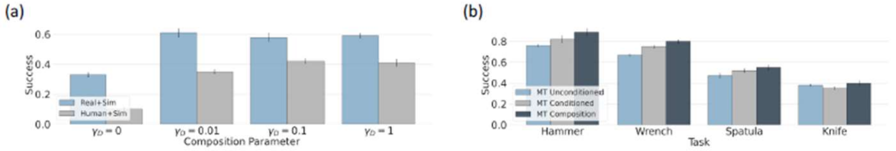


Fig. 7: **Effect of Policy Composition in Simulation.** **(a)** Task Composition performs the best in multitask policy evaluation in simulation. **(b)** Simulation policies help with composition across domains and evaluation in simulation.

For the real-world experiments, the researchers found that policy composition improves success rate across different scenes on four generalization axes: varying object poses, varying robot initial pose, varying tool poses, adding distractor objects, and replacing objects with novel instances from the same class. The researchers also found that multitask policies can perform on par with task-specific policies, stable in dexterous tasks, classifier-free training performs better than naively concatenating features, and composition hyperparameters must stay within a range to remain effective.

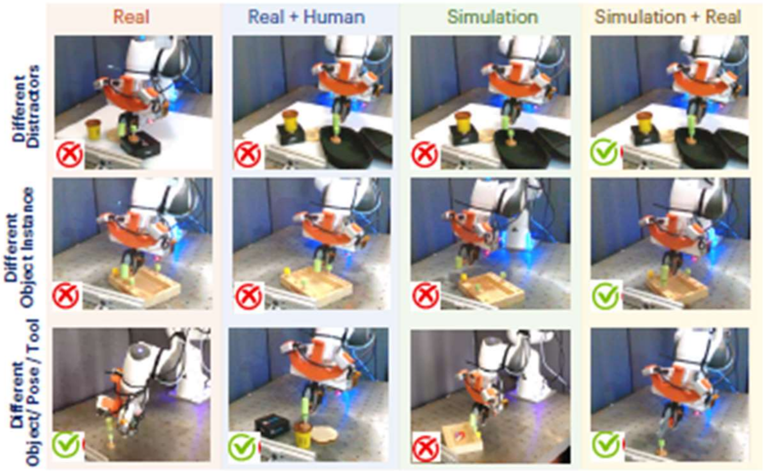
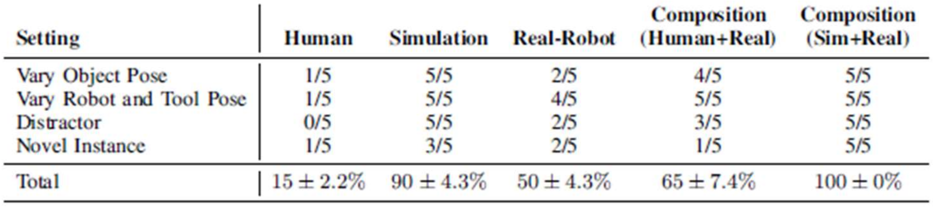
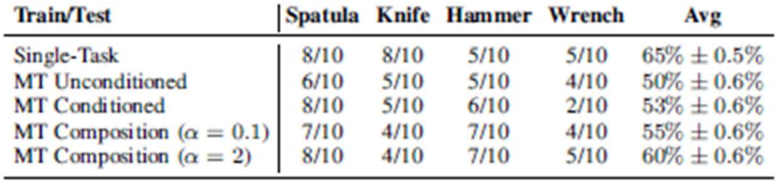


Fig. 9: **Domain Composition Comparison.** By composing policies trained in simulation, human, and real data, our approach can generalize across multiple distractors (first row), with varying object and tool poses (second row), and across new object and tool instances (third row).



Setting	Human	Simulation	Real-Robot	Composition (Human+Real)	Composition (Sim+Real)
Vary Object Pose	1/5	5/5	2/5	4/5	5/5
Vary Robot and Tool Pose	1/5	5/5	4/5	5/5	5/5
Distractor	0/5	5/5	2/5	3/5	5/5
Novel Instance	1/5	3/5	2/5	1/5	5/5
Total	$15 \pm 2.2\%$	$90 \pm 4.3\%$	$50 \pm 4.3\%$	$65 \pm 7.4\%$	$100 \pm 0\%$

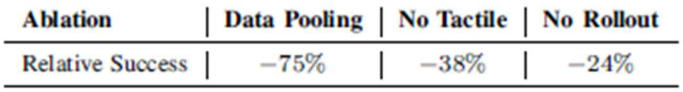
TABLE II: **Quantitative Results on Domain Composition.** Policy Composition improves average success rates compared to individual constituent policy across different scenes on four separate generalization axes, evaluated on the hammering task.



Train/Test	Spatula	Knife	Hammer	Wrench	Avg
Single-Task	8/10	8/10	5/10	5/10	$65\% \pm 0.5\%$
MT Unconditioned	6/10	5/10	5/10	4/10	$50\% \pm 0.6\%$
MT Conditioned	8/10	5/10	6/10	2/10	$53\% \pm 0.6\%$
MT Composition ($\alpha = 0.1$)	7/10	4/10	7/10	4/10	$55\% \pm 0.6\%$
MT Composition ($\alpha = 2$)	8/10	4/10	7/10	5/10	$60\% \pm 0.6\%$

TABLE III: **Policy Performance on Different Tool-use Tasks.** We compare among different ways to handle multitask (MT) diffusion policy training, and find that task composition overall leads to improved performance across tool-use tasks.

In the ablation study, researchers reported that a naïve approach results in a 75% performance drop.



Ablation	Data Pooling	No Tactile	No Rollout
Relative Success	−75%	−38%	−24%

TABLE IV: **Ablation of Real World Execution.** We analyze the effect of composition across modalities (data pooling), the use of tactile signals (no tactile) and the effect of predicting open-loop trajectory rollouts (no rollout) in the real world. The numbers represent the relative scale of the ablated method compared to the default method that involves policy composition from simulation and real world, tactile feedback, and closed-loop predictions.

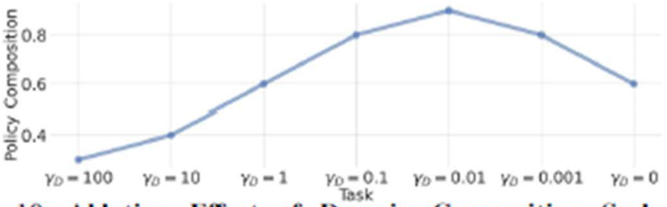


Fig. 10: **Ablation Effect of Domain Composition Scale.** We illustrate the effect of performance on the hammering task as we increase the composition coefficient between policies.

Strengths

The paper provides a new approach to handle heterogeneous data via policy composition. More importantly, it opens the door to further research in a more generalized robot particularly with more sensors and various data distributions. More interestingly, composing policies such that a trajectory could be sampled in a way that has the highest likelihood among multiple distributions. One thing I learned from this paper was a possible approach to handle heterogeneous data for a unified goal. More specifically, I learned that simply multiplying distributions together could serve as a feasible way to create a common distribution among multiple modalities. In fact, I thought it was a very clever way to approach the problem. The PoCo framework in general, I thought was very novel in the realms of robotics.

Potential Improvements

One thing I think the paper could improve upon to get closer to its stated goal/contributions to generalize PoCo more. Unfortunately, policy composition works only under certain assumptions as stated in the paper, that is, the action space must be the same. Another improvement that I think the paper could improve is handling the time consumption off the models. Since a separate diffusion model is trained for various tuples, this would mean parallelism is extremely important or else the framework would take too long to train/infer. The paper seems to be using the latest techniques or models but one thing they could do is have some sort of memory compression system as seen in LRLL. This would help reduce memory use significantly. In addition, using primitive abilities to learn new skills as an addition to PoCo would take advantage of the versatility of multiple data distributions and modalities.

Extensions

There are a couple of extensions or follow-ups I could think of for this paper. One possible extension would be creating a more generalized diffusion model for the different modality, domain, task, and behavioral cost tuples instead of having a separate one for each instance. Another extension would be to try a new approach to compose data from varying distributions and modalities such that it does not rely on the assumption that the action space must be the same. The final extension would be to expand PoCo to a larger variety of tasks instead of just using a couple of tools.