Data Augmentation for Online Learning of Rope Manipulation

Peter Mitrano
University of Michigan
Email: pmitrano@umich.edu

Dmitry Berenson
University of Michigan
Email: dmitryb@umich.edu

***Abstract*—The success of deep learning depends on large datasets, but in deformable object manipulation collecting these datasets is time-consuming, and therefore learning from small datasets is an important open problem. *Data augmentation* is a common solution to this problem, but most existing methods focus only on computer vision tasks. We propose an optimization-based approach to data augmentation for manipulation of deformable objects. The proposed method solves for rigid body transformations to trajectories of geometric state and action data that maximize validity, relevance, and diversity objectives. We test our method on the previously studied problem of learning a classifier of model accuracy [1]. We evaluate on a bimanual rope manipulation task both in simulation and in the real world, and we find that training with augmentations significantly increases the task success rate.**

I. Introduction

While interest in applying deep learning to robotic manipulation has recently increased, the lack of cheap data has proven to be a significant limitation [2, 3]. This is especially problematic for deformable object manipulation, because small datasets will inevitably have poor coverage of the state space, and accurate simulation datasets are missing or incomplete. To enable applications such as smart and flexible manufacturing, logistics, and care-giving robots [4], we must improve methods that learn from small real world datasets.

One of the simplest and most effective ways to mitigate the problem of small datasets is to use data augmentation [5, 6, 7, 8, 9, 10, 11, 12]. Data augmentation is the technique of generating new examples by modifying existing ones. While data augmentation has been shown to significantly improve generalization performance in tasks like image classification, these methods operate on different types of data and labels and are not applicable to many manipulation problems. Furthermore, most existing augmentation methods have one of two limitations: They are either restricted to operations which are valid on all examples [13, 12, 9, 8], or rely on training a generative model (VAE, GAN, etc.), which do not perform well on small datasets [14, 15, 5].

In our problem statement (Section II), we formalize data augmentation as an optimization problem based on three key criteria: *validity*, *relevance*, and *diversity*. We then design objective functions based on these ideas, specifically for manipulation. We do not claim that this formulation is useful for all manipulation problems, and we clearly define the physical assumptions behind this formulation in Section II.

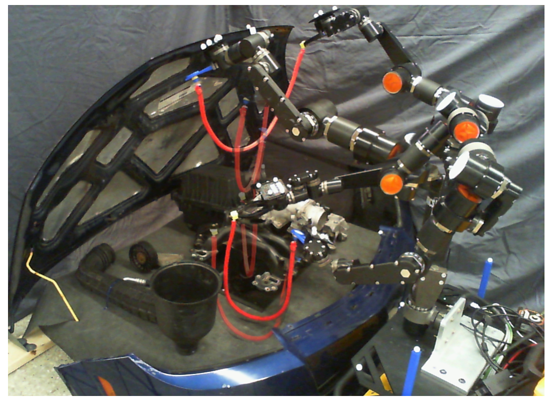

Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

Our contribution is a method that tackles this specialized augmentation problem. Our method operates on trajectories of object (including robot) poses and velocities, and our augmentations are rigid-body transformations applied to the moving objects in the scene. Our method encourages validity by preserving contacts and the influence of gravity. We encourage relevance by initializing the augmentations nearby the original examples and preserving motion near obstacles. Finally, we encourage diversity by pushing the augmentations towards uniformly randomly sampled targets.

We show that training on our augmentations increases the success rate on a bimanual rope manipulation task, both in simulation and in the real world (Figure 1). We test augmentation on the problem of learning a classifier of dynamics model error, as in [1]. Training is online, which means we augment data and fine-tune the network after each batch of data is collected. In the real robot experiment, augmentation increased the success rate from 27% to 50% in only 30 trials.

II. Problem Statement

In this section, we define the form of data augmentation studied in this paper. We define a dataset as a list of examples x and, optionally, labels y. Augmentation is a stochastic function $\tilde{x} \sim \phi(x)$ which takes an example and produces an augmented example. If the dataset contains labels y, we assume that the label is not changed by augmentation.

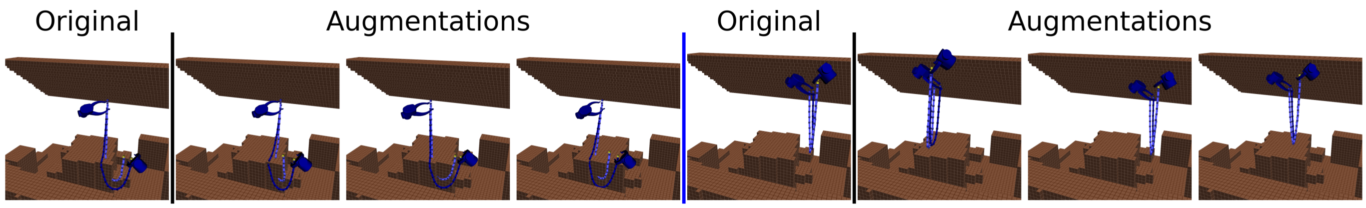

Fig. 2: Example augmentations generated by our method. The rope start (dark blue) and end (light blue) states are shown, plus the grippers at the start state. The environment is in brown, with the simplified engine block in the center. The left half shows three augmentations which preserve the original deformation on the engine block, and the right half shows three particularly diverse augmentations where the rope is in free space.

A key insight is that objects in the scene can be categorized as either robots, moved objects, or stationary objects, and that these should be considered differently in augmentation. Treating all moved objects as one category is one of the reasons why our method can handle scenes with many moving, possibly deformable, objects.

We denote the moved objects state as s, the robot state as r, the robot action as a, and the stationary objects as e (also called environment). Our method augments the moved object states, the robot state, and the actions, but not the stationary objects. Variables indexed by time are shown in boldface, and augmented state variables are shown with a tilde. With this, we can write augmentation as $\phi(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{a}, \boldsymbol{e}) \rightarrow \{\tilde{\boldsymbol{s}}, \tilde{\boldsymbol{r}}, \tilde{\boldsymbol{a}}, \boldsymbol{e}\}$. We formalize six objective functions which encourage validity, relevance, and diversity. The result is Problem (1):

$$\begin{aligned}
\min_{T} \quad & \mathcal{L}_{\mathbb{U}}(T, T^{\text{target}}) + \beta_1 \mathcal{L}_{\text{bbox}}(\tilde{\boldsymbol{s}}) + \beta_2 \mathcal{L}_{\text{valid}}(T) + \\
& \beta_3 \mathcal{L}_{\text{occ}}(\tilde{\boldsymbol{s}}, \boldsymbol{e}) + \beta_4 \mathcal{L}_{\Delta d^-}(\tilde{\boldsymbol{s}}, \boldsymbol{e}) + \\
& \mathcal{L}_{\text{robot}}(\tilde{\boldsymbol{s}}, \tilde{\boldsymbol{r}}, \tilde{\boldsymbol{a}}, \boldsymbol{e}) \qquad (1) \\
\text{s.t.} \quad & \{\tilde{\boldsymbol{s}}, \tilde{\boldsymbol{r}}, \tilde{\boldsymbol{a}}\} = \texttt{apply}(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{a}, T) \\
& T^{\text{target}} \sim \mathbb{U}[T^-, T^+]
\end{aligned}$$

The decision variable is the SE(3) transform T, which is applied in the `apply` function. We propose that diversity should be maximized by the transforms being uniformly distributed, and therefore $\mathcal{L}_{\mathbb{U}}$ penalizes the distance to a target transform T^{target} sampled uniformly within $[T^-, T^+]$. The magnitudes of different terms are balanced by $\beta_1, \beta_2, \beta_3, \beta_4$. In our experiments, we use $\beta_1 = 0.05, \beta_2 = 1, \beta_3 = 1, \beta_4 = 0.1$. We define the other terms in Section III-A, and describe how we solve this problem in Section III-B.

A. Assumptions

Our method for solving Problem (1) relies on the following assumptions:

- The geometry of the robot and all objects is known.
- The scene can be decomposed into objects which can be assigned or detected as either moving or stationary.
- Examples are time-series, consisting of at least two states.
- All possible contacts between stationary vs. moving objects have the same friction coefficient.
- Contacts between the robot and objects/environment (e.g. grasps) can be determined from the data.
- A rigid-body transformation of an object preserves internal forces (e.g. friction between fibers of a rope).
- Objects only move due to contact or under the force of gravity (e.g. we do not handle magnetism or wind).
- Class labels are preserved under augmentation.

Notably, the assumption that a rigid-body transformation preserves internal forces is what allows us to handle cluttered scenes with many moving objects, as well as deformable or articulated objects. While it could be valuable to augment the deformation or relative motion of the objects, doing so in a way that is valid would be challenging. Instead, we transform them all rigidly (See examples in Figure 2). Naturally, there are scenarios where these assumptions do not hold and thus where our algorithm may not perform well. However, our rope manipulation experiment demonstrates significant improvement, and we expect these assumptions extend to other scenarios.

III. Methods

In this section, we propose specific definitions for the objective terms in Problem (1) and explain how we solve it.

A. Objective Functions

1) Bounding Box Objective: The bounding-box objective is defined as $\mathcal{L}_{\text{bbox}} = \sum(\max(0, \tilde{\boldsymbol{s}} - s^+) + \max(0, s^- - \tilde{\boldsymbol{s}}))$, which keeps the augmented state $\tilde{\boldsymbol{s}}$ within the workspace/scene bounds defined by $[s^-, s^+]$ by summing the bounds violations over all state dimensions. The bounding box objective encourages relevance, since states outside the workspace are unlikely to be relevant for the task.

2) Transformation Validity Objective: The transformation validity objective $\mathcal{L}_{\text{valid}}$ assigns high cost to transformations that are always invalid or irrelevant, and is a function of only the transformation. For example, in our rope manipulation case, it is nearly always invalid to rotate the rope so that it floats sideways. This term can be chosen manually on a per-task basis, but we learn it from simulation data.

3) Occupancy Objective: The occupancy objective encourages validity by ensuring that the occupancy $O(p)$ of each point $\tilde{\boldsymbol{p}}_{s,i} \in \tilde{\boldsymbol{p}}_s$ in the augmented object state matches the occupancy of the corresponding original point $\boldsymbol{p}_{s,i} \in \boldsymbol{p}_s$. For this term, we directly define the gradient, which moves $\tilde{\boldsymbol{p}}_{s,i}$ in the correct direction when the occupancies do not match. We use the signed distance field (SDF) computed from the environment geometry. The objective is $\mathcal{L}_{\text{occ}} = \text{SDF}(\tilde{\boldsymbol{p}}_{s,i})\big(O(\boldsymbol{p}_{s,i}) - O(\tilde{\boldsymbol{p}}_{s,i})\big)$

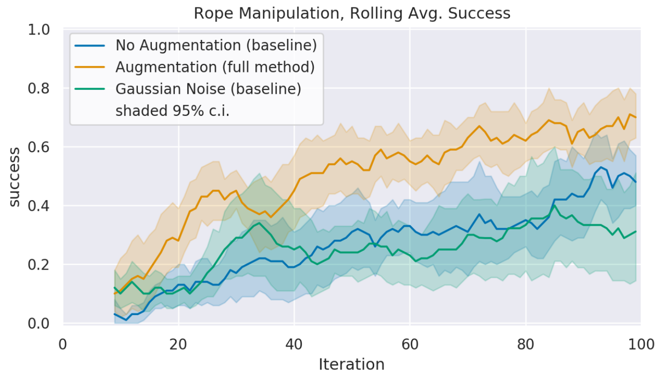

Fig. 3: The success rate on bimanual rope manipulation in simulation, using a moving window average of 10. Each iteration consists of planning, execution, and fine-tuning.

4) Delta Minimum Distance Objective: The delta minimum distance objective encourages relevance by preserving near-contact events. We preserve near-contact events because they may signify important parts of the task, such as being near a goal object or avoiding an obstacle. We define the point among the moved object points $\boldsymbol{p}_s$ which has the minimum distance to the environment $p_{d^-} = \text{argmin}_{\boldsymbol{p}_{s,i}} \text{SDF}(\boldsymbol{p}_{s,i})$. The corresponding point in the augmented example we call $\tilde{p}_{d^-}$. The objective is $\mathcal{L}_{\Delta d^-} = ||\text{SDF}(p_{d^-}) - \text{SDF}(\tilde{p}_{d^-})||_2$.

5) Robot Contact Objective: The robot contact objective $\mathcal{L}_{\text{robot}}$ encourages validity of the robot's state and the action. Any grasps or contacts involving the moved objects which existed in the original example must also exist in the augmented example. Let the contact points on the robot be p_r^c and the contact points on the moved objects' state be p_s^c. The objective is $\mathcal{L}_{\text{robot}} = \sum_i ||p_{r,i}^c - p_{s,i}^c||_2$

B. Solving the Augmentation Optimization Problem

We solve Problem (1) by splitting it into two parts.

a) Part 1: We optimize the transform T while considering all terms except $\mathcal{L}_{\text{robot}}$. This step alternates between stepping towards a uniformly randomly sampled target transform T^{target} to optimize $\mathcal{L}_{\mathbb{U}}$, and optimizing the other four terms with gradient descent.

b) Part 2: We optimize $\mathcal{L}_{\text{robot}}$. This corresponds to computing the augmented robot state $\tilde{\boldsymbol{r}}$ and action $\tilde{\boldsymbol{a}}$ given the augmented state information $\tilde{\boldsymbol{s}}$ and the environment $\boldsymbol{e}$. Minimizing $\mathcal{L}_{\text{robot}}$ means preserving the contacts the robot makes, which we do with inverse kinematics.

IV. Experiments and Results

Our experiments are designed to show that training on augmentations generated by our method improves performance on a downstream task. The task and methodology is based on prior work, and more details can be found in [1].

A. Bimanual Rope Manipulation

In this task, a bimanual robot holds the ends of a rope in a scene resembling the engine bay of a car. The rope is represented as 25 points. The complex deformation of the rope makes learning from a small dataset difficult, which motivates data augmentation. The robot plans trajectories from a single start to a single goal using an approximate model of the rope dynamics. These dynamics are inaccurate in some regions, and so the planner uses a learned classifier of model accuracy to avoid inaccurate transitions. The robot learns this classifier online, iteratively planning, executing, and collecting the data to update the classifier. We apply our data augmentation to the problem of learning this classifier.

In this experiment, 25 augmentations were produced for each original example, and examples are shown in Figure 2. The primary result is shown in Figure 3. Over the course of 100 iterations, the success of our method using augmentation is higher than the baseline of not using augmentation. Additionally, we include a baseline which adds independent Gaussian noise to each dimension of the state, robot, action, and environment data. The shaded regions show the 95th percentile over 10 runs. If we analyze the success rates averaged over the final 10 iterations, we find that without augmentation the success rate is 48%, but with augmentation the success rate is 70%. The Gaussian noise baseline has a final success rate of 31%. A one-sided T-test confirms statistical significance ($p < 0.001$).

B. Real Robot Results

In this section, we perform a similar experiment to the simulated bimanual rope manipulation experiment, but on real robot hardware. This demonstrates that our method is also effective on noisy sensor data. More importantly, it demonstrates how augmentation enables a robot to quickly learn a task in the real world.

We ran the rope classifier learning procedure with a single start configuration and a single goal region, both with and without augmentation. After 30 iterations of learning, we stop and evaluate the learned classifiers 26 times. With augmentation, the robot successfully placed the rope under the engine 13/26 times. Without augmentation, it succeeded 7/26 times. Videos showing learning progress and task execution can be found in the supplementary video.

V. Conclusion

This paper proposes a novel data augmentation method for trajectories of geometric state and action data. We argue that augmentations should be valid, relevant, and diverse, and use these to formalize augmentation as optimization. By leveraging optimization, augmentations are not limited to only operations valid on all examples. On the other hand, there are problems where the proposed objective functions do not ensure validity, relevance, and diversity. In these cases, new objective functions may be developed.

Our results show significantly better downstream task performance when training on small datasets (≈3k examples). In simulated bimanual rope manipulation, the success rate with augmentation is 70% compared to 47% without augmentation. We also perform the bimanual rope manipulation task in the real world, where the success rate improves from 27% to 50% with the addition of augmentation.

References

[1] Peter Mitrano, Dale McConachie, and Dmitry Berenson. Learning Where to Trust Unreliable Models in an Unstructured World for Deformable Object Manipulation. *Science Robotics*, 2021.

[2] Wojciech Zaremba. OpenAI Disbands Robotics. *Interview on YouTube*, 2021.

[3] Oliver Kroemer, Scott Niekum, and George Konidaris. A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms. *JMLR*, 2021.

[4] Johnny Wood. Robot workers are being hired at record rates in US companies - here's why. *World Economic Forum*, 2021.

[5] Sumeyra Demir, Krystof Mincev, Koen Kok, and Nikolaos G. Paterakis. Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting. *Applied Energy*, 2021.

[6] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A Survey of Data Augmentation Approaches for NLP. *Association for Computational Linguistics*, 2021.

[7] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6:60, 2019.

[8] P.Y. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. *International Conference on Document Analysis and Recognition*, 2003.

[9] Ekin Dogus Cubuk and Barret Zoph and Dandelion Mané and Vijay Vasudevan and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *ArXiv*, 2018.

[10] Brian Kenji Iwana and Seiichi Uchida. An Empirical Survey of Data Augmentation for Time Series Classification with Neural Networks. *PLOS One*, 2020.

[11] Albert Zhan, Philip Zhao, Lerrel Pinto, Pieter Abbeel, and Michael Laskin. A Framework for Efficient Robotic Manipulation. *NeurIPS (Deep RL workshop)*, 2021.

[12] Gregory W. Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Learning Invariances in Neural Networks from Training Data. *NeurIPS*, 2020.

[13] Yijiong Lin, Jiancong Huang, Matthieu Zimmer, Yisheng Guan, Juan Rojas, and Paul Weng. Invariant transform experience replay: Data augmentation for deep reinforcement learning. *Robotics and Automation Letters*, 2020.

[14] Toan Tran, Trung Pham, Gustavo Carneiro, Lyle J. Palmer, and Ian D. Reid. A Bayesian Data Augmentation Approach for Learning Deep Models. *NeurIPS*, 2017.

[15] Hiroshi Ohno. Auto-encoder-based generative models for data augmentation on regression problems. *Soft Computing*, 2020.