ESE 546 FINAL PROJECT: LEARNING AND PLANNING WITHIN A DEFORMABLE WORLD MODEL

IAN PEDROZA [IPEDROZA@SEAS],

ABSTRACT. We investigate approaches for learning world models that enable the manipulation of deformable objects from limited real-world data. Our methods consider both learned and pre-trained visual embeddings, coupled with a vision transformer-based dynamics model. The objective is to achieve effective deformable object shaping using as little as ten minutes of robot interaction data. By operating on latent representations rather than raw pixels, we show that our model can predict complex deformable dynamics and facilitate zero-shot planning for novel target shapes. Experiments in a differentiable plasticine simulator demonstrate that our world model can support shape formation tasks with improved sample efficiency.

1. INTRODUCTION

Enabling robots to effectively manipulate deformable objects remains a core challenge in robotics due to the high complexity and variability in material properties, deformation patterns, and partial observability. While model-based approaches have made progress for rigid body manipulation, extending these results to non-rigid settings is nontrivial. Common difficulties include the need for extensive, domain-specific data and the lack of analytical models that capture subtle deformation behaviors.

In this work, we explore a strategy to learn deformable object dynamics directly from vision-based observations. Drawing on recent advances in self-supervised representation learning [1, 2], we propose a world model that integrates powerful visual embeddings with a dynamics model for planning. Our approach focuses on visual patch embeddings, comparing pre-trained DINOv2 embeddings against learned embeddings initialized from scratch. We demonstrate that even under constrained data collection regimes (approximately ten minutes of real robot manipulation data), we can learn policies that manipulate plasticine-like objects into specified shapes, and find under preliminary tests that DINOv2 patch embeddings are not a more effective encoder for deformable settings than a learned encoding.

1.1. **Contributions.**

(1) We collect and process training episodes of deformable object interactions using a differentiable deformable object simulator (Plasticine Lab).
(2) We implement and train a vision-transformer-based dynamics model that, given encoded visual states and gripper actions, predicts the next latent representation of the deformable scene.
(3) We compare learned patch embeddings against pre-trained DINOv2 features, showing that pre-trained embeddings can eventually surpass learned ones in accuracy and generalization.
(4) We apply trajectory optimization (via the Cross-Entropy Method) in latent space to reach specified goal states, demonstrating zero-shot planning capabilities for novel target shapes.

2. BACKGROUND

We conduct experiments in a differentiable simulator (Plasticine Lab), which approximates the behavior of soft, moldable objects. In our setup, a deformable yellow cube rests at the center, while two cylindrical gripper primitives can be manipulated to deform it. By applying velocity commands that bring the grippers together, we effectively "squish" the object into new shapes. Each collected trajectory consists of a sequence of these grips. Our aim is to learn a latent-space dynamics model that captures the complex, nonlinear evolution of shape under applied actions, enabling downstream planning without relying on explicit particle-based state estimation or large-scale real-world data collection.

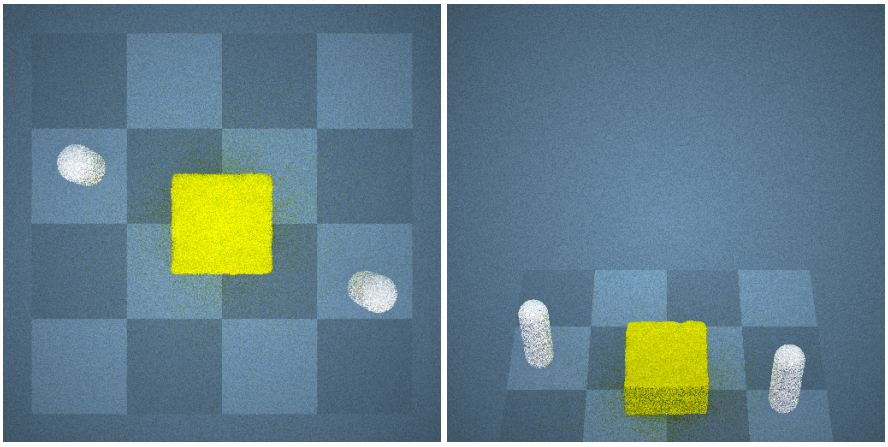

FIGURE 1. Example images of the environment

3. RELATED WORK

Deformable object manipulation is challenging due to complex, high-dimensional state spaces and nonlinear material dynamics. Prior research often leverages simulation methods, such as position-based dynamics or particle-based models [6], for internal representations , and employs data-driven techniques to learn dynamics from raw sensory inputs. While some studies rely heavily on large datasets or expert demonstrations , more recent works emphasize representation learning and latent dynamics modeling [8], aiming for task-agnostic, compact state representations. Prior work has also attempted to build graph-based networks [5, 6], but these often rely on many assumptions regarding object structure.

Meanwhile, pre-trained visual representations have boosted efficiency in numerous robotics and vision tasks. Pre-trained models like R3M [4] for robotics have allowed fast adaptation. Methods such as DINO [7] have demonstrated rich semantic and spatial features that can be extracted without labeled data. Incorporating such embeddings into latent-space world models can reduce the amount of task-specific training required [3], and improve generalization, making them appealing for challenging deformable object tasks where collecting large datasets is difficult.

For our approach, we aim to integrate these insights: we build a vision-based dynamics model that predicts future latent states from limited data and evaluates whether pre-trained embeddings yield better performance than those learned from scratch.

4. APPROACH

4.1. **Observation Model.** We start by encoding high-dimensional RGB images into a low-dimensional latent space. Two variants are considered:

(1) **Learned patch embeddings**: A trainable encoder that learns patch-level features from scratch, tailored to the deformable manipulation domain. This is implemented as a simple linear layer to the same dimensionality as the DINOv2 embedding, with layer normalization.
(2) **Pre-trained DINOv2 embeddings**: Patch embeddings extracted from a DINOv2 model (ViT-S/14) that provides standard, semantically meaningful spatial representations. This DINOv2 version has an embedding dimension of size 384, and would have size 14 x 14 patches.

These embeddings transform each input image into a set of feature patches, $\{z_t^1, z_t^2, \ldots\}$, providing a structured latent state representation.

Before both of these, we would transform the original 512x512 image (from a top-down view) downsampled into a 224x224 image.

4.2. **Dynamics Model.** The dynamics model is a vision transformer (ViT)-based module that predicts the next latent state given the current latent state and the action. Let z_t represent the latent patches at time t, and let $a_t \in \mathbb{R}^4$ be the action consisting of (x, y, z, r), where (x, y, z) is the midpoint position of the grippers and r is a rotation around the vertical axis. The model receives a history of latent states $(z_{t-1}, z_{t-2}, \ldots)$ and corresponding actions $(a_{t-1}, a_{t-2}, \ldots)$, and outputs a prediction $\hat{z}_t$.

We remove global pooling and CLS tokens from the standard ViT architecture to preserve spatial structure. Each patch can attend to every other patch over time, forming a spatiotemporal transformer that captures the evolution of

deformable object surfaces under manipulation. The training loss is a mean squared error (MSE) between predicted latent states and the ground-truth latent states at the next timestep. The action is put through a simple action encoder as well before entering the vision transformer.

4.3. **Planning with the World Model.** Once trained, we use the learned world model for planning. Given a current image and a goal image, we encode both into latent states (z_{start} and z_{goal}). We then apply trajectory optimization to find a sequence of actions that transform the deformable object from the current state to the goal state in latent space.

We use the Cross-Entropy Method (CEM) for optimization:

$$\min_{a_{1:T}} \|f(z_{\text{start}}, a_{1:T}) - z_{\text{goal}}\|^2,$$

where f is the dynamics model rollout. CEM samples candidate trajectories, evaluates their costs, and iteratively refines the action distribution. The final planned actions can be executed to achieve the goal shape.

5. Experimental Results

5.1. **Dataset and Models.** We collected 50 training trajectories (each with 3 grips) and 10 validation trajectories in the Plasticine Lab simulator. Each grip applies a velocity command that deformably reshapes the object. This velocity command is constant, and is executed over 40 timesteps (roughly 3 seconds). We trained two models with identical ViT-based dynamics architectures:

(1) **Learned embeddings (LE)**: Patch embeddings trained from scratch.
(2) **DINO embeddings (DINO)**: Patch embeddings extracted from a pre-trained DINOv2 model.

Each model was trained for 10 epochs with a batch size of 8. We present MSE-based training curves below, noting that LE initially achieves lower error but is eventually outperformed by DINO as training progresses. This suggests that while the pre-trained features may be initially less tailored to the domain, they contain richer spatial semantics that help with long-term accuracy and generalization.

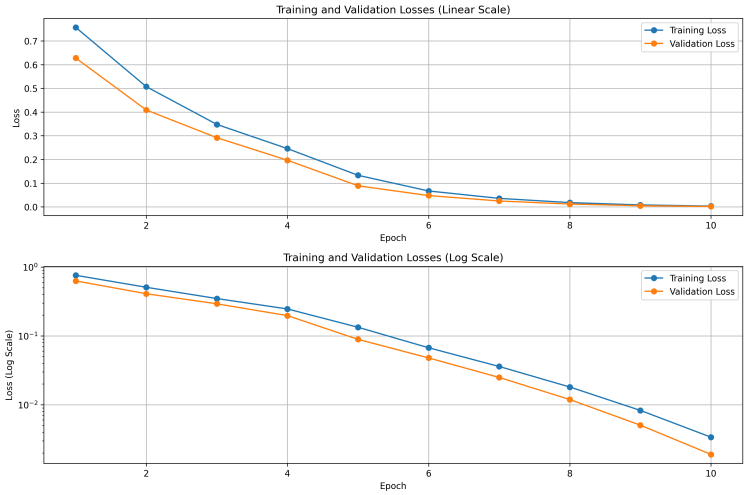

FIGURE 2. MSE Loss Curves for DINOv2 Patch Embeddings.

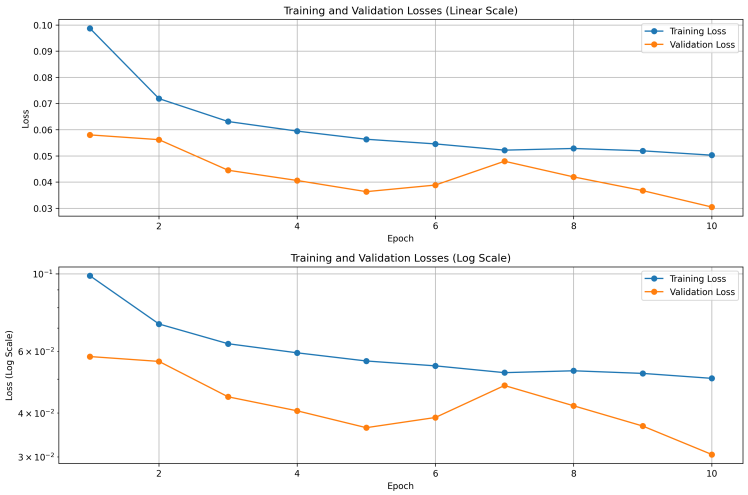

FIGURE 3. MSE Loss Curves for From Scratch Patch Embeddings.

5.2. **Goal Reaching.** To test planning, we specify 5 goal images that are out-of-distribution with respect to the training set. These goals were generated by applying additional grips (beyond the training regime) to create more challenging shapes. We use CEM with 16 samples and 20 iterations to find action trajectories that reshape the deformable object towards these goals.

Since we are dealing with a deformable object, we measure performance using the Chamfer Distance (CD) computed on the ground-truth particle states of the final object:

$$d_{\text{Chamfer}}(X, Y) = \sum_{x \in X} \min_{y \in Y} \|x - y\|_2^2 + \sum_{y \in Y} \min_{x \in X} \|y - x\|_2^2,$$

where X and Y are sets of 3D particle positions for the achieved and goal shapes, respectively. Lower values indicate closer shape match. This is a common loss used in comparing point clouds it is likely more ap for this context. For rigor, we also measured the Mean Square Error loss between the final and goal images. In Table 1 below, we can see the results of the planning testing.

Model	Mean CD	Mean MSE
LE	0.0291	0.209
DINO	0.0345	0.250

TABLE 1. Chamfer Distance and Image Mean Value Loss comparison for goal-reaching tasks

Preliminary results interestingly suggested better results in both Chamfer Distance and Mean Value Loss for the from-scratch embeddings. Given the lower training loss of the DINOv2 architecture, it can likely be inferred that some inherent structures present in a highly deformable object like the training set-up are not encoded within the foundation model. However, it's performance was relatively similar, and it's possible, at a higher scale, that using a pre-trained representation could lower compute time. Overall though, this did not suggest that DINOv2 embeddings were effective in this context.

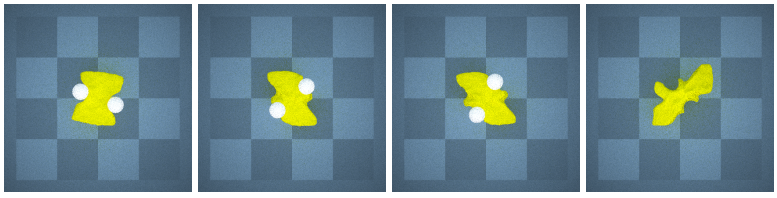

FIGURE 4. Example of a rollout over time, with the Goal State at the end for comparison

6. Discussion

Our results demonstrate that, by leveraging a learned dynamics model and latent representations of visual observations, it is possible to effectively manipulate deformable objects with limited data. Notably, while our comparison between pre-trained DINOv2 embeddings and from-scratch embeddings did not show a clear advantage for the pre-trained features in this particular deformable setting, the general framework remains promising. The discrepancy in performance suggests that deformable manipulation may demand features that capture fine-grained material properties and subtle texture cues, which current large-scale pre-trained representations may not emphasize. As such, future investigation into specialized or fine-tuned pre-trained models could yield improved outcomes.

From an applied perspective, we have shown that zero-shot planning is feasible with a latent world model, allowing the robot to shape deformable objects into novel configurations without additional on-policy training. However, the trajectory optimization procedure (CEM) is computationally expensive. Real-time control may require more sample-efficient optimization strategies or the integration of policy learning, where a policy network could rapidly produce near-optimal actions without iterative sampling.

Our experiments were conducted entirely in a simulation environment. Bridging the gap to real-world scenarios introduces further challenges: sensor noise, lighting variability, and differences in material properties could degrade performance. Moreover, scaling this approach to more complex tasks—such as manipulating multiple deformable objects or achieving intricate target shapes—may require richer representations and more advanced planning algorithms.

In summary, while our initial experiments suggest that general-purpose pre-trained embeddings may not always outperform custom-learned features for deformable object manipulation, the combination of representation learning and model-based control in latent space holds significant potential. Future directions include refining the observation model to better capture deformable dynamics, exploring alternative pre-trained features, integrating reinforcement learning for faster online adaptation, and validating these methods on physical hardware with real deformable materials.

References

[1] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9650-9660).

[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[3] Hafner, D., Pasukonis, J., Ba, J., & Lillicrap, T. (2024). Mastering diverse domains through world models. arXiv preprint arXiv:2301.04104.

[4] Nair, S., Rajeswaran, A., Kumar, V., Finn, C., & Gupta, A. (2022). R3M: A universal visual representation for robot manipulation. arXiv preprint arXiv:2203.12601.

[5] Zhang, K., Li, B., Hauser, K., & Li, Y. (2024). Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation. arXiv preprint arXiv:2407.07889.

[6] Shi, H., Xu, H., Huang, Z., Li, Y., & Wu, J. (2022). RoboCraft: Learning to See, Simulate, and Shape Elasto-Plastic Objects with Graph Networks. *arXiv preprint arXiv:2205.02909*.

[7] Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., & Bojanowski, P. (2024). DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*. ISSN: 2835-8856.

[8] Anonymous. (2024). DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. In *Submitted to The Thirteenth International Conference on Learning Representations*. Under review.

Appendices

Parameter	Value
Batch size	8
Epochs	10
Learning rate	1e-4
Optimizer	AdamW
Context length	3 steps
Action dimension	4 (x,y,z,r_z)

Table 2. ViT HyperParameters.

Hyperparameters.