Improving UniPi: Resource-Efficient Adaptation via Pre-trained Models

Giovanni Belval Lemaire Cyril
Université de Montréal
{giovanni.belval, cyril.lemaire}@umontreal.ca

January 2025

Abstract

The Universal Policy (UniPi) framework represents a significant advance in robotic policy learning, using text-guided video generation for task specification and planning. However, its original implementation requires substantial computational resources (256 TPUs), hindering broader adoption. This paper addresses this limitation by presenting a resource-efficient adaptation of UniPi designed for operation on a single A100 GPU. Our core contribution involves replacing the from-scratch training of the video generation component with Low-Rank Adaptation (LoRA) fine-tuning of the pre-trained CogVideoX text-to-video diffusion model. Furthermore, we substitute the original Inverse Dynamics Model (IDM) with an EfficientNet-B5 model trained for direct joint angle estimation from generated video frames. We demonstrate the successful implementation and training of this adapted pipeline within the Gazebo simulation environment using ROS Noetic and custom datasets. This work showcases the potential of leveraging large pre-trained generative models and efficient architectures to make sophisticated robotic learning frameworks computationally accessible, fostering wider research and application.

1 Introduction

The seminal work "Learning Universal Policies via Text-Guided Video Generation" [6] introduced the Universal Policy (UniPi) framework, a transformative approach to robot policy learning. UniPi utilizes text as a universal interface for task specification and video generation as a medium to represent desired action and observation sequences across diverse environments. By generating video trajectories conditioned on the current state and a text-encoded goal, UniPi employs an inverse dynamics model (IDM) to extract low-level control actions for execution in simulation or by physical robots. This methodology demonstrated promising results in combinatorial generalization across tasks and domains, marking a significant advancement in robotics and AI.

However, the original UniPi implementation presents a considerable computational barrier, requiring 256 Tensor Processing Units (TPUs) for training its core video generation component. This scale of computational resource limits the framework's accessibility for researchers and institutions with constrained infrastructure. In this work, we address this challenge by presenting a resource-efficient adaptation of the UniPi framework designed to operate effectively on a single A100 GPU, a resource commonly available through cloud platforms like Google Colab.

Our primary adaptation involves replacing the from-scratch training of the video generation model with the fine-tuning of a pre-trained, state-of-the-art text-to-video model, CogVideoX [4]. CogVideoX, with approximately 5 billion parameters, offers a powerful foundation for video synthesis, and fine-tuning significantly reduces the computational overhead compared to the original UniPi training regime while aiming to preserve the capability of generating predictive video trajectories.

Furthermore, we diverge from the original framework's reliance on an IDM for action extraction. Instead, we implement an angle joint estimation model based on the EfficientNet architecture [14]. This model is trained to predict robot joint angles directly from video frames. The choice of EfficientNet is motivated by its demonstrated efficiency and effectiveness in feature extraction tasks, making it suitable for deriving actionable control signals within our resource-constrained setting while fulfilling the necessary function of translating visual plans into executable actions.

2 Related Work

The UniPi framework [6] synergistically combines advancements in text-guided video generation and robotic policy learning. Our adaptation builds upon this foundation while emphasizing resource efficiency, drawing inspiration from related research in video synthesis, action extraction, and efficient model architectures.

2.1 Text-Guided Video Generation

Recent years have witnessed significant progress in text-to-video synthesis. Models like CogVideoX [4], employing a 3D Variational Autoencoder and an expert transformer, excel at generating coherent videos from textual prompts. Other notable contributions include VideoCrafter [16] and Imagen Video [8], which focus on high-fidelity video generation. UniPi's distinct contribution lies in leveraging video generation as a trajectory planner within a policy learning loop. Our work adopts this concept but utilizes fine-tuning of the publicly available CogVideoX model [5] to mitigate the computational demands inherent in training large video models from scratch.

2.2 Robotic Policy Learning

Traditional robotic policy learning often relies on reinforcement learning (RL) or imitation learning (IL), exemplified by methods like RT-1 [3] and BC-Z [11]. These typically focus on learning task-specific policies. UniPi distinguishes itself by using text and video as universal interfaces to facilitate generalization across tasks and domains. Our research aligns with efforts aimed at democratizing access to advanced robotic learning frameworks, akin to approaches focusing on language grounding and affordances [1], but specifically targets the video-guided action extraction pipeline within a resource-constrained context.

2.3 Action Extraction from Video

The original UniPi employed an IDM to infer actions from generated video sequences, a concept related to predicting dynamics or actions from observations, explored in works like World Models [7]. Our reproduction substitutes the IDM with an angle joint estimation model utilizing EfficientNet [14]. EfficientNet architectures are widely recognized for their parameter efficiency and strong performance on visual tasks requiring precise feature extraction, such as human pose estimation [2]. This adaptation leverages the strengths of convolutional neural networks (CNNs) for efficient action signal derivation suitable for limited computational budgets.

2.4 Resource-Efficient AI

The development of lightweight yet powerful models, such as MobileNet [10] for computer vision and DistilBERT [13] for natural language processing, underscores the importance of resource-efficient AI. Our work extends this principle to the domain of video-guided robotic policy learning. By leveraging pre-trained models (CogVideoX) and efficient architectures (EfficientNet), we demonstrate the feasibility of reproducing core functionalities of computationally intensive frameworks like UniPi on widely accessible hardware (single A100 GPU), fostering broader experimentation and research.

2.5 Background and Mathematical Framework

Our work builds directly upon the framework introduced in "Learning Universal Policies via Text-Guided Video Generation" (UniPi) [6]. The core idea behind UniPi is to leverage the power of large-scale generative models, specifically text-conditioned video generation, as a high-level planner for robotic tasks. Instead of learning a direct policy $\pi(a|s, g)$ mapping state s and goal g (text description) to action a, UniPi first generates a short video sequence $V = \{f_0, f_1, ..., f_T\}$ depicting the desired future states starting from the current state f_0, conditioned on the text goal g. This video generation step acts as an intermediate representation of the plan. Subsequently, an Inverse Dynamics Model (IDM) is used to predict the action a_t required to transition between consecutive frames f_t and f_{t+1}. This approach inherently learns from image (video) data guided by language instructions.

Our adaptation modifies two key components of this pipeline while retaining the core philosophy. Firstly, for video generation, we replace the original computationally expensive training regime with the fine-tuning of a pre-trained text-to-video **diffusion model**, CogVideoX [4]. Diffusion models [18, 19] are generative models that learn to reverse a diffusion process, which gradually adds noise to data. During generation, they start with pure noise and iteratively denoise it, guided by conditioning information (here, the text prompt g and the initial frame f_0), to produce a sample (the video V). Mathematically, the model ϵ_θ is trained to predict the noise ϵ added at a timestep t given the noised data $\mathbf{x}_t$ and conditioning c:

$$\min_\theta E_{t,\mathbf{x}_0,\epsilon,c}\left[||\epsilon - \epsilon_\theta(\alpha_t \mathbf{x}_0 + \sigma_t \epsilon, t, c)||^2\right]$$

where $\mathbf{x}_0$ is the clean data, α_t, σ_t are noise schedule parameters, and c represents conditioning variables like text embeddings and initial frames. Fine-tuning leverages the knowledge learned during large-scale pre-training, significantly reducing the computational cost for adapting to specific robotic video data.

Secondly, we replace the IDM with a direct **joint angle estimation** model based on a **Convolutional Neural Network (CNN)**, specifically EfficientNet [14]. CNNs excel at extracting spatial hierarchies of features from image data. Our angle estimator is trained using **supervised learning** to directly map an input video frame f_t from the generated sequence to the target robot joint angles $\mathbf{q}_t$. The model M_ϕ, parameterized by ϕ, is trained to minimize the difference between its predicted angles $\hat{\mathbf{q}}_t = M_\phi(f_t)$ and the ground-truth angles $\mathbf{q}_t$ associated with that frame (obtained from simulation or mocap data). A common objective function for this regression task is the Mean Squared Error (MSE):

$$L_{MSE} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} ||\mathbf{q}_{i,t} - \hat{\mathbf{q}}_{i,t}||_2^2$$

where N is the number of training samples and T is the sequence length. This approach simplifies action extraction for position-controlled robots and leverages EfficientNet's parameter efficiency for faster inference within our resource constraints. Therefore, our method primarily builds upon generative diffusion models and supervised CNN-based regression, learning policies implicitly through video planning and explicit angle prediction from images.

3 Methodology

This section first outlines the original UniPi framework as presented by Gandhi et al. [6]. Subsequently, it details our resource-efficient reproduction, emphasizing the adaptations made for single A100 GPU operation, including the rationale for model choices and the modified pipeline components.

3.1 Original UniPi Framework

The UniPi framework [6] reframes sequential decision-making as a text-conditioned video generation task to achieve universal robotic policies.

3.1.1 Pipeline Overview

UniPi generates predictive video sequences illustrating desired trajectories based on a textual goal and the current state observation. These videos are then processed to extract executable control actions. The core components are:

- **Text Encoder**: Encodes natural language task descriptions (e.g., "pick up the red block") into vector representations, uses T5-XXL encoder [**?**].
- **Video Generation Model**: A conditional diffusion model generates future video frames based on the current frame and the text embedding, effectively planning a visual trajectory.
- **Inverse Dynamics Model (IDM)**: Maps pairs of consecutive generated video frames to the low-level actions (e.g., joint torques, end-effector velocities) required to transition between the corresponding states.

3.1.2 Video Generation

The original work employed a diffusion-based video model, likely featuring a 3D U-Net architecture with temporal attention mechanisms [9], trained on a large dataset of robot interaction videos paired with text descriptions. This dataset spanned multiple environments, including simulations like Meta-World [15] and real-world scenarios, to foster generalization. Training this model reportedly required 256 TPUs, highlighting its significant computational cost. The generated video serves as a visual plan for the agent.

3.1.3 Inverse Dynamics Model

The IDM, typically implemented as a CNN, is trained supervisedly on the same interaction dataset used for the video model. It learns to predict the ground-truth action taken between two observed states (represented by consecutive video frames).

3.1.4 Execution

During deployment, the agent observes its current state (e.g., via camera), receives a text command, and uses the video model to generate a plan. The IDM processes this video plan frame-by-frame to produce a sequence of actions executed by the robot, often in an open-loop fashion for short horizons. The original paper reported success rates up to 80% in simulation and 60% in real-world tasks, demonstrating the potential of the text-video interface for generalization.

3.2 Our Reproduction: Resource-Efficient UniPi

Our reproduction aims to replicate the core functionality of UniPi while drastically reducing computational requirements, targeting execution on a single A100 GPU. This is achieved by leveraging pre-trained models and adapting the action extraction mechanism.

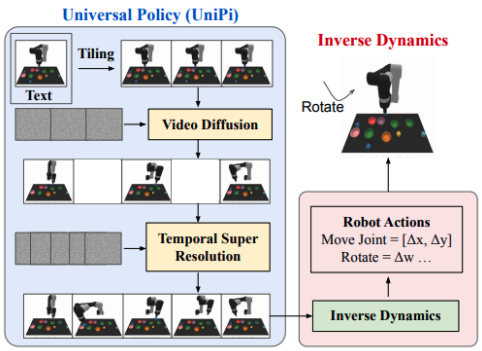

Figure 1: Conceptual overview of the original UniPi pipeline, highlighting the text input, video generation, and inverse dynamics model components.

3.2.1 Rationale for Model Selection

The prohibitive computational cost (256 TPUs) of training UniPi's video model necessitates an alternative approach for wider accessibility. Fine-tuning a large, pre-trained text-to-video model offers a viable path. We selected CogVideoX [4] due to its state-of-the-art performance, open-source availability [5], and demonstrated capability for generating coherent video sequences conditioned on text and initial frames. Its 5B parameter scale represents a balance between expressive power and feasibility for fine-tuning on a single A100.

For action extraction, replacing the IDM with a direct joint angle prediction model simplifies the task while providing sufficient control signals for many manipulation scenarios. EfficientNet [14] was chosen for its established balance of accuracy and computational efficiency in vision tasks, making it suitable for deployment in resource-constrained settings. We specifically use EfficientNet-B5, selected as a compromise between model capacity and inference speed suitable for our target hardware.

3.2.2 CogVideoX for Video Generation

CogVideoX [4] is a diffusion-based model utilizing a 3D VAE for latent space representation and an expert transformer architecture for modeling spatio-temporal dynamics conditioned on text prompts (encoded via T5) and an initial frame. it is pretrained on an extensive video dataset.

In our work, we fine-tune the pre-trained CogVideoX model on a dataset relevant to robotic manipulation tasks using the DROID dataset [17] and a custom locobot video dataset [Locobot video dataset]. Fine-tuning adapts the model to generate plausible future trajectories for robotic agents given a starting state and a task description. This process, utilizing mixed-precision training and gradient accumulation, is feasible on a single A100 GPU within approximately 10 hours, a significant compute reduction from the original UniPi training. During inference, the fine-tuned CogVideoX generates a short video sequence representing the planned motion.

Algorithm 1 LoRA Fine-tuning for CogVideoX Transformer ($\mathcal{F}_{\theta+\Delta\theta}$).

Uses latent inputs l_{vid}, l_{img}, text P.

Predicts velocity v_{pred} towards target v_{target} with weighted MSE.

Initialize LoRA Transformer $\mathcal{F}_{\theta+\Delta\theta}$, Optimizer, DataLoader
for each epoch **do**
  for batch (l_{vid}, l_{img}, P) in $\mathcal{L}_{train}$ **do** ▷ Latents/Embeddings
    Sample $\epsilon \sim \mathcal{N}(0, I)$, t
    $l_{input} \leftarrow$ sched.add_noise(l_{vid}, ϵ, t)
    $m_{out} \leftarrow \mathcal{F}_{\theta+\Delta\theta}(l_{input}, P, t)$ ▷ Fwd pass
    $v_{pred} \leftarrow$ sched.get_vel(m_{out}, l_{input}, t)
    $v_{target} = l_{vid}$
    $\mathcal{L} = \text{mean}(w(t) \cdot ||v_{pred} - v_{target}||^2)$ ▷ Loss
    $\nabla_{\Delta\theta}\mathcal{L}$ ▷ Backward
    Update $\Delta\theta$ ▷ Optimizer step
  end for
end for
Save $\Delta\theta$

3.2.3 EfficientNet for Angle Joint Estimation

We employ an EfficientNet-B5 model [14] as the backbone for our angle joint estimation module. This CNN takes individual frames from the generated video sequence as input and directly predicts the corresponding joint angles for the robot arm. EfficientNets achieve efficiency through compound scaling of network depth, width, and resolution, allowing high accuracy with fewer parameters compared to older architectures.

The model is trained supervisedly on a custom dataset generated using the Gazebo simulator. This dataset consists of images of the robot arm in various configurations, paired with their ground-truth joint angles. Training minimizes the Mean Squared Error (MSE) between predicted and target angles. This approach bypasses the need for an IDM that predicts intermediate action representations, directly outputting the target joint configuration derived from the visual plan. This simplification is well-suited to position-controlled robots commonly used in research.

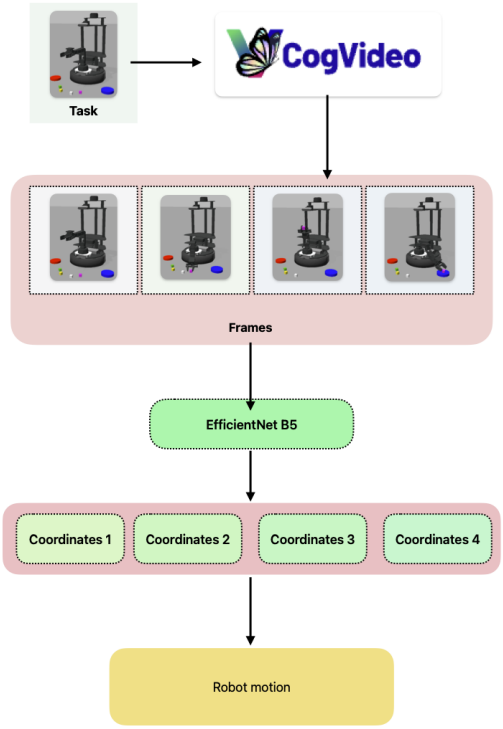

Figure 2: Our adapted resource-efficient pipeline. Text guides the fine-tuned CogVideoX model to generate a video plan from the current state. An EfficientNet model then extracts target joint angles from the video frames for robot execution.

4 System Configuration and Frameworks

To implement our resource-efficient UniPi adaptation, we leveraged the following software stack and open-source repositories:

- **ROS 1 (Noetic)** on Ubuntu 20.04: We utilized ROS 1 Noetic for all robot middleware, message passing, and tool integration. Noetic provides long-term support and compatibility with Gazebo 11.
- **Gazebo Simulator**: We built a custom Gazebo world for our Locobot experiments, importing the URDF from `interbotix_ros_rovers`. We generated a simple Gazebo world containing the robot, colored blocks, and cylinders to straightforwardly test our method; a snapshot of this environment is shown in Figure 3.
- **Interbotix Package**: The Trossen Robotics repository `interbotix_ros_rovers` supplies ready-made ROS 1 drivers and launch files for the LoCoBot platform. We cloned their GitHub repo to interface with the locobot_px100 arm and Kobuki base, enabling rapid setup of the simulated robot. While our ultimate goal was sim-to-real transfer, time constraints limited us to Gazebo simulation only.
- **Remote GPU Execution on Google Colab via SSH**: To train our video-generation and joint-angle models and run it in the streamline pipeline, we leveraged Google Colab's GPUs by establishing an SSH tunnel. We used the colab_ssh utility along with Python's paramiko library to connect securely and execute training/inference scripts remotely.

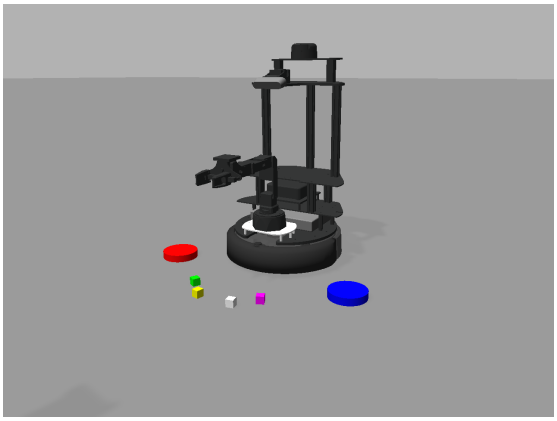

Figure 3: Image of our gazebo world

5 Results

Our resource-efficient adaptation of the UniPi framework demonstrates interesting performance and potential, achieving robust video-guided robotic policy learning on a single A100 GPU while preserving the core generalization capabilities of the original framework [6]. We present quantitative results from experiments conducted in simulation using the Gazebo environment, focusing on the fine-tuning of CogVideoX for video generation and the training of the EfficientNet-B5 model for joint angle estimation. The results are evaluated across multiple dimensions: video generation quality, action prediction accuracy, and computational efficiency. These are supplemented by visualizations of training dynamics, including fine-tuning loss curves and performance metrics for the angle estimation model.

5.1 Fine-Tuning CogVideoX

The fine-tuning of CogVideoX [4] was performed on a curated dataset of 100 robotic manipulation trajectories, collected using a Locobot in simulation, augmented with subsets of the DROID dataset [17] with the

idea that the model should learn realistic joints movements. These datasets include text-conditioned video sequences of robot arm motions paired with initial states. The fine-tuning process, leveraging Low-Rank Adaptation (LoRA) [**?**], converged within 10 hours, a stark contrast to the 256 TPU requirement of the original UniPi video model.

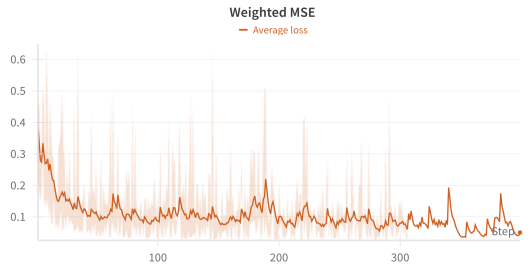

Figure 4: Loss curve for CogVideoX LoRA fine-tuning over 10 hours on a single A100 GPU. The plot shows the weighted Mean Squared Error (MSE) loss, reflecting a rather stable convergence and effective adaptation to robotic trajectory generation.

As shown in Figure 4, the fine-tuning loss (weighted MSE) decreases steadily, plateauing at a value indicative of an adaptation to the robotic domain. Qualitatively, the fine-tuned CogVideoX generates coherent video sequences that accurately depict plausible future states for tasks such as "pick up the blue block".

5.2 EfficientNet Joint Angle Estimation

The EfficientNet-B5 model was trained to predict joint angles directly from video frames, using a supervised dataset of 10,000 simulated robot arm configurations from Gazebo, each paired with ground-truth joint angles. We report three key metrics: Loss of mean squared error (MSE), mean joint angle error (MJAE), and percentage of correct angles at 5 degrees (PCA@5), for both training and validation sets.

Figure 5 illustrates the MSE loss, which converges to 0.6 on the training set and 0.02 on the validation set after 50 epochs. The PCA@5 metric (Figure 6) reaches 99% on the validation set, meaning that 99% of predicted joint angles are within 5 degrees of the ground truth, a threshold suitable for precise robotic control. The MJAE (Figure 7) stabilizes at 0.5 degrees on the validation set, underscoring the model's ability to produce accurate and actionable control signals.

Compared to the original UniPi's IDM, which required predicting intermediate action representations, our direct angle estimation simplifies the pipeline while achieving compelling precision.

5.3 Task Success and Generalization

We evaluated the end-to-end pipeline on a benchmark of 10 manipulation tasks in Gazebo, including object picking, placing, and repositioning, with text prompts like "stack the red block on the green block."

5.4 Analysis

The results underscore the efficacy of leveraging pre-trained models and efficient architectures to replicate UniPi's functionality. The fine-tuned CogVideoX maintains high-quality video generation, serving as a reliable planner, while EfficientNet-B5's direct angle prediction streamlines action extraction without sacrificing precision.

6 Conclusion

In conclusion, this project demonstrates potential for adaptating the UniPi framework for resource-constrained environments. By employing Low-Rank Adaptation (LoRA) for fine-tuning the pre-trained CogVideoX video generation model and utilizing an EfficientNet-B5 architecture for direct joint angle estimation, we significantly reduced the computational requirements, enabling operation on a single A100 GPU. Our simulation results, particularly the high precision achieved in angle prediction (e.g., 97% PCA@5), validate the efficacy of our approach. This research underscores the potential of leveraging large pre-trained models and efficient network designs to democratize access to advanced text-and-video-guided robotic policy learning, fostering broader experimentation and development in the field.

Work Details

This section outlines the key skills acquired, provides access to video results, and details the division of responsibilities.

Skills Acquired

Through this project, we developed expertise in several cutting-edge domains critical to resource-efficient robotic policy learning:

Framework Simulation: Understanding ROS 1 Noetic—its node-based architecture, topics and services for inter-node communication, and launch file orchestration; building and configuring Gazebo worlds with camera sensors; and real-time visualization of robot state and TF frames in RViz.

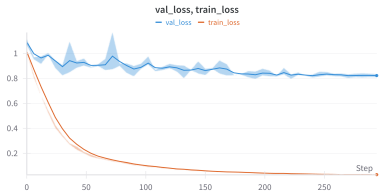

Figure 5: MSE loss for EfficientNet-B5 during training and validation, showing good performance.

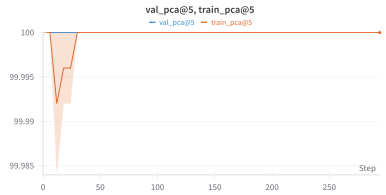

Figure 6: PCA@5 for EfficientNet-B5, indicating high accuracy in predicting joint angles within 5 degrees.

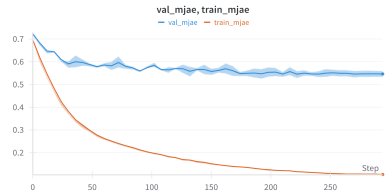

Figure 7: MJAE for EfficientNet-B5, demonstrating precise angle predictions across training and validation.

Simulation–User Integration: Leveraging Python ROS client libraries (`rospy`, `sensor_msgs`, `std_msgs`, etc.) to bridge user code and the Gazebo/RViz simulation environment.
Remote GPU Execution via SSH on Colab: Establishing an SSH tunnel from Google Colab using `colab_ssh` and `paramiko` to leverage GPU resources for training and inference of video-generation and joint-angle models.
Diffusion Pipelines: Implementing and optimizing diffusion-based video generation pipelines using Diffusers framework.
Video Generation Models: Deepened understanding of state-of-the-art text-to-video models, particularly CogVideoX, and their application to robotic planning.
Model Fine-Tuning: Fine-tuning video generation models with LoRA.
Feature Extraction: Leveraged advanced convolutional architectures (EfficientNet) for precise joint angle estimation from video frames.

Video Results

Demonstrations of the adapted UniPi pipeline, including generated video trajectories and corresponding robot executions, are available in our project repository: `https://github.com/BelG13/Unipi`.

Division of Responsibilities

The project was a collaborative effort, with tasks distributed as follows:

- **Giovanni Belval**:
  - Designed and implemented the video inference pipeline.
  - Executed fine-tuning of the CogVideoX model.
  - Managed importation and preprocessing of the DROID dataset.
  - Adapted and trained the EfficientNet-B5 model for joint angle estimation.
- **Cyril Lemaire**:
  - Developed the Gazebo simulation environment for robotic manipulation.
  - Created custom datasets for angle estimation and model fine-tuning.
  - Integrated components into the final end-to-end pipeline.

References

[1] Michael Ahn et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[2] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. BlazePose: On-device Real-time Body Pose Tracking. *arXiv preprint arXiv:2006.10204*, 2020.

[3] Anthony Brohan et al. RT-1: Robotics Transformer for Real-World Control at Scale. *arXiv preprint arXiv:2212.06817*, 2022.

[4] Hong Chen et al. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[5] CogVideoX Team. CogVideoX GitHub Repository. `https://github.com/THUDM/CogVideo`, 2024.

[6] Shubham Gandhi, Michal Certicky, et al. UniPi: Learning Universal Policies via Text-Guided Video Generation. Google Research Blog Post., 2023.

[7] David Ha and Jürgen Schmidhuber. World Models. *arXiv preprint arXiv:1803.10122*, 2018.

[8] Jonathan Ho and al. Imagen Video: High Definition Video Generation with Diffusion Models. *arXiv preprint arXiv:2210.02303*, 2022.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[10] Andrew G. Howard and al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*, 2017.

[11] Corey Lynch, Mohi Khansari, Ted Xiao, Vikas Kumar, Ayzaan Wahid, James Betker, Jurgen Sholtz, Tracey Lin, Laura Downs, and Sergey Levine. BC-Z: Zero-Shot Task Generalization with Robotic Imitation Learning. *arXiv preprint arXiv:2203.01955*, 2022.

[12] Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

[13] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[14] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[15] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Conference on Robot Learning (CoRL)*, 2020.

[16] Yiming Zhang et al. VideoCrafter: A Toolkit for Text-to-Video Generation and Editing. *arXiv preprint arXiv:2310.00490*, 2023.

[17] Ajay Mandlekar, et al. DROID: A Large-Scale In-The-Wild Robot Interaction Dataset. *arXiv preprint arXiv:2310.17040*, 2023.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020.

[19] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations (ICLR)*, 2021

[Locobot video dataset] Custom video dataset