AI Model for Robotic Action Frame Prediction

Fang Hanbo	Fang Zicheng	Shi Jiayi	Xu Bowen	Zhang Tong
121090119	121090122	121090478	121090661	121090788

Abstract

Predicting future visual observations of a robotic agent is crucial for safe and effective task execution. We fine-tune a pretrained InstructPix2Pix model on a task-specific dataset generated by RoboTwin to predict robot camera frames 50 steps ahead, conditioned on the current frame and a textual action instruction. Our dataset comprises 300 observations across three manipulation tasks: beat the block with the hammer, handover the blocks, and stack blocks. We evaluate the model using SSIM and PSNR, demonstrating its capability to generate high-fidelity predictions and discussing limitations and future directions.

1. Introduction

Robotic systems operating in dynamic environments benefit significantly from the ability to predict their future observations. Such predictive models enable anticipatory planning, collision avoidance, and more reliable interaction with surroundings. In this work, we focus on learning a visual forecasting model that, given an image of the robot's current view and a natural language instruction (*e.g.*, "beat the block with the hammer"), generates the robot's view 50 frames later in a virtual simulation environment. We leverage a pretrained InstructPix2Pix as the base model and fine-tune it on a small task-specific dataset generated using RoboTwin. Our contributions include: (1) a methodology for adapting InstructPix2Pix to future frame prediction; (2) a dataset generation pipeline with 100 examples per task; and (3) quantitative evaluation using SSIM and PSNR metrics.

2. Related Work

We review five relevant areas:

- **Future Frame Prediction:** Finn *et al.* proposed action-conditioned video prediction via convolutional LSTMs [3].
- **Text-to-Image Generation:** Rombach *et al.* introduced Stable Diffusion for high-fidelity image synthesis from text [6].
- **Instruction-Guided Editing:** Brody *et al.* developed InstructPix2Pix for controllable image editing [1].
- **Evaluation Metrics:** Wang *et al.* formalized SSIM for image similarity evaluation [7].

3. Method

3.1. Base Model: InstructPix2Pix

We adopt the InstructPix2Pix architecture [1], a conditional diffusion framework that edits an input image I_t based on a textual instruction L. The model comprises three main components:

- **Latent Autoencoder:** An encoder $E(\cdot)$ projects I_t into a latent $z_t = E(I_t)$, and a decoder $D(\cdot)$ reconstructs images from latents.
- **Diffusion U-Net:** A U-Net backbone $\epsilon_\theta(z_t^{(k)}, \tau, k)$ predicts noise residuals for latent $z_t^{(k)}$ at diffusion step k, conditioned on the text embedding $\tau = T(L)$ (e.g., a CLIP encoder).
- **Cross-Attention Layers:** Within each U-Net block, cross-attention modules fuse visual features with language embeddings, enabling precise instruction following.

The denoising update at time step k follows:

$$z_t^{(k-1)} = \frac{1}{\sqrt{\alpha_k}}\Big(z_t^{(k)} - \frac{1-\alpha_k}{\sqrt{1-\bar{\alpha}_k}}\,\epsilon_\theta(z_t^{(k)}, \tau, k)\Big),$$

where $z_t^{(k)} = \sqrt{\bar{\alpha}_k}\, z_{t+50} + \sqrt{1-\bar{\alpha}_k}\,\epsilon$ and $\epsilon \sim \mathcal{N}(0, I)$.

3.2. Fine-Tuning for Frame Prediction

To repurpose InstructPix2Pix for robotic action frame prediction, we make the following adaptations:

- **Input Construction:** We encode the current frame I_t to latent z_t, compute instruction embedding $\tau = T(L)$, and append a learned positional embedding p_{50} to signify the 50-frame horizon.
- **Training Loss:** Given the ground-truth future latent z_{t+50}, we minimize the MSE denoising objective:

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{k, z_{t+50}, \tau, \epsilon}\big[\|\epsilon - \epsilon_\theta(z_t^{(k)}, \tau, k)\|^2\big],$$

where $z_t^{(k)} = \sqrt{\bar{\alpha}_k}\, z_{t+50} + \sqrt{1-\bar{\alpha}_k}\, \epsilon$.

- **Optimization:** We fine-tune the U-Net parameters θ with AdamW (learning rate 1×10^{-4}) for 30 epochs, batch size 8, on our 300-sample dataset.

4. Experiments

4.1. Dataset Generation

To train our frame prediction model, we utilized the RoboTwin simulation platform [5], which offers a suite of dual-arm robotic manipulation tasks. Specifically, we focused on three tasks: *block_hammer_beat*, *block_handover*, and *blocks_stack_easy*. Each task was paired with a corresponding textual instruction: "beat the block with the hammer", "handover the blocks", and "stack blocks", respectively.

For each task, we conducted 100 simulation episodes. During each episode, the simulation environment was configured to record RGB-D images at a resolution of 640×480 pixels from multiple camera viewpoints, including wrist-mounted and overhead cameras. The simulation ran at 30 frames per second, and each episode lasted approximately 7 seconds, resulting in over 200 frames per episode.

To prepare the dataset for training, we extracted frame pairs (I_t, I_{t+50}) from each episode, where I_t represents the current frame at time t, and I_{t+50} is the frame 50 steps ahead, corresponding to approximately 1.67 seconds into the future. Each pair was associated with the task-specific textual instruction L. This process yielded 100 frame pairs per episode, resulting in a total of 10,000 frame pairs per task. Across all three tasks, we obtained 30,000 frame pairs.

We divided the dataset into training and testing sets using an 80/20 split. The training set comprised 24,000 frame pairs, while the testing set contained 6,000 frame pairs

4.2. Training Details

We fine-tuned the InstructPix2Pix model using the Hugging Face `diffusers` library, leveraging the `accelerate` framework for efficient training with mixed precision. The training was conducted on virtual GPU cloud provider AutoDL with 32GB of VRAM.

Preprocessing. All images were resized to 256×256 resolution. Random horizontal flipping was applied as a data augmentation strategy to enhance model generalization.

Optimization. We employed the AdamW optimizer with a learning rate of 5×10^{-5}. Gradient accumulation over 4 steps was used to achieve an effective batch size of 8. Gradient checkpointing was enabled to reduce memory consumption during training. No learning rate warm-up was applied.

Model Architecture. The UNet backbone was modified to accommodate 8 input channels, accounting for the concatenation of the original image and the conditioning image. The first convolutional layer was adjusted accordingly, with the additional channels initialized to zero, following the implementation details provided in the `diffusers` library [2].

Checkpointing. Model checkpoints were saved every 5000 steps, with a maximum of one checkpoint retained to conserve storage space.

Reproducibility. A fixed random seed of 42 was set to ensure reproducibility of the training process.

4.3. Evaluation Metrics

We evaluate the performance of our robotic action frame prediction model using two widely adopted image quality metrics: Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR). These metrics compare the predicted future frame against the ground-truth frame and reflect how visually and structurally similar the generated image is to the actual outcome.

Structural Similarity Index (SSIM). SSIM measures the perceptual similarity between two images by comparing their luminance, contrast, and structural information. Unlike pixel-wise metrics such as MSE, SSIM aligns better with human visual perception. It is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where μ_x and μ_y denote the mean of images x and y, σ_x^2 and σ_y^2 represent their variances, and σ_{xy} is the covariance between them. Constants C_1 and C_2 are used to stabilize the division, and are typically set based on the dynamic range of the pixel values.

In our task, SSIM evaluates how well the fine-tuned model preserves structural details (e.g., object boundaries, spatial relationships) in the predicted frame after performing an action such as "stack blocks" or "hit the block with the hammer."

Peak Signal-to-Noise Ratio (PSNR). PSNR measures the ratio between the maximum possible pixel value and the mean squared error (MSE) between the predicted and ground-truth images. It is defined as:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{\text{MSE}} \right)$$

where I and K denote the ground-truth and predicted images, respectively, and MAX_I is the maximum possible pixel value (255 for 8-bit images). PSNR provides a measure of reconstruction quality: the higher the PSNR, the more similar the predicted image is to the ground truth.

In our evaluation, PSNR complements SSIM by quantifying the overall fidelity of the prediction. A higher PSNR indicates less distortion in pixel values, which is critical in ensuring accurate visual feedback for robotic control.

Evaluation Metrics To assess the quality of predicted future frames, we employ two widely used image similarity metrics: Structural Similarity Index Measure (SSIM) and Peak Signal-to-Noise Ratio (PSNR). SSIM evaluates the perceived quality of images by considering luminance, contrast, and structural information, providing a value between 0 and 1, where 1 indicates perfect similarity. PSNR measures the ratio between the maximum possible power of a signal and the power of corrupting noise, expressed in decibels (dB); higher values denote better quality.

We compute SSIM and PSNR between each predicted frame and its corresponding ground-truth frame in the test set across the three defined tasks: *block_hammer_beat*, *block_handover*, and *blocks_stack_easy*. The average results are summarized in Table **??**.

Table 1. Average SSIM for each task (Finetuned vs. Not Finetuned).

Task	Finetuned	Not Finetuned
block_hammer_beat	0.7890	0.6176
block_handover	0.7935	0.6578
blocks_stack_easy	0.7826	0.6195

Table 2. Average PSNR (dB) for each task (Finetuned vs. Not Finetuned).

Task	Finetuned	Not Finetuned
block_hammer_beat	15.76	9.93
block_handover	17.86	11.65
blocks_stack_easy	14.55	10.18

The results demonstrate that fine-tuning the model significantly improves both SSIM and PSNR across all tasks, indicating enhanced structural and pixel-level accuracy in the predicted frames.

5. Conclusion and Future Work

In this work, we demonstrate that fine-tuning the InstructPix2Pix model enables plausible future frame predictions in robotic manipulation tasks. Our approach, conditioned on current observations and textual instructions, shows significant improvements in SSIM and PSNR metrics across various tasks, indicating enhanced structural and pixel-level accuracy.

However, the current methodology has limitations. The dataset size is relatively small, and the model lacks robustness in multimodal fusion, particularly in integrating temporal dynamics and task progression. These constraints hinder the model's ability to generalize to more complex or longer-horizon tasks.

To address these challenges, future work will explore the integration of the GR-MG (Generative Robot Policy with Multi-modal Goals) framework [4]. GR-MG introduces a progress-guided goal image generation model that incorporates task progress information into the generation process. By conditioning on both textual instructions and goal images, GR-MG leverages partially annotated data, such as videos without action labels or robot trajectories without text annotations, enhancing the generalization capabilities of robots.

Implementing GR-MG necessitates augmenting our dataset to include detailed annotations of task progress for each frame. This enhancement will facilitate the generation of more accurate goal images and improve the policy's ability to predict actions based on both visual and textual inputs. Additionally, we plan to expand our dataset to encompass a broader range of tasks and longer sequences, enabling the model to learn more complex behaviors and improve its planning capabilities.

In summary, integrating GR-MG into our framework represents a promising direction for advancing robotic action prediction. By leveraging partially annotated data and incorporating task progress information, we aim to develop a more robust and generalizable model capable of handling a wider array of manipulation tasks.

6. Distribution of the Workload

- Data generation and simulation: Zhang Tong
- Model adaptation and fine-tuning: Zhang Tong and Xu Bowen
- Evaluation and metrics analysis: Fang Hanbo
- Report writing and LaTeX formatting: Fang Zicheng and Shi Jiayi

References

[1] Shahar Brody, Omer Bar-Tal, Yuval Alaluf, Roy Shalev, Yotam Nitzan, Amit H Bermano, and Daniel Cohen-Or. In-

structpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2301.12950*, 2023. 1

[2] Hugging Face. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2023. 2

[3] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29, 2016. 1

[4] Peiyan Li, Hongtao Wu, Yan Huang, Chilam Cheang, Liang Wang, and Tao Kong. Gr-mg: Leveraging partially annotated data via multi-modal goal-conditioned policy. *arXiv preprint arXiv:2408.14368*, 2024. 3

[5] Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). *arXiv preprint arXiv:2409.02920*, 2024. 2

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1

[7] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 1