Bootstrapping Action-Grounded Visual Dynamics in Unified Vision–Language Models

Anonymous ACL submission

Abstract

Can unified vision–language models (VLMs) perform *forward dynamics prediction* (FDP), i.e., predicting the future state (in image form) given the previous observation and an action (in language form)? We find that VLMs struggle to generate physically plausible transitions between frames from instructions. Nevertheless, we identify a crucial asymmetry in multimodal grounding: fine-tuning a VLM to learn *inverse dynamics prediction* (IDP)—effectively captioning the action between frames—is significantly easier than learning FDP. In turn, IDP can be used to bootstrap FDP through two main strategies: 1) weakly supervised learning from synthetic data and 2) inference time verification. Firstly, IDP can annotate actions for unlabelled pairs of video frame observations to expand the training data scale for FDP. Secondly, IDP can assign rewards to multiple samples of FDP to score them, effectively guiding search at inference time. We evaluate the FDP resulting from both strategies through the task of *action-centric image editing* on AURORA-BENCH with two families of VLMs. Despite remaining general-purpose, our best model achieves a performance competitive with state-of-the-art image editing models, improving on them by a margin between 7% and 13% according to GPT4o-as-judge, and achieving the best average human evaluation across all subsets of AURORA-BENCH.[1].

1 Introduction

World models are instrumental in training embodied agents to endow them with specific abilities, such as planning and simulation (Qin et al., 2024; Brohan et al., 2023; Huang et al., 2022; Li et al., 2025; Reed et al., 2022a; Yang et al., 2023; Hafner et al., 2025, *inter alia*). However, learning a specialised world model is challenging due to the scarcity of real-world data (Liu et al., 2024; Motamed et al., 2025). Conversely, a promising alternative is endowing pre-existing unified[2] vision-language models (VLMs) with world modelling abilities. In fact, VLMs are already imbued with plenty of real-world knowledge of both action (in language form) and perception (in vision form), because of their large-scale pre-training.

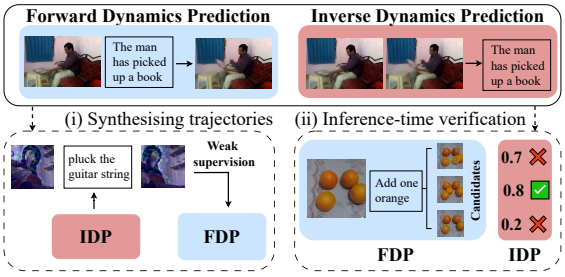


Figure 1: A high-level illustration of our two strategies to bootstrap Forward Dynamics Prediction from Inverse Dynamics Prediction in unified Vision–Language Models: (i) synthesising trajectories for weak supervision (**left**) and (ii) inference-time verification of candidate future observations (**right**).

Firstly, we probe whether unified VLMs already contain reliable forward dynamics models (FDP), i.e., the ability to predict the next image frame given the previous image frame and an action expressed as a language instruction. For this assessment, we limit ourselves to single-step trajectories, as a first step towards longer horizons. Based on our evaluation, we empirically demonstrate that existing VLMs do not prefer ground-truth trajectories compared to adversarially generated ones. Hence, we verify that the world model implicit in the original VLMs *per se* is not well grounded on real-world forward dynamics (Gao et al., 2024; Qiu et al., 2024; Abdou et al., 2021).

Surprisingly, we also find that predicting the ac-

[1]The code and models developed in this paper will be made available at `[anonymised]`.

[2]Here 'unified' means models capable of interleaving text and images during generation architecturally.

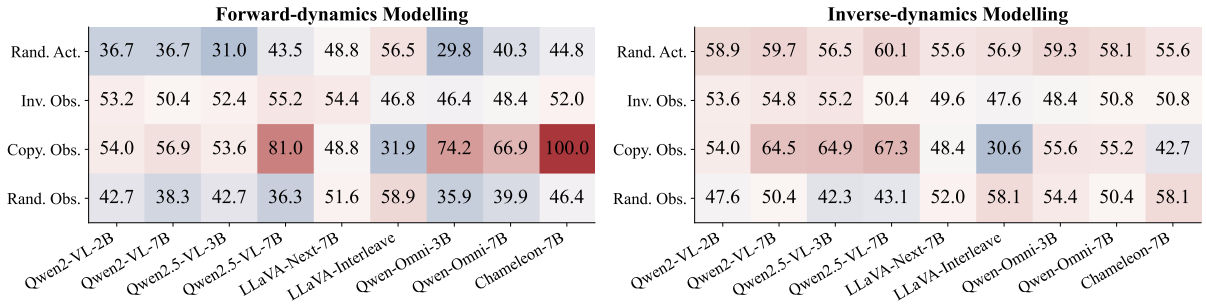


Figure 2: Percentage of times 9 VLMs assign higher probability to observation–action–observation Reference trajectories compared with 4 types of Negative (i.e., adversarially manipulated) trajectories, for both forward dynamics and inverse dynamics prediction. Higher values are better.

tion $a \in \mathcal{A}$ taking place between observations $o \in \mathcal{O}$—also known as *inverse dynamics prediction* (IDP; $\mathcal{O}_t \times \mathcal{O}_{t+1} \rightarrow \mathcal{A}_t$)—via supervised fine-tuning is substantially easier than FDP ($\mathcal{O}_t \times \mathcal{A}_t \rightarrow \mathcal{O}_{t+1}$). Inspired by this finding, we propose two strategies to bootstrap the FDP from the IDP in a unified VLM, namely (i) **learning from synthetic trajectories** in videos automatically labelled with actions by the dynamics model; and (ii) **test-time verification** of predicted observations sampled from the FDP through the IDP. Figure 1 illustrates our two strategies.

For the weak supervision strategy, we use a VLM fine-tuned for IDP on the AURORA dataset (Krojer et al., 2024) to annotate with linguistic actions ($\hat{a} \in \mathcal{A}$) motion key-frame pairs extracted from 45 hours of unlabelled real-world videos. These are sourced from movements-in-time (Monfort et al., 2019), Kinetics700 (Kay et al., 2017; Carreira et al., 2019) and UCF-101 (Soomro et al., 2012). Together with the ground-truth trajectories in AURORA, the synthesised trajectory triplets are then used for weakly supervised fine-tuning of the very same VLM for FDP. To encourage training to focus on image regions that change the most, we additionally propose a *loss-weighting method* which weights the loss of each image token according to the visual difference between the ground-truth previous and next observations, as estimated by a recognition model. Instead, in the verification strategy, we use the IDP to assign scores to multiple candidate samples generated by the FDP. Selecting the highest-score prediction can effectively guide search at inference time.

We conduct a thorough evaluation of both strategies to bootstrap FDP from IDP on several datasets from AURORA-BENCH (Krojer et al., 2024): MagicBrush, Action-Genome, Something-Something, WhatsUp and Kubric. We experiment with two families of unified VLMs for FDP, Chameleon-7B and Liquid-8B. Training on trajectories synthesised by IDP, our FDP can achieve an overall performance superior to state-of-the-art diffusion models specialised for image editing, both in terms of GPT4o-as-a-judge and human evaluation. Inference-time verification can also improve FDP to a comparable degree as trajectory synthesis, providing an effective training-free bootstrapping method.

While limited to single-step action–observation trajectories, this work offers promising early evidence that unified VLMs can be successfully endowed with FDP and hence may be suitably developed into long-horizon world models in the future.

2 Unified VLMs Lack a Consistent Preference for Real-World Trajectories

To understand whether off-the-shelf VLMs are already suitable for FDP, the first research question we investigate in this paper is: **To what extent do VLMs exhibit a preference for sequences of actions and observations that align with real-world trajectories?**

To address this question, we evaluate 9 different VLMs on ground-truth trajectories from 5 subsets of AURORA-BENCH (Krojer et al., 2024): MagicBrush, Something-Something, Action-Genome, Whatsup, and Kubric. Each subset contains 50 trajectory triplets of the form (o_s, a, o_t), where o_s is the source observation (image frame), a the action (text), and o_t the target observation.[3]

We then manually curate four types of negative trajectories using rule-based manipulations. The first kind is an action-level manipulation. 1) **Ran-**

[3]We choose these 9 VLMs using the following criteria: 1) they are publicly accessible and 2) they have been exposed to interleaved multimodal data during their pre-training.

dom Action: for a given pair of observations, we substitute the original action with another randomly sampled within the same subset. We also perform three observation-level manipulations. 2) **Random Observation**: we randomly substitute the target observation with another in the same subset. 3) **Copy Observation**: we copy the source observation as the target observation. 4) **Inverse Observation**: we swap the source and target observations.

In Figure 2, we compare the log-likelihood that VLMs assign to each ground-truth trajectory (Reference) against its corresponding manipulated one (Negatives). We evaluate the VLMs in two tasks: action prediction (i.e., as an inverse-dynamics model) and next-observation prediction (i.e., as a forward-dynamics model). For each kind of negative trajectory, we report the percentage of samples where the model favours the reference trajectory over the negative trajectory. From Figure 2, it emerges that VLMs display no clear preference for the ground-truth trajectories in a zero-shot setting (around 50%).

In the action prediction task (right panel), there is a slightly higher tendency to favour the ground-truth over the group with random actions; however, even in the best case, Qwen2.5-VL-7B prefers the reference in only 60.08% of the samples. The only negative group that seems to be identifiable for VLMs is observation copying, where Qwen2.5-VL-7B has 67.34% of correct preference. In the next-observation prediction task (left panel), the VLM mostly fails in effectively differentiating the ground truth from the negatives. Although the underlying reason remains uncertain, one plausible explanation for this behaviour is that the model's ability to solve next-observation prediction tasks depends on their alignment with training sequences: for instance, it is plausible that Chameleon's data rarely features two identical adjacent images. We provide a discussion breaking down Chameleon's performance in Appendix D.

3 Bootstrapping FDP from IDP

Overall, the results from Section 2 reveal that off-the-shelf VLMs are not suitable for FDP and IDP *per se*; however, they also show that IDP is a more feasible task, as VLMs already achieve accuracies above random chance in a zero-shot setting. Possibly, IDP abilities may be even improved with a small amount of fine-tuning. This underlies the key intuition behind our work: **can we leverage this asymmetry to bootstrap forward-dynamics abilities from inverse-dynamics ones within a unified VLM?**

We propose two strategies to leverage VLMs fine-tuned for IDP to enhance their own FDP abilities: (i) generating synthetic trajectories by annotating large-scale key-frame pairs from videos with actions predicted by IDP, then using these as weak supervision to train for FDP (Section 3.2); and (ii) using the IDP as a verifier at test time to score candidate next observations sampled from the FDP (Section 3.3).

3.1 Inverse-dynamics Modelling

First, we fine-tune the unified VLM as an inverse-dynamics model (**IDM**) $p_{\text{IDM}}(a \mid o_s, o_t)$, which predicts the probability of an action given the previous and next observations. As training data, we rely on high-quality triplets from AURORA (Krojer et al., 2024) and the action recognition track of EPIC-Kitchen (Damen et al., 2018), which is based on videos with an egocentric view. We use the first and last frame in the EPIC-Kitchen video clips as the source and target observation o_s and o_t and the annotated action as a. We provide full details on IDM training data and setting in Appendix H.1.

3.2 Weakly Supervised Learning from Unlabelled Videos

Synthetic Trajectories. Taking advantage of the resulting high-quality IDM, we then explore the first of our strategies to bootstrap the FDP in VLMs: we annotate pairs of motion key-frames of unlabelled videos with a textual description of the action with the IDM. To ensure both scale and quality, we collect approximately 45 hours of video from Moments-in-Time (Monfort et al., 2019), Kinetics-700 (Kay et al., 2017; Carreira et al., 2019), and UCF-101 (Soomro et al., 2012), all of which consist of curated clips focused on human actions. To ensure the selected pairs of motion key-frames are meaningful, i.e., they express a valid action, we then calculate the optical flow to quantify the dynamics per frame in the video clips, and select the top-K_f frames while ensuring that the interval between two selected frames is I_f. Specifically, we set $I_f = 20$ and $K_f = 6$ for all three datasets. This results in approximately 20K, 46K, and 21K annotated trajectory triplets from Moments-in-Time, Kinetics-700, and UCF-101, respectively. Finally, we apply a filtering strategy to further guarantee the quality of the resulting triplets. Specifically, we

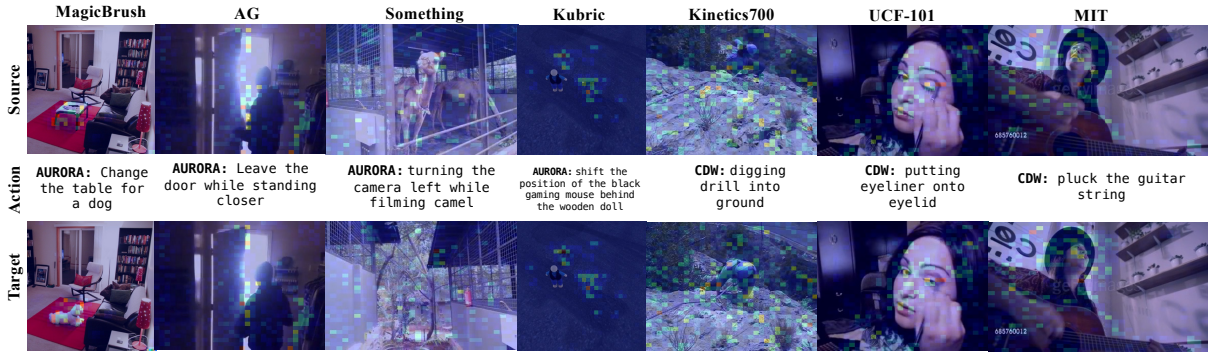


Figure 3: Heatmap visualization of image token weights predicted by the recognition model on examples from AG, Something-Something, MagicBrush, and Kubric, and UCF-101, Kinetics700 and MIT.

apply stratified top-k sampling based on the IDM's predicted likelihood for each trajectory triplet (o_s, $\hat{a}_{\text{IDM}}$, o_t)[4] to select a subset of triplets. We show statistics of the scores and action classes for the selected triplets in Figure 9. We also provide one example for each dataset in Figure 3.

FDP. Finally, we fine-tune a VLM as a forward-dynamics model (**FDM**), $p_{\text{FDM}}(o_t \mid a, o_s)$ on both AURORA's supervised triplets $\mathcal{D}_{\text{sup}}$ and unsupervised triples $\mathcal{D}_{\text{unsup}}$ with actions sampled from the IDM. Normally, a FDM would be trained with maximum likelihood estimation as an objective:

$$\min_{\theta} \mathbb{E}_{(a,o_s,o_t)\sim\mathcal{D}_{\text{sup}}}\left[-\log p_\theta(o_t \mid a, o_s)\right] + \mathbb{E}_{(o_s,o_t)\sim\mathcal{D}_{\text{unsup}}}\left[\mathbb{E}_{\hat{a}\sim p_{\text{IDM}}(a|o_s,o_t)}\left[-\log p_\theta(o_t \mid \hat{a}, o_s)\right]\right] \tag{1}$$

where θ are the parameters for FDM, and $\hat{a}$ is an action sampled from the IDM.

Recognition-Weighted Training Loss. Nevertheless, the objective in Equation 1 is limited by treating all regions (i.e., image patches) of the target observation equally, even if some of them remain identical to the source whereas others change. This may result in degenerate solutions such as always copying the source. As an alternative, we therefore propose a novel training objective for FDM that overcomes this assumption. This objective weights the loss of next-observation image tokens based on their importance. The intuition is that not all image patches in source and target observations contribute equally to modelling real-world transitions; instead, the model should focus on patches most indicative of the action's consequences. To this end, we leverage a recognition model $f_{\text{rec}}(w; o_s, o_t)$, which outputs token-level weights that represent the similarity between source and target patches. These weights modulate the loss during training, emphasising learning on semantically meaningful regions and down-weighting irrelevant ones. We formulate our objective as:

$$\min_{\theta} \sum_{l=1}^{L} f_{\text{rec}}(w; o_s, o_t)^{(l)} \cdot \left(-\log p_\theta(o_t^{(l)} \mid o_t^{(<l)}, o_s, a)\right), \tag{2}$$

where θ are the parameters of VLM as FDM and a L is the number of image tokens. $o_t^{(l)}$ and $o_t^{(<l)}$ represent the image tokens of o_t at position l and the history of previous positions, respectively. For simplicity, we use the pre-trained vector-quantised encoder of the unified VLMs as the recognition model, by computing the squared L_2 norm of pre-quantized features $\mathbf{z}_{o_s} \in Z_{o_s}$ and $\mathbf{z}_{o_t} \in Z_{o_t}$ where Z_{o_s} and Z_{o_t} are the sets of features of source and target observations, respectively. We visualise the token weights in Figure 3, which capture where acting on the source observation yields the largest effects on the target one.

3.3 Test-time Verification

Finally, we introduce an inference-time strategy which harnesses the IDM as a verifier to enhance FDM performance. Inspired by recent work on scaling test-time compute (Muennighoff et al., 2025; Snell et al., 2024), we let the FDM generate N candidate observations. Each candidate is paired with the source and scored by the IDM, which assigns each a predicted likelihood, interpreted as a reward. The final prediction of the FDM is selected by maximising the IDM's reward:

$$\hat{o}_t = \underset{i\in\{1,\ldots,N\}}{\text{argmax}}\; p_{\text{IDM}}\left(a \mid o_s, o_t^{(i)}\right), \quad \text{where } o_t^{(i)} \sim p_{\text{FDM}}(o_t \mid o_s, a).$$

[4]The details of this algorithm are provided in Appendix E.

where $\hat{o}_t$ is the selected prediction. While this strategy is training-free, it requires sampling multiple candidates at inference time.

4 Experiments and Results

In Section 4.1, we first describe the experimental setup, including benchmarks, baselines, and evaluation metrics. We then report results on the inverse dynamics prediction task in Section 4.2 to verify that fine-tuning on a limited amount of examples is sufficient to develop robust IDP in a VLM. This is followed in Section 4.3 by automatic and human evaluation of both strategies introduced in Section 3 to bootstrap FDP, as well as ablation studies. Finally, we analyse inference-time verification and examine the transfer of forward dynamics prediction to two spatial reasoning benchmark.

4.1 Experimental Setting

Benchmarks. We select AURORA-BENCH (Krojer et al., 2024) for evaluation of both IDM and FDM. This dataset provides high-quality data for action-centric edits, covering a wide array of phenomena and assessing a model's alignment with the physical world. We choose 5 subsets: **MagicBrush** for specialised image editing, **Action-Genome (AG)** and **Something-Something (Something)** for real-world actions. **Whatsup** focuses on spatial reasoning, whereas **Kubric** contains samples from a physical engine (Greff et al., 2022).

Models and Baselines. We experiment with two unified VLMs, Chameleon-7B (Chameleon Team, 2024) and Liquid-8B (Wu et al., 2024b). The variants of these models fine-tuned on both supervised and weakly supervised data with loss weighting (i.e., with the training-time bootstrapping strategy) are indicated as **C-FDM** and **L-FDM**, respectively. As our first baselines, we use the same two VLMs either zero-shot (**C-ZS** and **L-ZS**) or fine-tuned *only* on AURORA's supervised data (**C-FT** and **L-FT**). Additionally, we include three state-of-the-art diffusion models specialised for image editing as baselines, such as **PixInstruct** (Brooks et al., 2023), **GoT** (Fang et al., 2025) and **SmartEdit** (Huang et al., 2024). Finally, as a sanity check, we also report the metric scores obtained by simply copying the source observation input as the next-observation prediction (**Copy**).

Metrics. For next-observation prediction (FDP) evaluation, following Fang et al. (2025), we rely on **GPT4o-as-a-judge** as it is the only metric that reliably penalises Copy. In Appendix B, we show four other metrics, e.g., CLIP, which assign high scores to Copy. GPT4o-as-a-judge scores consider two criteria, one for the editing success rate and one for visual consistency with the original. We take the minimum of the two as the final score. The prompt for the judge is provided in Appendix F.

	BS	R-1	R-L	BLEU
Chameleon ZS	0.05	0.09	0.08	0.00
Chameleon IDM	**0.45**	**0.45**	**0.44**	0.20
Liquid IDM	0.41	0.41	0.36	**0.21**

Table 1: Performance of Inverse Dynamics Models on action prediction, measured by text similarity metrics: BERTScore (BS; Zhang et al. 2020), ROUGE (R-1, R-L; Lin 2004) and BLEU (Papineni et al., 2002).

4.2 Inverse Dynamics Prediction

We evaluate the IDM based on the textual similarity of the predicted action with the ground-truth action in AURORA-BENCH. Results are shown in Table 1. Our results demonstrate that a moderate amount of fine-tuning is necessary to let unified VLMs verbalise the dynamics connecting two observations, as evidenced by the large It is also worth noting that both models' IDP performance (in terms of BLEU) is similar. We leverage these IDP versions of Chameleon and Liquid to bootstrap FDP, whose results are reported in Section 4.3.

4.3 Forward Dynamics Prediction

Next, we report FDP results for each of the two bootstrapping strategies: in Section 4.3.1 for trajectory synthesis and in Section 4.3.2 for inference-time verification.

4.3.1 Synthesising Trajectories with IDP

Automatic evaluation. To test trajectory synthesis, we evaluate FDM on next-observation prediction for each of the AURORA-BENCH subsets, reporting GPT4o-as-a-judge scores in Table 2. We first notice that the state-of-the-art image editing models (i.e., PixInstruct, GoT, SmartEdit) tend to specialise on the proper image editing subset MagicBrush (5.96 and 6.71 GPT4o scores for GoT and SmartEdit). Nevertheless, in the action-centric subsets, including Action-Genome (AG), Something and Kubric, they are mostly behind bootstrapped models (C/L-FDM) and even the fine-tuned baselines (C/L-FT). Crucially, considering the average performance of C/L-FDM on all subsets of

Dataset	Models								
	PixInst	GoT	SE	C-ZS	C-FT	C-FDM	L-ZS	L-FT	L-FDM
Forward-dynamics Prediction (GPT4o score)									
MagicBrush	3.12	5.96	**6.71**	0.00	2.52	3.92	0.68	5.56	5.82
AG	1.20	1.61	3.08	0.17	2.48	**3.64**	0.08	2.59	2.98
Something	0.96	2.62	2.81	0.37	**3.11**	2.92	0.42	2.72	2.88
WhatsUp	0.00	1.58	0.76	0.15	0.88	0.54	0.82	2.88	**3.30**
Kubric	1.88	3.92	3.70	0.14	7.30	**7.32**	2.22	6.28	6.60
AURORA-BENCH Avg.	1.43	3.14	3.41	0.17	3.26	3.67	0.84	4.04	**4.32**
Spatial Reasoning (Accuracy)									
SpatialMQA	–	–	–	26.1	25.8	27.2	27.7	**28.0**	27.8
EmbodiedSpatial-Bench	–	–	–	15.1	21.2	17.5	32.6	33.2	**33.8**

Table 2: Performance on AURORA-BENCH (GPT-4o-as-a-judge score) and spatial reasoning benchmarks (accuracy). We **bold** the best model per dataset. Among our variants, we highlight the best and worst scores. SE: SmartEdit. Greedy decoding is used for spatial reasoning evaluation and applied only to VLMs.

	Weighted		Standard	
	ES (↑)	ME (↑)	ES (↑)	ME (↑)
MB	3.73	8.17	3.68	8.46
AG	3.18	8.03	2.37	8.13
ST	3.32	7.01	2.78	7.20
WU	0.54	7.25	0.76	7.19
KU	7.75	8.49	7.24	8.70
Avg.	3.71	7.80	3.37	7.94
GPT4o	3.67		3.58	

Table 3: Detailed scores of GPT4o-as-a-judge evaluation for loss-weighting and standard training. We report the scores for **E**diting **S**uccess (**ES**) and **M**inimal **E**diting (**ME**). MB: MagicBrush, AG: Action-Genome, ST: Something-Something, WU: WhatsUp, KU: Kubric. We highlight the best and worst scores for each category. We report the average of ES and ME as **Avg.** and the final score as **GPT4o**.

AURORA-BENCH reveals the benefit of augmenting the training data with synthetic triplets bootstrapped from the IDM (vs. FT), as it yields a 13% gain for Chameleon and 7% for Liquid, and the benefit of fine-tuning off-the-shelf VLMs for FDP more broadly (vs. ZS).

Human Evaluation. Following Krojer et al. (2024), we conduct a blind human evaluation comparing GoT, SmartEdit, C-FT, and C-FDM. We randomly sample 5 examples from each subset within AURORA-BENCH and present the outputs generated by each of the four models. Human annotators are asked to identify the best and worst generated observations based on three criteria: (1) *Realism*: the generated image should exhibit natural textures and lighting while remaining faithful to the input scene; (2) *Instruction-Following Ability*: the edit should clearly reflect the given action; and (3) *Over-Editing*: the modification should be minimal and focused, altering only what is necessary to fulfil the action. Each model receives +1 point for being selected as the best, -1 for the worst, and 0 otherwise. We compute the average scores over 350 annotated samples, as reported in Table 5. The results are well aligned with automatic evaluations in Table 2: image-editing models excel in the MagicBrush and WhatsUp subsets, but fall short on action-centric datasets such as Action-Genome, Something-Something, and Kubric. In contrast, C-FDM outperforms C-FT (and all other baselines) on all three of these datasets, highlighting its strength in next-observation prediction in real-world, action-centric trajectories despite remaining a general-purpose VLM.

Ablation Study on Synthetic Trajectories. To assess the importance of extra supervision from IDM-synthetic trajectories, Table 4 reports GPT-4o's scores for this ablation. We see performance drops on most datasets—particularly on Something and AG—when the additional training data from unlabelled videos is removed, highlighting the effectiveness of bootstrapping C-FDM with large-scale real-world data via IDM. An exception is the WhatsUp dataset, which focuses on specific actions within a fixed scene; in this case, training in an open-domain setting may not transfer effectively.

Ablation Study on Loss Weighting. Based on Table 4, we also observe consistent degradation when loss weighting is removed, demonstrating the benefit of explicitly incorporating the recognition model into visual next-token prediction. To better understand the effect of loss weighting, Table 3 re-

	C-FDM	*w/o Synth.*	*w/o LW*
MB	**3.48**	-0.28	-0.22
AG	**3.02**	-0.35	-0.08
ST	**3.06**	-0.18	-0.19
WU	**0.46**	0.40	0.08
KU	**7.14**	-0.03	-0.33
All	**3.43**	-0.09	-0.15

Table 4: Ablation study of synthetic trajectories (Synth.) and loss weighting (LW) in C-FDM. Numbers are GPT-4o-as-judge scores (↑, average of 3 runs). MB: MagicBrush, AG: Action-Genome, ST: Something-Something, WU: WhatsUp, KU: Kubric.

	GoT	SE	C-FT	C-FDM
MB	0.06†	**0.29**†	-0.32†	-0.03
AG	-0.23†	-0.46†	0.32	**0.37**
ST	0.00	-0.37†	0.18	**0.20**
WU	**0.25**	-0.38†	0.14	0.00
KU	-0.52†	-0.22†	0.34	**0.40**
All	-0.09†	-0.23†	0.13	**0.19**

Table 5: Human evaluation results. † indicates all results whose gap with respect to C-FDM is significant, based on a Wilcoxon signed-rank test ($p = 0.05$). MB: MagicBrush, AG: Action-Genome, ST: Something-Something, WU: WhatsUp, KU: Kubric.

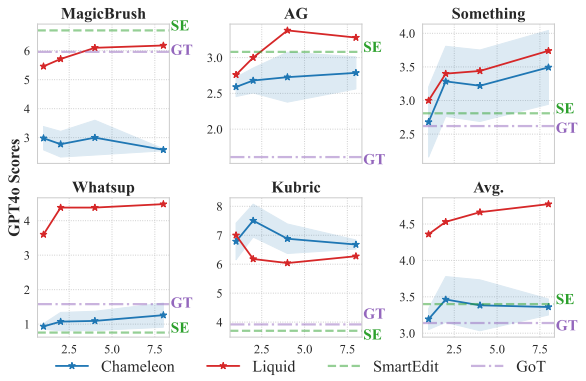


Figure 4: GPT-4o scores for test-time verification with K samples, where $K \in \{1, 2, 4, 8\}$. We use a blue line for C-FT and a red line for L-FT. For C-FT, we plot the standard deviation as the shaded area due to its large variance. We indicate the scores for GoT (GT) and SmartEdit (SE) as horizontal lines.

ports the average scores for two criteria used in the GPT-4o-as-a-judge evaluation separately: Editing Success (ES), which measures how well the model captures the intended action and performs the corresponding edit, and Minimal Editing (ME), which assesses whether the model introduces unnecessary modifications. The full distribution of GPT-4o scores is provided in Appendix G. Our analysis reveals that the primary bottleneck for C-FDM remains its ability to reliably follow the instruction, as reflected by the fact that ES scores are significantly lower than ME scores. Loss weighting partly solves this problem, increasing the editing success and reducing copying behaviour, albeit at the cost of sometimes over-editing the source observation.

Image Editing as an Auxiliary Task. Finally, we assess whether enhanced FDP capabilities are beneficial for a broader range of vision–language tasks. Since AURORA and the action-annotated videos contain diverse spatial relations (e.g., left-/right orientation), we evaluated whether this supervision helps VLMs generalise beyond editing. We tested our models on two spatial reasoning benchmarks: SpatialMQA (Liu et al., 2025) and EmbodiedSpatial-Bench (Du et al., 2024), and report the corresponding accuracy in Table 2. Both Chameleon and Liquid trained with the FDP objective outperform the zero-shot baseline, demonstrating that the FDP task transfers beyond action-centric editing and highlighting FDP as a signal for enhancing spatial reasoning.

4.3.2 Inference-time Verification

We evaluate our test-time strategy using IDM to choose among candidates generated by C/L-FT in Figure 4, using $K \in 1, 2, 4, 8$. By increasing exploration on more candidate next observations, Chameleon and Liquid benefit from test-time verification on most datasets with real-world trajectories (e.g., AG, Something, WhatsUp), indicating the reliability of IDM's trajectory preferences. Increasing K does not always improve performance (MagicBrush, Kubric), however, suggesting that bootstrapping with IDM that shares the same foundation model backbone may be limiting. In summary, IDM-based verification boosts performance to a similar level as FDM, by leveraging more samples during inference rather than training.

Qualitative Example. Figure 5 presents a real-world example demonstrating that C-FDM is also capable of iteratively generating future observations in multiple steps while maintaining consistency with previous frames.

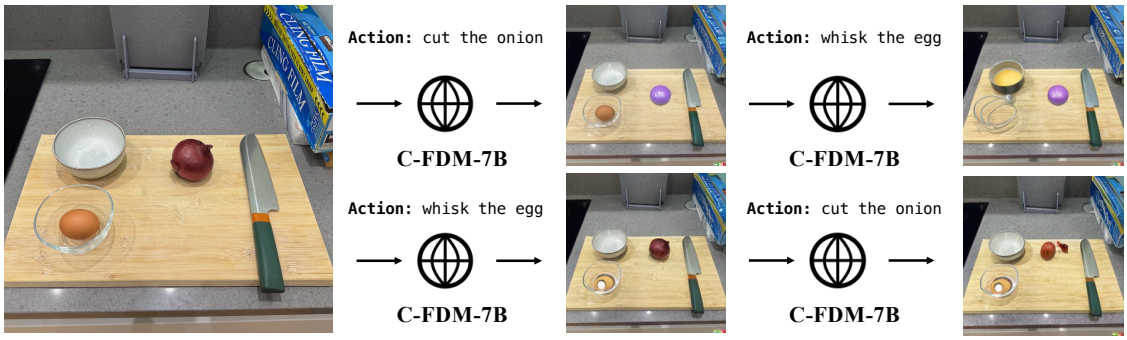


Figure 5: A qualitative case of real-world next-observation prediction, demonstrating C-FDM's ability to steer predictions using language and perform *sequential* predictions. More cases from AURORA-BENCH are in Appendix C.

5 Related Work

Despite the surge in interest for world modelling (Ha and Schmidhuber, 2018; Sutton, 1988; Hafner et al., 2019a), previous works focused mostly on building specialised *ad-hoc* world models. These world models can be explicitly learnt as a visual simulator (Agarwal et al., 2025; Bruce et al., 2024; Brooks et al., 2024), or enable planning with model-based reinforcement learning (Hafner et al., 2019b; Micheli et al., 2022; Robine et al., 2023; Alonso et al., 2024; Hafner et al., 2025). Instead, we focus on leveraging general-purpose, pre-trained vision-language models (Lin et al., 2024; Chen et al., 2025; Wu et al., 2024c) to develop world models, which is more attractive due to the inductive bias they provide from their extensive training.

This is possible thanks to frameworks that integrate observations, actions, and rewards into a unified sequence of tokens in autoregressive Transformers (Wu et al., 2024a), building on pioneering works such as Decision Transformers (Chen et al., 2021) and GATo (Reed et al., 2022b). Related to our work, Chen et al. (2024) initialise the parameters of RL policies with VLMs, thus taking advantage of the abundant and general world knowledge encoded in their representations. 3D-VLA (Zhen et al., 2024) integrates a set of interaction tokens into a Large Language Model to engage with the environment as an embodied agent. Yang et al. (2024) and Soni et al. (2024) explore large-scale self-supervised learning via next token or frame prediction to build a unified model absorbing internet knowledge, learning from interaction via video.

Most similar to our work, Baker et al. (2022) train a dynamics model which aims to uncover the underlying action between video frames in unlabelled video frames from the Minecraft game. Through this model, they synthesise trajectories to train a policy for sequential decision making. In contrast with Baker et al. (2022), we focus on actin-centric next-observation prediction as a task to evaluate FDP. First, this allows us to port the observation space to real-world frames, rather than simulated ones, hence assessing whether VLMs can eventually be developed into world models. Second, this broadens the space of actions from a few choices to the combinatorially infinite and expressive space of language, capturing a significantly more diverse range of dynamics.

6 Conclusions

In this work, we explored whether unified vision–language models (VLMs) can be endowed with the ability to predict forward dynamics, i.e., the next observation in the environment (e.g., an image frame) given the past observation and an action (e.g., a textual instruction). We first show that these models lack a clear preference for ground-truth real-world action–observation trajectories compared with adversarially manipulated ones.

To address this, we leverage an inverse-dynamics model (IDM) fine-tuned from the same VLM, which consists instead of predicting actions taking place between observations and is easier to learn, to bootstrap a better forward-dynamics model (FDM). Specifically, the IDM can be used to 1) automatically annotate pairs of frames from unlabelled videos, which are then used for weakly supervised training of the FDM; or 2) verify the best sample among multiple candidates generated from the FDM at inference time. Experiments confirm the effectiveness of both strategies, with our general-purpose forward-dynamics model achieving state-of-the-art performance compared to existing approaches specialised for image editing.

Limitations

While overall our results demonstrate the effectiveness of our approaches across AURORA-BENCH, we would like to highlight a few limitations that we have discovered:

- Despite efforts to guide the model via weakly supervised fine-tuning with loss weighting or inference-time verification (Table 2), we observe that the model may still resort to copying the source observation, especially under low sampling temperatures or ambiguous instructions.
- While we show promising preliminary results of language-steered next-observation prediction in Figure 5, fine-grained control remains limited, and understanding subtle instructions (e.g., spatial or quantitative edits) remains challenging.
- We observe high variance across different runs of experiments for Chameleon, likely due to the sensitivity of sampling from a weak model. To address this, we report results averaged over multiple runs.

References

Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. 2021. Can language models encode perceptual structure without grounding? A case study in color. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 109–132, Online. Association for Computational Linguistics.

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, and 1 others. 2025. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. 2024. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. 2022. Video PreTraining (VPT): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654.

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, and 1 others. 2023. Do as I can, not as I say: Grounding language in robotic affordances. In *Conference on robot learning*, pages 287–318. PMLR.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. InstructPix2Pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, and 1 others. 2024. Video generation models as world simulators. 2024. *URL https://openai. com/research/video-generation-models-as-world-simulators*, 3:1.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, and 1 others. 2024. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*.

Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. 2019. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*.

Chameleon Team. 2024. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.

William Chen, Oier Mees, Aviral Kumar, and Sergey Levine. 2024. Vision-language models provide promptable representations for reinforcement learning. In *Automated Reinforcement Learning: Exploring Meta-Learning, AutoML, and LLMs*.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. 2025. Janus-Pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*.

Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. 2024. ANOLE: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and 1 others. 2018. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736.

Mengfei Du, Binhao Wu, Zejun Li, Xuan-Jing Huang, and Zhongyu Wei. 2024. Embspatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 346–355.

Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, and 1 others. 2025. GoT: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *arXiv preprint arXiv:2503.10639*.

Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. 2024. Physically grounded vision-language models for robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12462–12469. IEEE.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, and 1 others. 2022. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761.

David Ha and Jürgen Schmidhuber. 2018. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2019a. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2019b. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2025. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, and 1 others. 2022. Inner Monologue: Embodied reasoning through planning with language models. In *6th Annual Conference on Robot Learning*.

Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, and 1 others. 2024. SmartEdit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8362–8371.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, and 1 others. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Benno Krojer, Dheeraj Vattikonda, Luis Lara, Varun Jampani, Eva Portelance, Christopher Pal, and Siva Reddy. 2024. Learning Action and Reasoning-Centric Image Editing from Videos and Simulations. In *NeurIPS*. Spotlight Paper.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5971–5984.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. 2024. World model on million-length video and language with blockwise RingAttention. *arXiv preprint arXiv:2402.08268*.

Jingping Liu, Ziyan Liu, Zhedong Cen, Yan Zhou, Yinan Zou, Weiyan Zhang, Haiyun Jiang, and Tong Ruan. 2025. Can multimodal large language models understand spatial relations? *arXiv preprint arXiv:2505.19015*.

Vincent Micheli, Eloi Alonso, and François Fleuret. 2022. Transformers are sample-efficient world models. In *Deep Reinforcement Learning Workshop NeurIPS 2022*.

Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, and 1 others. 2019. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–8.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. 2025. Do generative video models learn physical principles from watching videos? *arXiv preprint arXiv:2501.09038*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Yiran Qin, Zhelun Shi, Jiwen Yu, Xijun Wang, Enshen Zhou, Lijun Li, Zhenfei Yin, Xihui Liu, Lu Sheng, Jing Shao, and 1 others. 2024. WorldSimBench: Towards video generation models as world simulators. *arXiv preprint arXiv:2410.18072*.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay B Cohen. 2024. Are large language model temporally grounded? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7057–7076.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, and 1 others. 2022a. A generalist agent. *arXiv preprint arXiv:2205.06175*.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, and 1 others. 2022b. A generalist agent. *arXiv preprint arXiv:2205.06175*.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. 2023. Transformer-based world models are happy with 100k interactions. *arXiv preprint arXiv:2303.07109*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Achint Soni, Sreyas Venkataraman, Abhranil Chandra, Sebastian Fischmeister, Percy Liang, Bo Dai, and Sherry Yang. 2024. VideoAgent: Self-improving video generation. *arXiv preprint arXiv:2410.10076*.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Richard S Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. 2024a. iVideoGPT: Interactive VideoGPTs are scalable world models. In *Advances in Neural Information Processing Systems*, volume 37, pages 68082–68119. Curran Associates, Inc.

Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. 2024b. Liquid: Language models are scalable and unified multi-modal generators. *arXiv preprint arXiv:2412.04332*.

Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, and 1 others. 2024c. VILA-U: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. 2023. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2):6.

Sherry Yang, Jacob Walker, Jack Parker-Holder, Yilun Du, Jake Bruce, Andre Barreto, Pieter Abbeel, and Dale Schuurmans. 2024. Video as the new language for real-world decision making. *arXiv preprint arXiv:2402.17139*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 2024. 3D-VLA: A 3D Vision-Language-Action Generative World Model. In *International Conference on Machine Learning*, pages 61229–61245. PMLR.

A Potential Risks

This work develops models for action-centric image editing for visual world modelling. While our primary aim is to advance fundamental research in world modelling, we acknowledge potential risks, particularly in the generation of realistic future observations.

A core concern is the potential misuse of the models for creating deceptive visual content, including fabricated action sequences or manipulated images that imply false causality. Although the model is not explicitly designed for these tasks, its ability to generate coherent visual predictions from the linguistic action could be adapted for such uses if deployed irresponsibly.

Even in intended use, risks include over-reliance on generated outputs in downstream tasks such as robotic control, or interactive systems. Model failures—e.g., copying artefacts, hallucinations, or broken object continuity—can lead to incorrect inferences or reinforce dataset biases.

To mitigate potential misuse, we limit our model release to research purposes under a non-commercial license and clearly communicate its capabilities and limitations. We urge caution when adapting them for deployment, particularly in settings with high societal or ethical sensitivity.

B Model Performance on AURORA-BENCH with 5 Metrics

In addition to GPT4o-as-a-judge evaluation, we further employ a diverse set of automatic metrics covering both low-level and semantic fidelity: 1) we compute the **L1 distance** between the predicted and target observation as a pixel-level metric. 2) We extract visual features and compute the cosine similarity in their respective embedding spaces for several image encoders, including (**CLIP-I** and **DINO**), to assess semantic similarity. Additionally, to measure alignment between image content and the action semantics, we compute **CLIP-T**, the similarity between the edited image and its BLIP-generated caption. These metrics are evaluated in addition to GPT4o-as-a-judge metric following previous works in image editing (Huang et al., 2024; Fang et al., 2025; Krojer et al., 2024). We report the detailed results with 5 metrics in Table 6. We notice that copy baseline exhibits the best performance as measured by the distance-based and visual encoder-based approach, as indicated in Table 2. This poses a challenge to the reliability of the traditional metrics in fairly evaluating the action-centric image editing task. On the other hand, GPT4o-as-a-judge metric robustly assigns 0 score to Copy, indicating its robustness in detecting copying generation while putting GPT-as-a-judge as the most reliable metric to interpret.

C Qualitative Cases

In this section, we present additional qualitative examples from AURORA-BENCH in Figure 6. We observe several common failure modes in image editing models. First, they sometimes fail to preserve the scene from the source observation (e.g., PixInstruct on Action-Genome and MagicBrush). Second, some models generate near-identical copies of the source as the target (e.g., GoT on Something-Something). Third, producing realistic outputs remains difficult, as seen in GoT's result on Kubric. Finally, maintaining object consistency is also a challenge—SmartEdit alters the object in WhatsUp, and C-FDM does so in Something-Something.

Despite the challenges, we also observe several positive editing behaviours from C-FDM. On Action-Genome, C-FDM correctly predicts spatial changes, such as *opening and closing a drawer*, which requires a strong understanding of the spatial concepts. In Something-Something, it is the only model to accurately capture the spatial concept of "falling down." On Kubric, it demonstrates basic counting ability by correctly adding one keyboard. In WhatsUp, C-FDM correctly grounds the action to the laptop, while other models mistakenly edit the monitor.

D Detailed Discussion for Chameleon's Predicted Likelihoods

From Figure 7, it emerges that Chameleon-7B displays a very limited preference for the ground-truth trajectories in a zero-shot setting. In the action prediction task (top panel), there is a slightly higher tendency to favour the ground-truth; however, even in the best case (counterfactual action), the model prefers the reference in only 58.1% of the samples. The high correlation in likelihoods indicates that the VLM struggles also on visual manipulations. In the next-observation prediction task (bottom panel), the VLM mostly fails in effectively differentiating the ground truth from the negatives. An exception to this is the copy manipulation, where the model can always tell them apart. Although the underlying reason remains uncertain, one plausible

Datasets	Metrics	Models								
		Copy	**PixInstruct**	**GoT**	**SE**	**CM**	**C-FT**	***+Best-of-3***	**C-FDM**	***+Best-of-3***
MagicBrush	L1	0.027	0.114	**0.063**	0.068	0.287	0.075	0.075	0.090	0.078
	CLIP-I	0.959	0.877	0.930	**0.937**	0.671	0.913	0.914	0.906	0.909
	CLIP-T	0.289	0.275	0.286	0.290	0.227	0.289	0.289	**0.291**	0.291
	DINO	0.931	0.761	0.881	**0.894**	0.292	0.883	0.883	0.864	0.864
	GPT-4o	0.000	3.120	5.960	**6.710**	0.000	2.520	3.270	3.920	3.920
AG	L1	0.069	0.220	0.174	**0.137**	0.314	0.170	0.168	0.168	0.167
	CLIP-I	0.943	0.757	0.846	0.811	0.609	0.872	0.872	**0.881**	0.883
	CLIP-T	0.279	0.254	0.280	0.268	0.214	0.280	0.284	**0.284**	0.284
	DINO	0.929	0.557	0.785	0.774	0.258	0.801	0.817	**0.816**	0.816
	GPT-4o	0.000	1.200	1.610	3.080	0.170	2.480	2.740	**3.640**	3.640
Something	L1	0.135	0.232	0.184	**0.163**	0.293	0.184	0.184	0.196	0.184
	CLIP-I	0.870	0.709	0.807	0.773	0.649	**0.820**	0.820	0.804	0.804
	CLIP-T	0.275	0.238	0.269	0.265	0.232	**0.271**	0.269	0.268	0.268
	DINO	0.797	0.453	0.636	0.662	0.297	**0.675**	0.653	0.666	0.666
	GPT-4o	0.000	0.957	2.620	2.810	0.370	**3.110**	3.110	2.920	3.310
WhatsUp	L1	0.039	0.138	0.078	0.067	0.251	**0.066**	0.066	0.070	0.070
	CLIP-I	0.954	0.817	**0.923**	0.888	0.721	0.877	0.880	0.870	0.883
	CLIP-T	0.326	0.287	**0.316**	0.312	0.243	0.309	0.310	0.306	0.307
	DINO	0.908	0.615	**0.850**	0.805	0.424	0.836	0.841	0.831	0.838
	GPT-4o	0.000	0.000	**1.580**	0.755	0.146	0.880	0.980	0.540	0.540
Kubric	L1	0.011	0.104	**0.026**	0.064	0.276	0.044	0.044	0.044	0.044
	CLIP-I	0.963	0.796	0.895	0.868	0.660	0.897	0.899	**0.897**	0.898
	CLIP-T	0.282	0.259	0.281	0.271	0.213	**0.287**	0.287	0.287	0.288
	DINO	0.955	0.676	0.857	0.798	0.161	**0.906**	0.906	0.902	0.902
	GPT-4o	0.000	1.880	3.920	3.700	0.140	7.300	7.300	**7.320**	7.780
All	GPT-4o	0.000	1.430	3.140	3.410	0.165	3.260	3.480	**3.670**	3.840

Table 6: Model performance at MagicBrush, Action-Genome, Something, WhatsUp and Kubric on AURORA-BENCH. For C-FT and C-FDM We report both the model performance and their performance in the *best-of-N* distribution. We report the average GPT4o scores for each model at the bottom. We highlight the better GPT-4o scores for C-FT and C-FDM. We **bold** the best performance among all models, except Copy and *best-of-N* performances. SE: SmartEdit.

explanation for this behaviour is that the model's ability to solve next-observation prediction tasks depends on their alignment with training sequences: for instance, it is plausible that Chameleon's data rarely features two identical adjacent images. In Figure 8, we visualize the predicted likelihoods produced by Chameleon's fine-tuned inverse dynamics model (IDM). We observe that fine-tuning on ground-truth trajectories substantially increases the model's ability to distinguish ground-truth actions from negative alternatives. Specifically, the probability that the ground-truth action is assigned a higher likelihood increases from 55.6% to 73.2% under random-action negatives, and from 58.1% to 72.2% under counterfactual-action negatives. These results demonstrate the strong potential of learning effective inverse dynamics models directly from real-world trajectories. In summary, the zero-shot Chameleon-7B does not exhibit a preference for ground-truth trajectories over negative ones, constructed through action- or observation-based manipulations. However, it is possible to learn an effective IDM from the real-world trajectories.

E Details of Processing IDM Annotations for Unlabelled Videos

We present the raw dataset statistics before sampling for Movements-in-Time, UCF-101 and Kinetics700 in Table 7. Figure 9 shows the distribution of IDM's predicted scores across action classes in Movements-in-Time, Kinetics700, and UCF-101. The predicted likelihoods are nearly uniform within each class, indicating that our sampling method maintains both class diversity and high overall likelihoods. The sampling procedure for IDM-annotated trajectories is detailed in Algo-

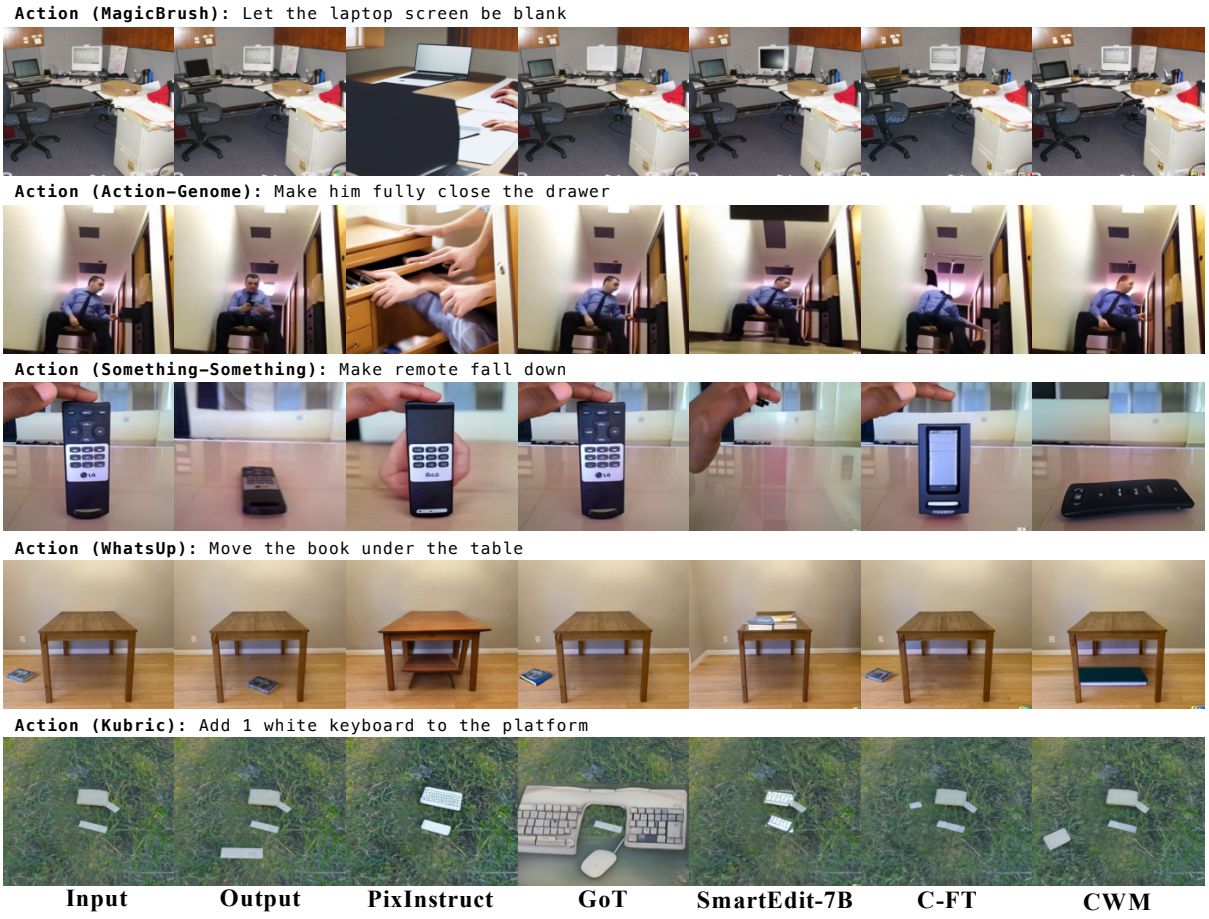


Figure 6: Qualitative examples of the predicted next observation from the state-of-the-art specialised image editing models, and our models including C-FT and C-FDM, on AURORA-BENCH.

rithm 1.

F Prompt Template for Using GPT4o-as-a-Judge

We provide the prompts used for evaluating image editing performance with GPT-4o in Figure 10. We use `GPT-4o-2024-11-20`. The final score is the average of the minimum value of the two scores for each sample, as in (Fang et al., 2025).

G Detailed GPT4o Scores for Editing Success and Minimal Editing

Figure 11 shows the distribution of editing success (ES) and minimal editing (ME) scores for standard training and loss-weighted training. Loss weighting tends to improve editing success, with a modest trade-off in minimal editing quality in most of the datasets.

H Implementation Details

H.1 Chameleon Dynamics Model

We fine-tune the Chameleon-7B checkpoint from the Anole-7B version (Chern et al., 2024) to predict the action given a pair of observations, framed as an action-prediction task. The model is trained on a merged dataset from Action-Genome, Kubric, MagicBrush, Something-Something from AURORA's annotated trajectories, and 15K EPIC-Kitchens processed by us. We downsample Kubric's trajectories to 10K. Training is performed for 10 epochs with a batch size of 64, using a learning rate of 2e-4

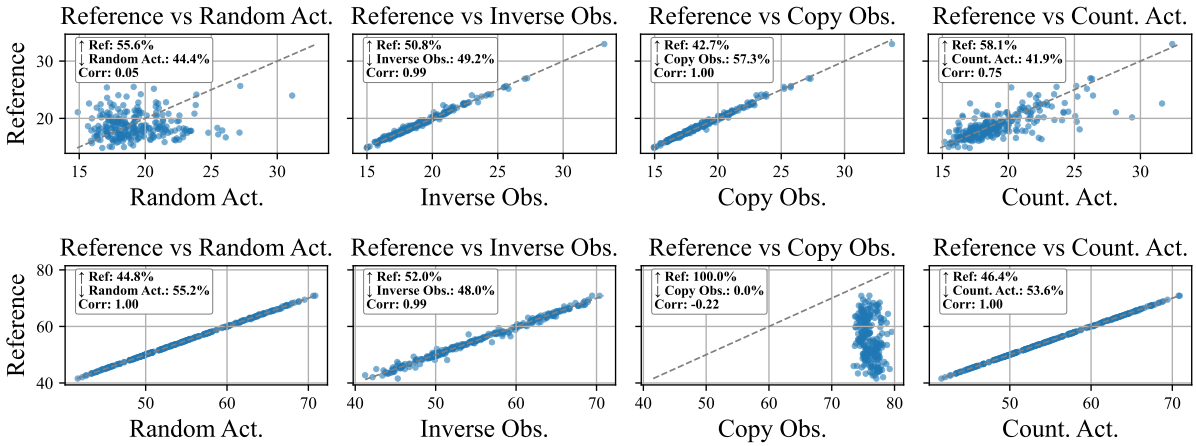


Figure 7: Comparison of predicted negative log-likelihoods (lower values indicate stronger model preference) for ground-truth real-world trajectories versus four types of negative trajectories. **Top**: Action prediction task for the IDP (observation × observation → action). **Bottom**: Next observation prediction task for the FDP (observation × action → observation). The legend shows the percentage of times the model prefers the ground-truth trajectory (↑) over the negatives (↓).

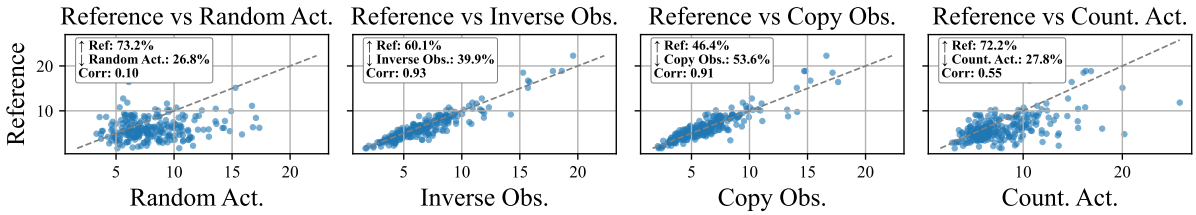


Figure 8: Comparison of negative log-likelihoods (lower values indicate stronger model preference) of the action predicted by IDM for ground-truth trajectories versus four types of negative trajectories.

and cosine scheduling (500 warm-up steps). We use bfloat16 mixed-precision training and apply LoRA (Hu et al., 2022) for parameter-efficient fine-tuning (rank 16, $\alpha = 32$, dropout 0.05). Only the completion loss is used to optimise the generation of action. Training is conducted on 4 NVIDIA-H100-80GB-HBM3 GPUs using DeepSpeed for distributed optimisation.

H.2 C-FT Baseline

We fine-tune the Chameleon-7B checkpoint from the Anole-7B version (Chern et al., 2024). The model is trained on a combined dataset from Action-Genome, Kubric, MagicBrush, and Something-Something, formatted as the image editing task. We downsample Kubric's trajectories to 10K. Training is conducted for 40 epochs with a batch size of 96 using the AdamW optimiser and a cosine learning rate scheduler (learning rate of 5e-4, 400 warm-up steps). We use mixed-precision training with bfloat16 and apply LoRA (Hu et al., 2022) for efficient fine-tuning (rank 16, $\alpha = 32$, dropout 0.05). We only train the model with the truncated loss from the completion part. We use 4 NVIDIA-H100-80GB-HBM3 GPUs with DeepSpeed for distributed training. During inference, we apply a logits processor to mask out non-image tokens, set the temperature to 1, and use top-1 sampling. We observe that temperature is critical in controlling model behaviour: lower values often cause the model to copy the source observation instead of generating meaningful edits.

H.3 Chameleon FDM

We fine-tune the Chameleon-7B checkpoint from the Anole-7B version (Chern et al., 2024). The model is trained on a combined dataset from Action-Genome, Kubric, MagicBrush, Something-Something from AURORA's annotated trajectories, together with 7K trajectories from Movements-in-Time, 7K trajectories from UCF-101 and 7K trajectories from Kinetics700, formatted as the image editing task. Again, we downsample Kubric's trajectories to 10K. Training is conducted for 40 epochs with a batch size of 96 using the AdamW optimiser and a cosine learning rate scheduler

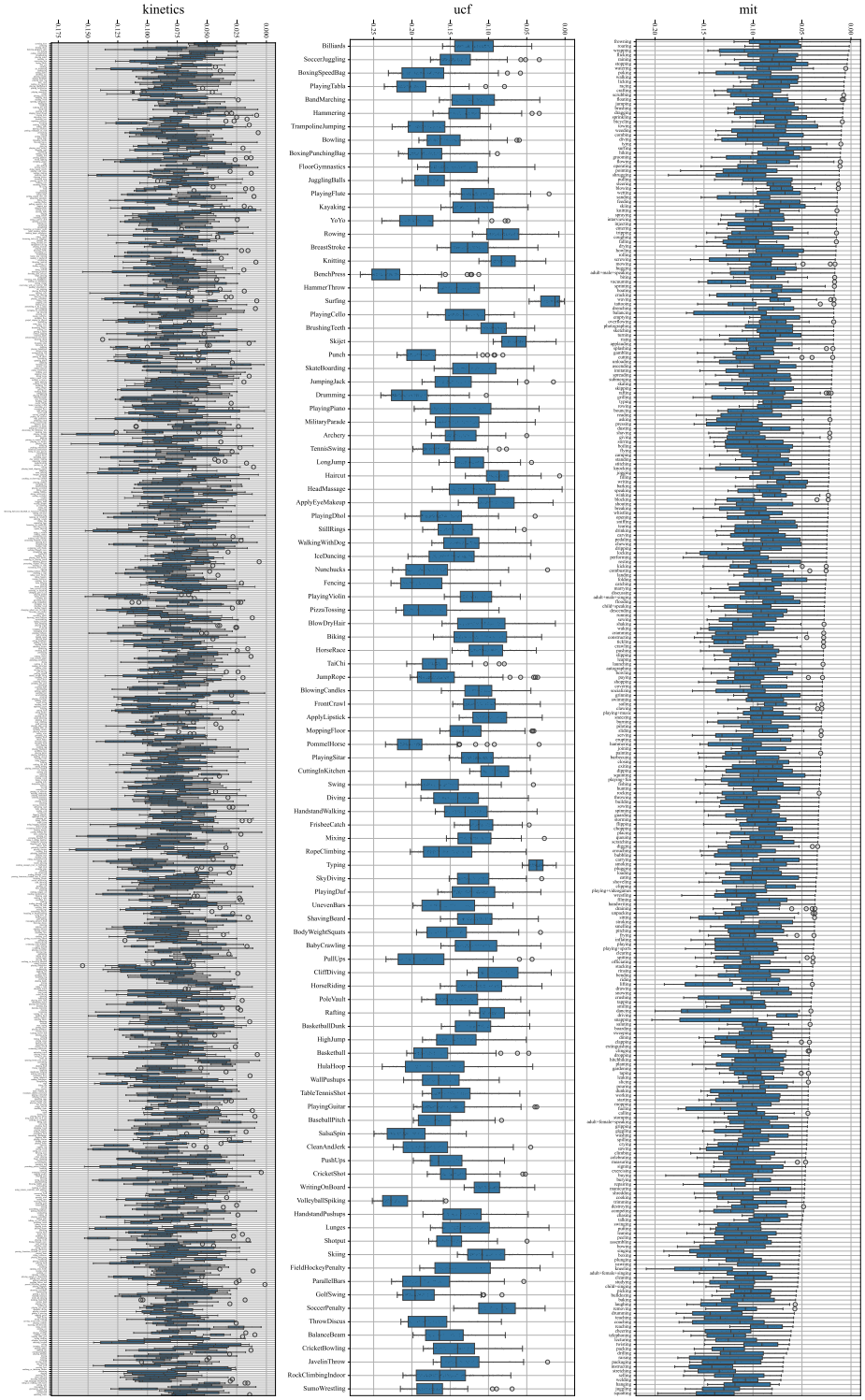


Figure 9: Distributions of triplet log-likelihoods predicted by IDM on Movements-in-Time, UCF-101, and Kinetics-700, based on 7K synthetic triplets per dataset. Triplets are uniformly sampled from each action class while maximising overall predicted likelihoods.

Algorithm 1 Stratified Top-K Sampling with Action Class Uniformity

Require: Trajectory triplet set $X = \{(o_s^i, o_t^i, a^i, s^i, c^i)\}_{i=1}^N$, where s_i is the predicted likelihood of a^i, $c_i \in \mathcal{C}$ is the class, number of samples K

Sort X descending by score s_i
Initialize $S \leftarrow \emptyset$, and class_counts$[c] \leftarrow 0$ for all $c \in \mathcal{C}$
while $|S| < K$ **do**
  for all class $c \in \mathcal{C}$ in round-robin order **do**
    $X_c \leftarrow$ top unsampled item from class c in X
    if $X_c \neq \emptyset$ **then**
      $S \leftarrow S \cup \{X_c\}$
      Remove X_c from X
      class_counts$[c] \leftarrow$ class_counts$[c] + 1$
    end if
    if $|S| = K$ **then**
      break
    end if
  end for
end while
return S

Dataset	Video		Triplet			
	Avg. Length	Total Length	#Samples	#Avg. OPV	#Avg. APV	#Avg. WPA
MIT	3.04 seconds	2.57 hours	19,658	2.05	1.05	7.10
UCF-101	7.24 seconds	26 hours	10,965	3.00	2.00	8.96
Kinetics700	9.02 seconds	18 hours	26,959	2.71	1.71	7.39

Table 7: Dataset statistics for the video and triplets from the trajectories annotated by IDM. **OPV**: observations (i.e., extracted key-frames) per video, **APV**: actions per video, **WPA**: words per action.

(learning rate of 5e-4, 400 warm-up steps). We use mixed-precision training with bfloat16 and apply LoRA (Hu et al., 2022) for efficient fine-tuning (rank 16, $\alpha = 32$, dropout 0.05). We only train the model with the truncated loss from the completion part, but we weight the image tokens using L_2 strategy as introduced in Section 3. We use 4 NVIDIA-H100-80GB-HBM3 GPUs with DeepSpeed for distributed training. We use the same hyperparameters as C-FT during the inference time.

H.4 Computing Resources

All training experiments were conducted on a compute node equipped with 4× NVIDIA H100 80GB GPUs, 256 CPU cores, and 256GB of memory. The total GPU hours required for training C-FT, C-FDM, and IDM were approximately 200, 400, and 100 hours, respectively.

For inference, we used a single NVIDIA A100 80GB GPU with 8 CPU cores and 128GB memory. Inference for C-FT and C-FDM takes approximately 1 GPU hour per model. When applying verification with $K = 8$, inference time increases to around 8 GPU hours. IDM only takes around 0.3 GPU hours for inference.

H.5 Assets and Licenses

In this section, we list the public assets we used in this paper and the corresponding links.

Datasets. We include the detailed license and URL for the datasets we used in this paper.

- AURORA and AURORA-BENCH (Krojer et al., 2024): MIT license, the reader can find the corresponding version we use in this paper in https://github.com/McGill-NLP/AURORA.
- Movements-in-Time (Monfort et al., 2019): BSD-2-Clause license and its own License for Non-Commercial Use, the reader can find the corresponding version we use in this paper in http://moments.csail.mit.edu/.

Prompt Template for GPT4o-as-a-judge Evaluation

You are a professional digital artist. You will have to evaluate the effectiveness of the AI-generated image(s) based on the given rules.

You will have to give your output in a valid way of a Python dictionary format (Keep your reasoning concise and short.):

```
{{"score": [...], "reasoning": "..." }}
```

and don't output anything else. Two images will be provided:

- The first being the original AI-generated image
- The second being an edited version of the first.

The objective is to evaluate how successfully the editing instruction has been executed in the second image.
Note that sometimes the two images might look identical due to a failure in image editing. From a scale of 0 to 10:

- A score from 0 to 10 will be given based on the success of the editing.
- A second score from 0 to 10 will rate the degree of minimal editing.

Editing instruction: {instruction}

Figure 10: Prompt template used for GPT-4o-as-a-judge evaluation.

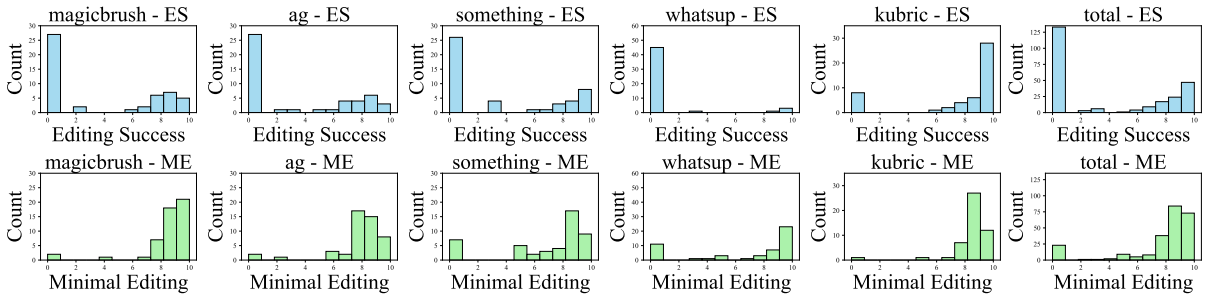
- UCF-101 (Soomro et al., 2012): unknown license, the reader can find the corresponding version we use in this paper in https://huggingface.co/datasets/flwrlabs/ucf101.
- Kinetics700 (Kay et al., 2017; Carreira et al., 2019): Creative Commons Attribution 4.0 International License, the reader can find the corresponding version we use in this paper in https://research.google/pubs/the-kinetics-human-action-video-dataset/.
- EPIC-Kitchens (Damen et al., 2018): Creative Commons Attribution-NonCommercial 4.0 International License, the reader can find the corresponding version we use in this paper in https://epic-kitchens.github.io/.

Implementation. We use the other following code for the implementations:

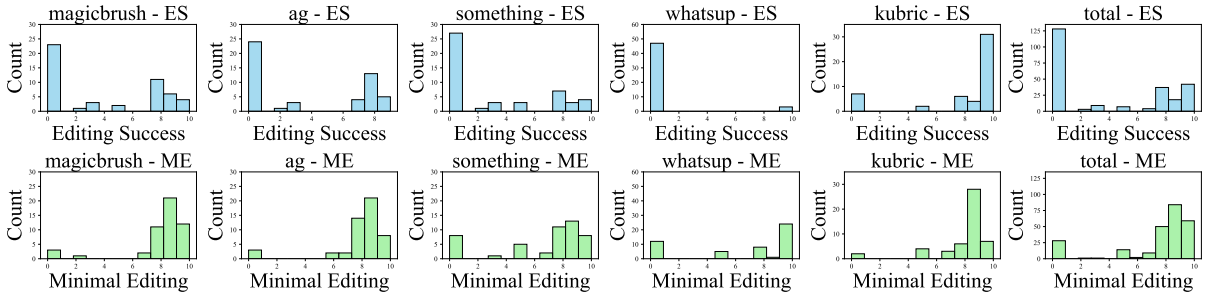
- Transformers (Wolf et al., 2020): Apache-2.0 license. We use the 4.47.0 version, following the link at https://github.com/huggingface/transformers.
- DeepSpeed: We use the 0.14.4 version, following the link at https://github.com/deepspeedai/DeepSpeed.

Model. We use the following models or checkpoints for the implementations:

- Chameleon (Chameleon Team, 2024): Chameleon Research License, the reader can find the corresponding version we use in this paper in https://github.com/facebookresearch/chameleon.
- Anole-7B (Chern et al., 2024): Chameleon Research License and MIT License, the reader can find the corresponding version we use in this paper in https://github.com/GAIR-NLP/anole.
- VILA-U (Chern et al., 2024): MIT License, the reader can find the corresponding version we use in this paper in https://github.com/mit-han-lab/vila-u.
- SmartEdit (Huang et al., 2024): Apache-2.0, the reader can find the correspond-



(a) Detailed GPT4o scores for C-FDM trained with the standard loss.



(b) Detailed GPT4o scores for C-FDM trained with the L_2-weighted loss.

Figure 11: GPT4o scores' distributions of editing success (ES) and minimal editing (OE) for C-FDM trained with standard loss or our loss-weighting method.

ing version we use in this paper in https://huggingface.co/TencentARC/SmartEdit-7B.

- GoT (Fang et al., 2025): MIT License, the reader can find the corresponding version we use in this paper in https://github.com/rongyaofang/GoT.

- PixInstruct (Brooks et al., 2023): PixInstruct customised license, the reader can find the corresponding version we use in this paper in https://github.com/timothybrooks/instruct-pix2pix.

I Details of Human Evaluation

We conducted a human evaluation using a custom-built interface, with the full interface and instructions shown in Figure 12. A total of 14 participants were recruited, all of whom are PhD-level graduate students or higher. Participation was voluntary. Each participant was asked to evaluate 25 samples, which typically required 15–20 minutes to complete.

The evaluation process, including recruitment, instructions, and data processing and storage, followed our institution's ethical guidelines for human subject research. All participants were informed of the purpose of the study and provided consent. No personally identifiable information was collected, and all data were stored and analysed in accordance with privacy standards.

J Safeguards

C-FDM performs observation prediction through image generation and, while its outputs are task-specific, we acknowledge that any generative model may carry potential for misuse. To mitigate these risks, we commit to the following safeguards upon release:

The model will be released solely for research purposes under a license that prohibits commercial use or any other harmful applications. The GitHub repository will include clear usage guidelines and terms of use, aligned with responsible AI principles.

We will include a disclaimer that the model is intended only for academic research in controlled environments. The datasets used for training are publicly available, action-centric image editing benchmarks that do not include sensitive or personally identifiable content.

Given the targeted nature of our model and the safeguards in place, we believe the risk of misuse is limited. Nonetheless, we encourage responsible use and welcome feedback from the community regarding potential improvements to safety.

K LLMs Usage Declaration

We declare that the large language model (LLM) was only used to assist in minor tasks, including revising the manuscript for grammatical correctness, improving phrasing, and performing small technical implementations such as debugging code snippets. All scientific ideas, results, analyses, and conclusions presented in this paper are entirely the work of the authors.

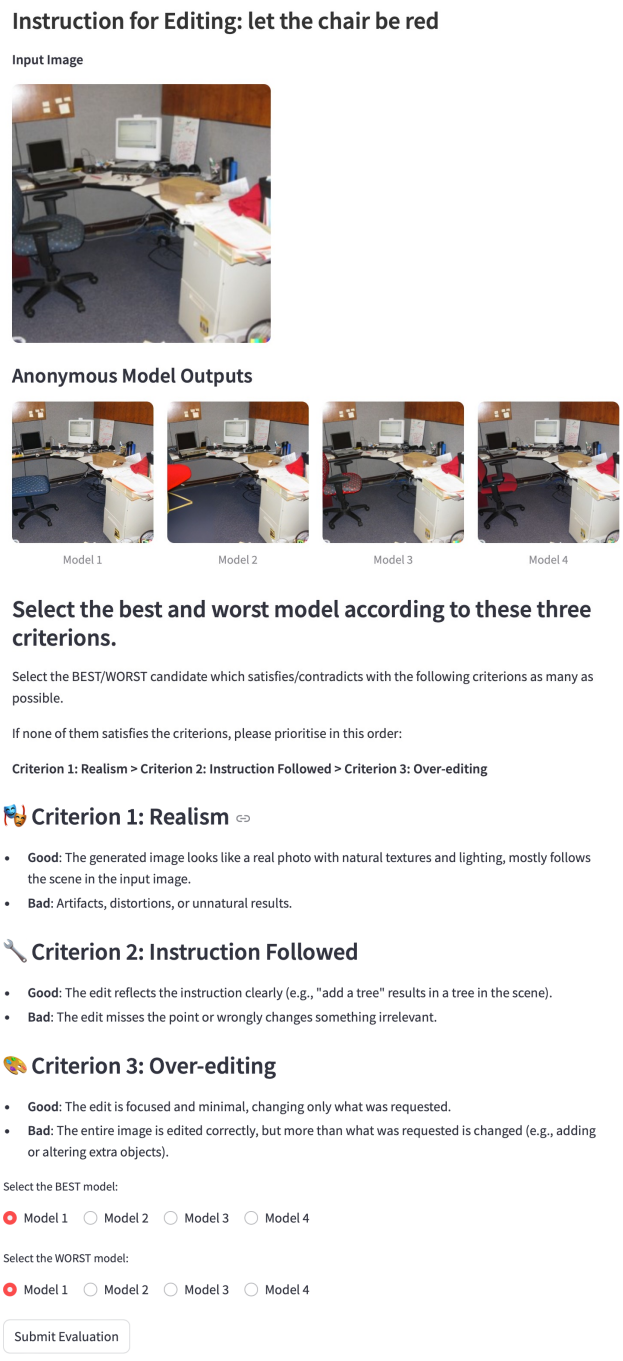


Figure 12: The screenshot for the instructions given to participants and the interface developed for conducting the evaluation.