LEGO: Learning EGOcentric Action Frame Generation via Visual Instruction Tuning

Supplementary Materials

Bolin Lai[1,2,†], Xiaoliang Dai[1], Lawrence Chen[1], Guan Pang[1], James M. Rehg[3], and Miao Liu[1,*]

[1] GenAI, Meta
[2] Georgia Institute of Technology
[3] University of Illinois Urbana-Champaign
bolin.lai@gatech.edu {xiaoliangdai,lawrencechen,gpang,miaoliu}@meta.com jrehg@illinois.edu

This is the supplementary material for the paper "LEGO: Learning EGOcentric Action Frame Generation via Visual Instruction Tuning". We organize the content as follows:

[†]This work was done during Bolin's internship at GenAI, Meta.
[*]Corresponding author.

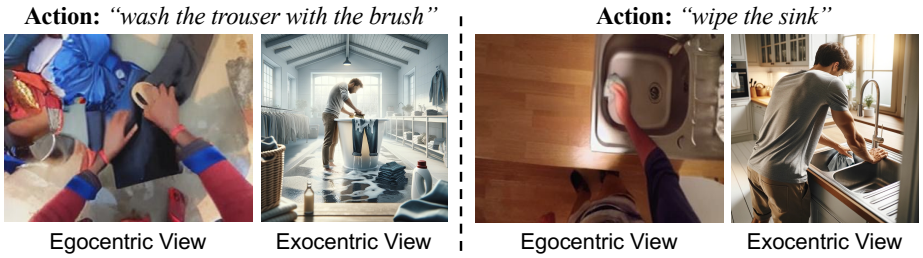

Fig. 1: Comparison of action frames generated in the egocentric and exocentric views. The egocentric images are synthesized by our model, and the exocentric images are synthesized by an off-the-shelf text-to-image generation model.

A Comparison of Egocentric and Exocentric Views

Our proposed problem and model mainly focus on action frame generation in the egocentric view rather than exocentric view (*i.e.*, third-person view). The examples for the comparison of the two views are illustrated in Fig. 1. The instructional images generated in the exocentric view may be far from the camera. The important details of hand-object interactions are unclear and occluded, thus making it hard for users to follow. In contrast, the action frames generated in the egocentric view by our model exactly match the user's viewpoint and clearly capture important details for action execution. In addition, it's more feasible for users to take a picture that captures the current contexts (*i.e.*, the input frame to our model) from the egocentric perspective than exocentric perspective, especially when the user is working alone. The egocentric image can be obtained even more easily by using a wearable device. These advantages further motivate us to focus on the egocentric view for action frame generation.

B Domain Gap Between Existing Diffusion Models and Our Problem

In the egocentric action frame generation problem, one challenge is the domain gap of the existing diffusion models and our problem (elaborated in Sec. 1 of the main paper). In this section, we show some specific examples to provide more insights. In Fig. 2, we present a few image-text pairs for text-to-image diffusion model training (left) and data samples for egocentric action frame generation (right). The diffusion model training data is mostly captured from the exocentric perspective and the prompts are about the objects rather than human daily activities. We also evaluate the performance of InstructPix2Pix [2] using *off-the-shelf* weights and compare it with the finetuned counterpart in Tab. 1. The prominent performance drop in all metrics further confirms the impact of the domain gap. Though we can finetune the model with egocentric data, such a big domain gap still limits the performance of existing diffusion models applied to our problem. To bridge this gap, our proposed LEGO model incorporates

Fig. 2: Demonstration of the domain gap of off-the-shelf diffusion models and our problem. The existing diffusion models are pre-trained with exocentric images with prompts mainly about objects, while our problem requires the capability of action generation in the egocentric view.

Table 1: Performance of InstructPix2Pix model on our problem without finetuning. As a comparison, we also show the results of finetuned model in parentheses.

	EgoVLP	EgoVLP$^+$	CLIP	FID ↓	PSNR	LPIPS ↓
Ego4D	38.2 (62.2)	67.7 (78.8)	68.5 (78.8)	46.0 (24.7)	11.6 (12.2)	41.4 (37.2)
EK-100	34.8 (43.0)	46.9 (61.1)	68.1 (77.0)	51.0 (20.6)	10.9 (11.2)	43.1 (40.8)

the embeddings from a visual instruction tuned LLM as additional conditioning, which enables the diffusion model to learn action state transition more effectively from an egocentric perspective.

C Comparison with Prior Image Generation Models Using LLMs

We compare our model with prior image generation and image editing models that utilize LLMs to improve the performance (see Tab. 2). Our model differs from prior methods mainly in four aspects: (1) All prior models are designed for exocentric image generation, while our model is proposed for image generation in the egocentric view, which is still understudied. (2) Prior models are trained with data that doesn't have the domain gap with diffusion model pre-training data. Therefore, prior models can directly use the *off-the-shelf* LLM parameters (*i.e.*, without finetuning) for their tasks. In contrast, our model finetunes the LLM by visual instruction tuning to narrow the domain gap. (3) We also innovatively incorporate LLM embeddings into the diffusion model to boost image generation performance, which has not been investigated in prior work. (4) Prior image editing methods focus on local object manipulation and global style transfer. In our work, we focus on generating images of actions conducted in the same contexts as input, which has not been studied in prior image editing methods. These differences consolidate our contributions and thus notably distinguish our model from prior work.

In addition to general image generation/editing models, there is some work about hand-object interaction (HOI) generation, which is also relevant with our

Table 2: Comparison with prior text-to-image generation (T2I) and image editing methods that incorporate large language models. Please see Sec. C for more discussions.

Methods	Domain Gap	Source of LLM Parameters	How to use LLM	LLM Embed.	Edited Content	Main Task
Chen *et al.* [4]	No	off-the-shelf	Auto-label	w/o	N/A	T2I
Liu *et al.* [12]	No	off-the-shelf	Auto-label	w/o	N/A	T2I
Lian *et al.* [11]	No	off-the-shelf	Enrich Prompt	w/o	N/A	T2I
Yu *et al.* [22]	No	off-the-shelf	Enrich Prompt	w/o	N/A	T2I
Wu *et al.* [20]	No	off-the-shelf	Controller	w/o	N/A	T2I
Wen *et al.* [18]	No	off-the-shelf	Multi-round Refine	w/o	N/A	T2I
Wu *et al.* [19]	No	off-the-shelf	Multi-round Refine	w/o	N/A	T2I
Chakrabarty *et al.* [3]	No	off-the-shelf	Controller	w/o	Obj.&Style	Edit
Koh *et al.* [10]	No	off-the-shelf	Controller	w/o	Obj.&Style	Edit
Wang *et al.* [17]	No	off-the-shelf	Controller	w/o	Object	Edit
Chen *et al.* [5]	No	off-the-shelf	Multi-round Refine	w/o	Object	Edit
LEGO (Ours)	**Yes**	**Instruction-tuned**	**Enrich Prompt**	**w/**	**Action**	**Edit**

problem. To show the difference, we compare with two important models in this area – AffordanceDiffusion [21] and HOIDiffusion [23]. The two models take a 2D object image or 3D object model as inputs. This results in a much easier synthesis task, as these models simply have to add a hand in a correct pose to augment the input. In contrast, our real-world egocentric input image already contains the hands and complex scene context. The diffusion model has to identify the initial state of the action and synthesize the correct action state afterwards with the hands in the correct location and an adjusted camera viewpoint, while preserving important scene context. This is a much harder task with unique challenges resulting in a more substantial domain gap.

D Analysis of Image-to-Text Metrics

In Tab. 3, we report the image-to-text CLIP score of InstructPix2Pix (IP2P) [2] baseline and the ground truth. Ideally, the CLIP score between the ground truth and the text prompt should serve as a performance uppperbound (UB). However, the image-to-text CLIP score of upperbound is very close to the baseline on Ego4D, and even lower than the baseline on Epic-Kitchens. It suggests the CLIP model fails to align action descriptions with corresponding egocentric images in semantics, thus resulting in a quick saturation in CLIP score. In our experiments, we use BLIP to caption the generated image and measure the text-to-text similarity of captions and action descriptions (following [9]). The two metrics (BLIP-B and BLIP-L) that use two different BLIP structures both result in larger gap between the baseline model and the upperbound (3.68%/2.96% vs. 0.85% and 1.61%/1.44% vs.−1.08%). Therefore, we adopt BLIP based metrics and user study to measure image-text alignment. Note that, our model still performs on-par or slightly better than IP2P in image-to-text CLIP score, and exceeds IP2P notably when using BLIP based metrics and user study (see Tab. 2 and Fig. 4 in the main paper).

Table 3: Image-to-text metrics of the baseline model and uppperbound. The gray row shows the gap of IP2P to the upperbound. The upperbound measured by CLIP score is comparable with or even lower than the baseline model (highlighted by red).

Methods	Ego4D			Epic-Kitchens		
	CLIP	BLIP-B	BLIP-L	CLIP	BLIP-B	BLIP-L
IP2P [2]	20.53	20.00	20.56	21.68	25.37	26.36
UpperBound (UB)	21.38	23.68	23.52	20.60	26.98	27.80
Δ = UB−IP2P	0.85	3.68	2.96	-1.08	1.61	1.44

Table 4: Comparison of LEGO trained with single dataset and both datasets (denoted as *scaleup*). ↓ means a lower score in this metric suggests a better performance. The better results are highlighted with **boldface**. The performance of LEGO model can be effectively improved by involving more training data.

	Methods	EgoVLP	EgoVLP$^+$	CLIP	FID ↓	PSNR	LPIPS ↓
Ego4D	LEGO	65.65	80.44	80.61	23.83	12.29	36.43
	LEGO (scaleup)	**66.32**	**80.77**	**80.90**	**23.56**	**12.30**	**36.33**
EK-100	LEGO	45.89	62.66	78.63	21.57	11.33	40.36
	LEGO (scaleup)	**47.46**	**63.51**	**78.90**	**19.64**	**11.40**	**39.88**

There's also another evaluation strategy used by the concurrent work – GenHowTo [6]. They train a 10-class classifier to distinguish real initial state images and generated final state images of 5 action categories, and use the testing results on real final state images as the metric. Our EgoVLP$^+$ metric shares the same motivation of evaluating action state transition as this metric. However, EgoVLP$^+$ is not limited to a fixed set of action categories and does not require training a classifier for each baseline model. Thus we prefer to use EgoVLP$^+$ in our experiments.

E Additional Experiment Results

E.1 Performance at Different Transition Time

As explained in Sec. 4.1 of main paper, for an action beginning at t, we select the frame at $t - \delta_i$ as input and the frame at $t + \delta_o$ as target. We divide the test data into four bins according to the action state transition time from input frame to target frame $\delta = \delta_i + \delta_o$. We establish the threshold for each bin to ensure that the quantity of data samples in each bin is relatively similar. The performance of LEGO and baselines at different transition time is demonstrated in Fig. 3. The flat curves suggest the egocentric action frame generation problem is equally challenging regardless of transition time. Our model still surpasses all

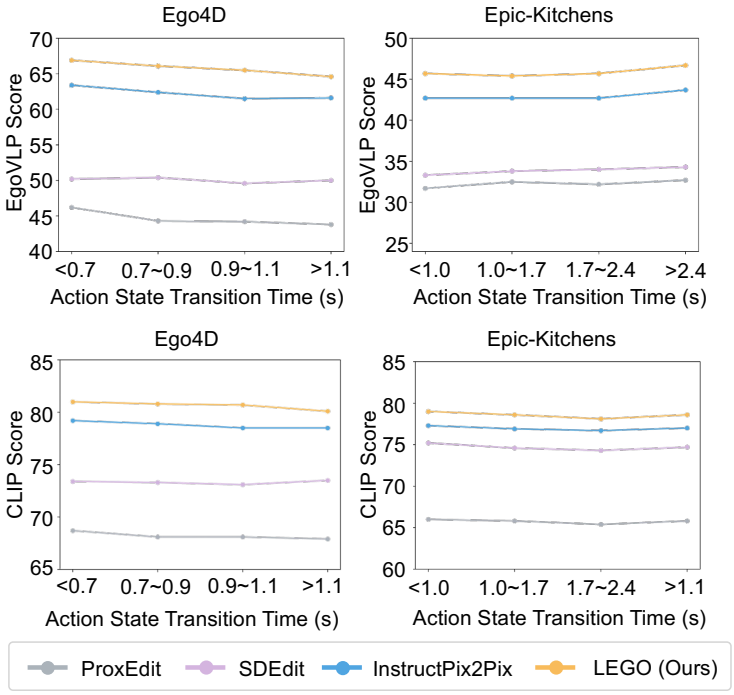

Fig. 3: Comparison with baselines at different action transition time. Our model outperforms all baselines across all transition time.

baselines by a notable margin across all transition time. This fine-grained result further validates the superiority of our approach.

E.2 Effect of Dataset Scaleup

We further analyze how the dataset scale affects our model performance by merging the training set of Ego4D and Epic-Kitchens. The training strategy and other hyper-parameters remain identical to separate training on each dataset (described in Sec. F.3). We demonstrate the results in Tab. 4. The performance of our model is further boosted by leveraging more training data (*i.e.*, scaleup). Notably, the gains on Epic-Kitchens are more prominent than gains on Ego4D (*e.g.*, 1.57% vs. 0.67% on EgoVLP score, 1.93 vs. 0.27 on FID, *etc.*). The possible reason is that Ego4D dataset has more training data covering more diverse scenarios and actions. Hence, it can greatly compensate for the low diversity of Epic-Kitchens dataset after merging them. The improvement on two datasets suggests our model can be effectively boosted by scaling up the training data.

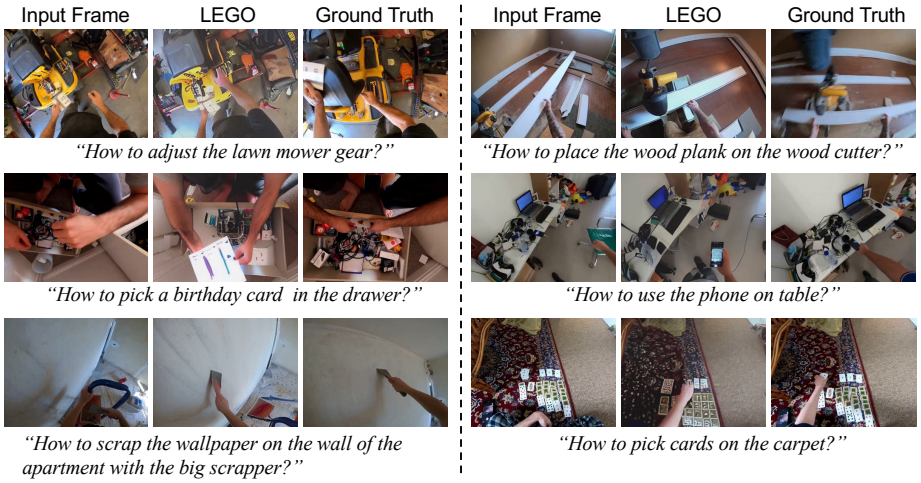

Fig. 4: Additional visualization of LEGO output on Ego4D as well as the ground truth. Ground truth shows how each action is actually conducted in the real world.

E.3 Additional Visualization

We demonstrate more results of our model and the corresponding ground truth on Ego4D (see Fig. 4) and Epic-Kitchens (see Fig. 5). The generated frames are well aligned with the user query and the ground truth in these examples. To better understand the limitation of our model, we also illustrate some failure cases in Fig. 6. Our approach may fail to associate the action with the correct objects when the objects are not distinct enough in the egocentric perspective, *e.g.*, the *marker* and *croissant* in the first row of failure cases. In addition, generating the action frame in some scenarios needs more contexts than a static input frame. For example, the model fails to understand which object is the furniture and incorrectly drives the nails into the wood under it (*i.e.*, the second failure case of Ego4D). It also lacks the context that the user already holds a bag of noodles, so it synthesizes a frame of taking out the noodles from a cupboard (*i.e.*, the second failure case of Epic-Kitchens). These weaknesses can inspire more future studies in action understanding and egocentric action frame generation. Please refer to Sec. G for more discussions.

F More Implementation Details

F.1 Prompt and Examples for Data Curation

In data curation, we randomly select 12 examples from each datasets covering diverse scenarios. All examples are shown in Fig. 7 and Fig. 8. We also clarify our requirements in the prompt sent to GPT-3.5. The complete prompt is shown in Fig. 9. We specify the composition of input query, the expected detailed

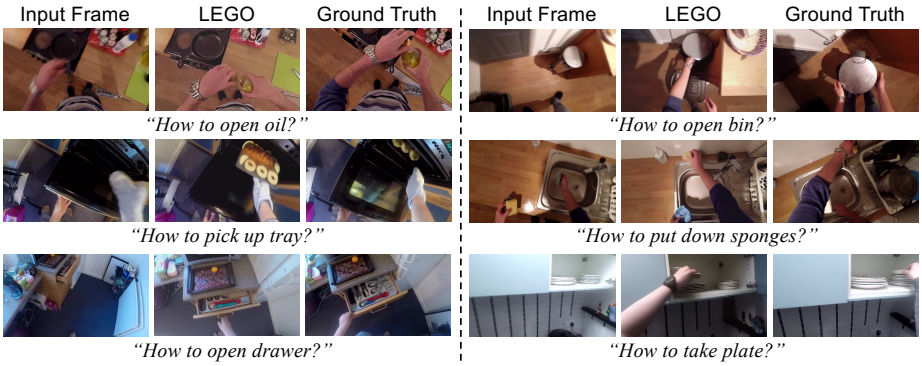

Fig. 5: Additional visualization of LEGO output on Epic-Kitchens as well as the ground truth. Ground truth shows how each action is actually conducted in the real world.

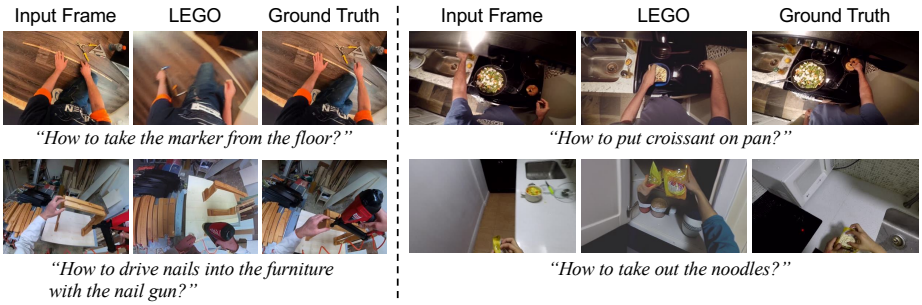

Fig. 6: Failure cases of our model in Ego4D (on the left of the dash line) and Epic-Kitchens (on the right of the dash line). Please refer to Sec. E.3 for more analysis.

information and extra requirements for the output in the system information. Then we fill the examples for in-context learning and the example for inference in the prompt and input it to GPT-3.5 for data curation.

F.2 Details of Data Preparation and Improvement

Data Preparation for Visual Instruction Tuning. For our dataset curation, we randomly select 20,891 actions with bounding box annotations from the Ego4D training set and 17,922 actions with VISOR [7] mask annotations from the Epic-Kitchens training set. We leverage GPT to produce detailed descriptions of these actions as described in Sec. 3.1 of main paper. For instruction tuning, we insert the original action label into a prompt template to construct the full user prompt. In order to diversify the prompt structure, we prepare 10 prompt templates and randomly select one for each action at training time.

Data Improvement for Action Frame Generation. For an action starting at t, we select an egocentric image frame δ_i seconds before the action begins as the input, and an image δ_o seconds after the action begins as the target frame.

Fig. 7: All Ego4D examples used for data curation with GPT-3.5 via in-context learning. For simplicity, the bounding boxes are only shown on images. We input the coordinates of bounding boxes to GPT-3.5 in practice.

Due to the possible drastic camera motion, the egocentric image frames at $t-\delta_i$ and $t+\delta_o$ may be blurry. As a mitigation, we first calculate aesthetic scores [1] of the frames at $t-\delta_i$ and $t+\delta_o$ as well as 3 frames before and after them. The corresponding frames with the highest aesthetic score are used as the input and ground truth of our model. In addition, the egocentric body motion may have huge variance depending on the action type, meaning that the input frame and target frame may look almost identical in more stationary actions (*e.g.*, camera wearer is reading book), or significantly different in more active actions (*e.g.*, outdoor activities with frequent head motion). Such large variances may incur additional barriers for the diffusion model training. Therefore, we calculate the similarity of the input frame and target frame, and we filter out the instances where the similarity is lower than 0.81 or higher than 0.97. With these steps, we ultimately curate 85521/9931 data samples for the train/test sets from Ego4D and 61841/8893 data samples for the train/test sets from Epic-Kitchens.

Preprocessing Examples. For each input frame or target frame, we calculate aesthetic scores [1] of the current frame as well as 3 frames before and after (7 frames in total). As demonstrated in Fig. 10(a), the frame of the highest

Fig. 8: All Epic-Kitchens examples used for data curation with GPT-3.5 via in-context learning. For simplicity, the bounding boxes are only shown on images. We input the coordinates of bounding boxes to GPT-3.5 in practice.

aesthetic score usually has the best image quality. We then use this frame as input or target frame. We also calculate the similarity of input and target frame for each action. We empirically filter out data whose similarity is lower than 0.81 or higher than 0.97. Some examples of abandoned data are shown in Fig. 10(b). A very low similarity usually indicates a big change in the background due to the head motion. A very high similarity implies the action involves very small hand movements. Such a big variance in these data samples increases the challenge for generative models to learn action state transition.

F.3 Training Details for Visual Instruction Tuning and Action Frame Generation

Visual Instruction Tuning. We train the model with a batch size of 128 and a learning rate of 2×10^{-5}. Warm-up strategy and consine anneal [14] are also used in training. It takes 24 hours to train the model on 8 NVIDIA A100-SXM4-40GB for 3 epochs. AdamW [13] is adopted as the optimizer for training.

Egocentric Action Frame Generation. In training, we feed the input frame, ground truth frame (to obtain the gaussian noise through the diffusion process) together with the enriched action descriptions and VLLM embeddings into the

System: *You are an AI assistant that provides a description of an image based on the action and object context. The action consists of a verb and nouns. Each object location is represented by a bounding box. For each bounding box, four numbers are provided in brackets – they are [x-coordinate of top-left, y-coordinate of top-left, x-coordinate of bottom-right, y-coordinate of bottom-right]. The origin is at the top-left of each frame. The x-axis is on the top and the y-axis is on the left. All coordinates are normalized to the range from 0 to 1. This information can be used to infer the spatial relation of hands and objects. Note that the detailed narration is in a natural and holistic style. Please add more details in the action. For example, try to describe which hand is used in each action like "with right hand" or "using left hand". Try to describe the spatial relation of these objects like "on the right", "on the left", "from ... to ..." or use some spatial words like "in", "on", "out", "front", "back", etc. Describe the image in only one sentence. Do not describe objects or actions that are not presented in action or objects locations context. Many examples are provided for learning and an additional example is provided for inference.*

User: *Examples for learning: (1) {Example-1} (2) {Example-2} ... (12) {Example-12}*

User: *Example for inference: {Inference Example}*

Fig. 9: The structure of the prompt sent to GPT-3.5. We specify the composition of the input query (highlighted in blue). Then we articulate the requirements for action enrichment (highlighted in green) and extra demands (highlighted in yellow). Example-1 to Example-12 consist of the action label, object bounding boxes and manual annotation of detailed action descriptions. The inference example consists of just action label and object bounding boxes.

model. We finetune the latent diffusion model with a batch size of 256 and an initial learning rate of 10^{-4} without warm-up strategy. Horizontal flipping is used as data augmentation. We train the model with optimizer AdamW [13] for 20,000 iterations on 8 NVIDIA A100-SXM4-40GB over 38 hours. In inference, we feed the input frame, a randomly-sampled gaussian noise as well as enriched action descriptions and VLLM embeddings into the latent diffusion model. We apply 100 denoising steps for each instance.

F.4 Details about Classifier-free Guidance

We use classifier-free guidance for two conditions (following [2]) by sharing the same guidance scale across the enriched action descriptions, VLLM image embedding and VLLM text embedding. As defined in Sec. 3.2 in main paper, we use $\mathcal{C}$ to denote the three conditions and use $\mathcal{X}$ to denote the input frame. Specifically, we randomly set only the image conditioning $\mathcal{X} = \varnothing$ at a probability of 5%, only the conditioning from VLLM $\mathcal{C} = \varnothing$ at a probability of 5% and both $\mathcal{X} = \varnothing$ and $\mathcal{C} = \varnothing$ at a probability of 5%. Then the score estimate of our model

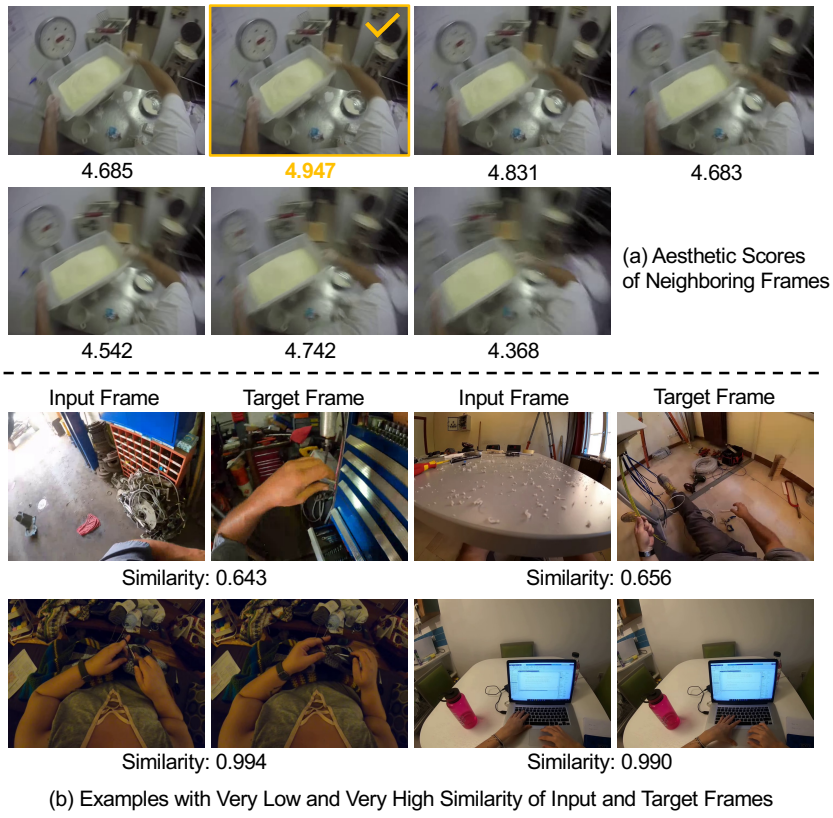

Fig. 10: Data preprocessing in our work. (a) The fame with the highest aesthetic score (highlighted) is less blurry and then used as the target frame of this action. (b) The actions with too low (<0.81) or too high similarity (>0.97) between input and target frames are filtered out from the datasets.

is formulated as

$$\tilde{e}_\theta(z_t, \mathcal{X}, \mathcal{C}) = e_\theta(z_t, \varnothing, \varnothing) \tag{1}$$
$$+ s_x \cdot (e_\theta(z_t, \mathcal{X}, \varnothing) - e_\theta(z_t, \varnothing, \varnothing)) \tag{2}$$
$$+ s_c \cdot (e_\theta(z_t, \mathcal{X}, \mathcal{C}) - e_\theta(z_t, \mathcal{X}, \varnothing)), \tag{3}$$

where θ refers to the parameters in the denoising UNet. z_t is the noisy latent at timestep t, which is obtained by diffusion process in training and randomly initialized by a gaussian noise in inference. s_x and s_c are the guidance scales corresponding to the conditioning $\mathcal{X}$ and $\mathcal{C}$ respectively. In inference, we use $s_x = 7.5$ and $s_c = 1.5$ which are identical to the settings in InstructPix2Pix [2].

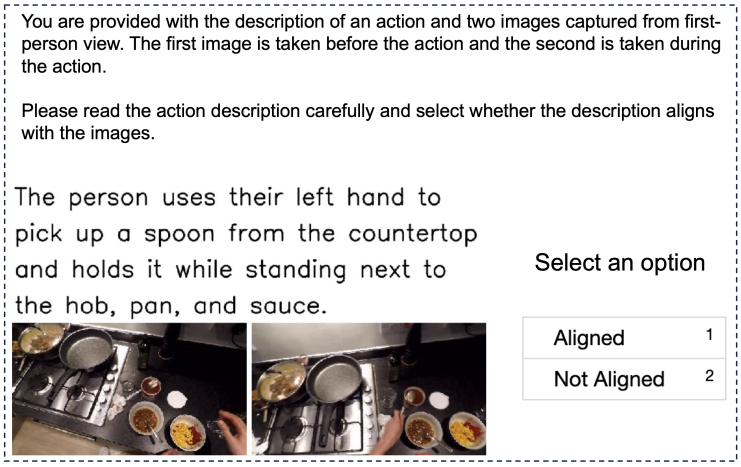

Fig. 11: The interface used for evaluation of enriched action descriptions. Both input and target frames are shown to the raters.

F.5 Implementation Details of the Baseline Models

We compare our model with three prior image editing models – ProxEdit [8], SDEdit [15] and InstructPix2Pix [2]. We provide more implementation details for training the three models on our dataset. ProxEdit and SDEdit rely on a pre-trained diffusion model to edit the input image. Given the domain gap of the off-the-shelf diffusion models and our problem, we finetune a latent diffusion model with the egocentric action data using the default training hyper-parameters in [16]. During finetuning, we use the action label as textual prompt and the target action frame as the ground truth. Then we implement ProxEdit and SDEdit on the finetuned diffusion model. In terms of InstructPix2Pix, we finetune it end to end with the egocentric action datasets using the same training hyper-parameters as our LEGO model (see Sec. F.3). Note that for all three baseline models, we use the short action labels as input rather than the detailed descriptions. The proposed action description enrichment is one of our key contributions, so we use it only for our model to show the benefit.

F.6 Details and Interfaces for User Study

User Study for the Enriched Action Descriptions. In Sec. 4.5 of the main paper, we apply the user study to evaluate the quality of enriched action descriptions from our instruction tuned VLLM and the off-the-shelf VLLM. We randomly sample 100 examples from the test set of each dataset. For each instance, we show the input frame, target frame and the action descriptions generated by VLLM. The rater is asked to select whether the description aligns with the two frames. We hire 5 raters for each instance on Amazon Mechanical Turk

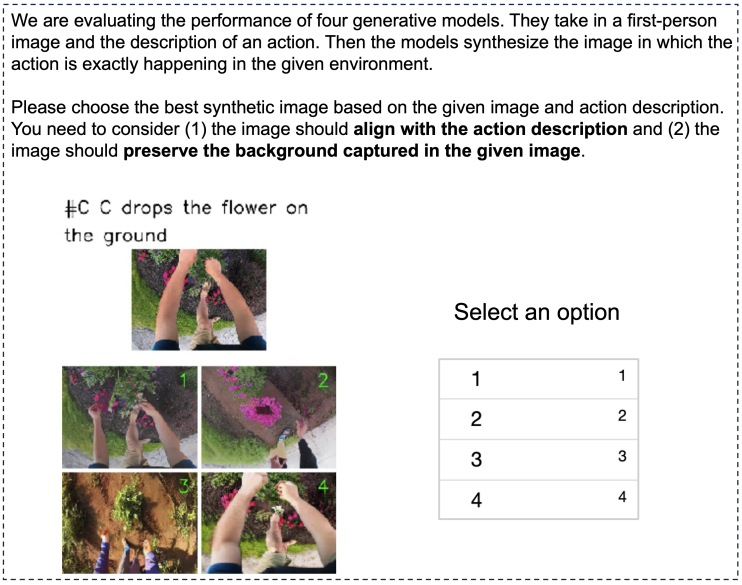

Fig. 12: The interface used for evaluation of generated action frames. The four generated frames are randomly shuffled to avoid potential bias.

and finally get 500 samples for each dataset. The interface shown to raters is illustrated in Fig. 11.

User Study for Generated Action Frames. The user interface for evaluation of generated action frames is illustrated in Fig. 12. We show the input frame and shuffled outputs from the four models to raters. To make a fair comparison, we show the action label instead of the enriched action description because the baseline models only take original action labels as input.

G Limitation and Future Work

In this paper, we use an egocentric image to capture the user's environment contexts and generate the action frame to provide visual instructions. However, there are still some problems that are not explicitly solved by our method. We find it's hard for our model to associate the action with correct objects when there are too many irrelevant objects around. Synthesizing the diverse and complex hand-object interactions is also a big challenge especially when people are operating some machines. In addition, our work indicates a few valuable directions for the future study.

- The embeddings from the visual large language model (VLLM) are fed into the UNet together with the CLIP based text representation as additional conditioning. How to leverage the VLLM embeddings more effectively in diffusion models deserves future study.

- Recognizing and localizing the objects that are relevant with the action descriptions in a chaotic environment may be a bottleneck for the application in real-world problems, which deserves more attention.
- It's difficult to synthesize correct interactions of hands and objects in some professional work, such as using a wood cutter, operating a lawn mower and sewing clothes on a sewing machine. How to combine affordance understanding with generative models may be a key step to address this problem.
- Existing automatic image-to-text similarity metric doesn't generalize well to the egocentric domain. We expect more investigation of better evaluation metrics for image-text alignment.

H Code and Data Release

We will release our code, pre-trained model weights, additional data annotations, train/test split, the enriched action descriptions and VLLM embeddings on the website (`https://bolinlai.github.io/Lego_EgoActGen/`) to the research community to facilitate future studies.

References

1. Aesthetic score predictor. `https://github.com/christophschuhmann/improved-aesthetic-predictor` (2023)
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023)
3. Chakrabarty, T., Singh, K., Saakyan, A., Muresan, S.: Learning to follow object-centric image editing instructions faithfully. arXiv preprint arXiv:2310.19145 (2023)
4. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., et al.: Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. arXiv preprint arXiv:2310.00426 (2023)
5. Chen, W.G., Spiridonova, I., Yang, J., Gao, J., Li, C.: Llava-interactive: An all-in-one demo for image chat, segmentation, generation and editing. arXiv preprint arXiv:2311.00571 (2023)
6. Damen, D., Wray, M., Laptev, I., Sivic, J., et al.: Genhowto: Learning to generate actions and state transformations from instructional videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6561–6571 (2024)
7. Darkhalil, A., Shan, D., Zhu, B., Ma, J., Kar, A., Higgins, R., Fidler, S., Fouhey, D., Damen, D.: Epic-kitchens visor benchmark: Video segmentations and object relations. In: Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks (2022)
8. Han, L., Wen, S., Chen, Q., Zhang, Z., Song, K., Ren, M., Gao, R., Stathopoulos, A., He, X., Chen, Y., et al.: Proxedit: Improving tuning-free real image editing with proximal guidance. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4291–4301 (2024)

9. Joseph, K., Udhayanan, P., Shukla, T., Agarwal, A., Karanam, S., Goswami, K., Srinivasan, B.V.: Iterative multi-granular image editing using diffusion models. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2024)
10. Koh, J.Y., Fried, D., Salakhutdinov, R.R.: Generating images with multimodal language models. Advances in Neural Information Processing Systems **36** (2024)
11. Lian, L., Li, B., Yala, A., Darrell, T.: Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. arXiv preprint arXiv:2305.13655 (2023)
12. Liu, B., Zhang, H., Liu, J., Wang, Q.: Acigs: An automated large-scale crops image generation system based on large visual language multi-modal models. In: 2023 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). pp. 7–13. IEEE (2023)
13. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2017)
14. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (2017)
15. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (2022)
16. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
17. Wang, Q., Zhang, B., Birsak, M., Wonka, P.: Instructedit: Improving automatic masks for diffusion-based image editing with user instructions. arXiv preprint arXiv:2305.18047 (2023)
18. Wen, S., Fang, G., Zhang, R., Gao, P., Dong, H., Metaxas, D.: Improving compositional text-to-image generation with large vision-language models. arXiv preprint arXiv:2310.06311 (2023)
19. Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., Duan, N.: Visual chatgpt: Talking, drawing and editing with visual foundation models. arXiv preprint arXiv:2303.04671 (2023)
20. Wu, S., Fei, H., Qu, L., Ji, W., Chua, T.S.: Next-gpt: Any-to-any multimodal llm. arXiv preprint arXiv:2309.05519 (2023)
21. Ye, Y., Li, X., Gupta, A., De Mello, S., Birchfield, S., Song, J., Tulsiani, S., Liu, S.: Affordance diffusion: Synthesizing hand-object interactions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22479–22489 (2023)
22. Yu, Q., Li, J., Ye, W., Tang, S., Zhuang, Y.: Interactive data synthesis for systematic vision adaptation via llms-aigcs collaboration. arXiv preprint arXiv:2305.12799 (2023)
23. Zhang, M., Fu, Y., Ding, Z., Liu, S., Tu, Z., Wang, X.: Hoidiffusion: Generating realistic 3d hand-object interaction data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8521–8531 (2024)