Supplementary Material for How Do I Do That? Synthesizing 3D Hand Motion and Contacts for Everyday Interactions

Aditya Prakash[*1] Benjamin Lundell[2] Dmitry Andreychuk[2] David Forsyth[1]
Saurabh Gupta[†1] Harpreet Sawhney[†2]
[1]University of Illinois Urbana-Champaign [2]Microsoft
https://bit.ly/LatentAct

We first provide additional dataset details used in the experiments. We then present additional analysis of our results. Next, we show qualitative comparisons of our LatentAct model with the best baseline. Finally, the video contains an overview of our key ideas and results.

1. Dataset Details

Statistics: We process the HoloAssist dataset into 4 generalization settings: object-level, action-level, task-level and scene-level, spanning 24 action categories, 120+ object categories with 800 tasks performed by different subjects in 2 geographical regions (Redmond and Zurich). HoloAssist also provides temporally localized action clips with text description, with each atomic action lasting 1-2 seconds. We report the number of sequences, object categories, action categories and tasks in different splits of all the generalization settings considered in our experiments in Tab. 1.

3D Hand poses: Lab datasets compute 3D hand annotations in 2 ways: marker-based motion capture setup (ARCTIC [1], GRAB [10]) or multiview optimization from RGB-D (HO3D [2, 3], H2O [5]) or RGB (FreiHand [12]) images. While marker-based MoCap setup gives sub-mm level accuracy, getting 3D annotations in the absence of these markers in very challenging. Multiview optimization methods on lab datasets typically require RGB-D images [2, 3] or human-in-the-loop [12] and incur an error of around 1 cm. The hand poses in these datasets is significantly simpler than our setting due to constrained capture setup. For single RGB images in the wild, SMPLify [7] fits SMPL-X [7] model (consisting of body, hands & faces) using 2D features and MoCap priors, resulting in 3D mesh vertex & joint error of 5-6 cm. This highlights the challenging nature of obtaining 3D hand annotations in the wild. We also operate in the wild and our data engine uses HaMeR [8] to get the 3D hand poses from RGB images. While HoloAssist provides 3D joints positions from hand sensors on the HoloLens [6], it is not very accurate in hand-object interaction settings and is not synchronized with the RGB images (non-linear offset between the hand sensor stream & RGB stream since the hand sensor needs to maintain a constant fps). Thus, we can only get an approximate measure of the accuracy of the 3D hand poses using the hand sensor data from HoloLens. We notice a 3D position error of 4.75 cm for the hand centroid, which reinforces the challenging nature of obtaining 3D hand poses in the wild, as observed in earlier works [7].

Contact Maps: We represent the contact points as a binary mask over the hand mesh, *i.e.*, contact maps. These are estimated by projecting the 3D hand mesh vertices into the image (using known camera parameters) and considering each vertex which projects into the 2D contact region (overlap between hand & object masks with padding applied at the boundary) as 3D contact point. Note that we do not consider depth of different vertices during projections so 2 vertices (*e.g.* on the front and back of the finger) can project to the same 2D pixel and be considered as contact points. To verify the these contact maps, we consider a part-level classification task where each hand vertex is assigned to one of 16 parts (associated with different joints) of the hand (defined using the adjacency matrix available in MANO [9]), *i.e.*, a part-level map. We consider the images from HO3D [2] which provide 3D contact point annotations. We run the contact module of our data engine on these images to get the contact map and convert it into a part-level map. We then compare the predicted joint map with the ground truth part-level map, resulting in a precision of 0.67, recall of 0.95 and f1 score of 0.76. Note that these are computed only for the images involving contact between the hand and object.

2. Additional Analysis

In our experiments, we study generalization w.r.t. 4 aspects: novel objects, actions, tasks, and scene and report 3 metrics:

[*]Part of the work was done during an internship at Microsoft
[†]Joint last authors, indicates equal contribution

	Object-level			Action-level			Task-level			Scene-level		
	Train	Val	Test	Train	Val	Test	Train	Val	Test	Train	Val	Test
# Sequences	17518	1149	2607	18495	471	2308	17209	2106	1959	13028	4325	3921
# Object classes	108	7	16	127	40	80	129	57	58	67	112	109
# Action classes	24	20	17	17	2	5	24	16	17	22	22	21
# Tasks	710	65	82	674	41	142	685	86	86	464	569	529

Table 1. **Dataset Statistics**. We report the number of sequences, object categories, action categories and tasks in different splits of all the generalization settings considered in our experiments.

MPJPE (M-PE: cm), MPJPE-PA (M-PA: cm) & F1 score (for contact maps) to measure the accuracy of the predicted trajectories. We adapt two recent methods from human pose literature to work with image inputs, *i.e.*, HCTFormer: ViT encoder to extract features from image followed by a transformer decoder (similar to pose decoder of T2P [4]), HCT-Diff: MDM [11] modified to predict interaction trajectories from image inputs. We consider 2 variants of our setting: hand is visible in the image, hand is not present (often the case at the start of an interaction).

In the main paper, we report results for T=30 timesteps. Here, we show results for $T = 16$ timesteps in Tab. 2 & Tab. 3.We observe same trends as the main paper, *i.e.*, LatentAct leads to consistent gains in absolute hand poses & contact maps compared to other baselines and training InterPred with diffusion losses leads to better hand poses but worse contact maps w.r.t. LatentAct, across all settings.

3. Visualizations

We show the predictions from LatentAct and the best baseline on the task-level generalization for both forecasting (Fig. 1, Fig. 2) & interpolation (Fig. 3, Fig. 4) settings for 3 timesteps (t=5, 10, 15). The left column shows the input image with the contact point (projected in the image in cyan blob), goal image (for interpolation setting) & action text. The other columns show the hand poses & contacts for the baseline, ground truth & our model: (a) Camera View: this captures the placement of the hand around the contact point in the camera view, (b) Another View: visualizations from a different camera viewpoint, with the contact point at the center, (c) Contact Maps are shown as red parts of the hand mesh. LatentAct leads to better placement in the scene, hand poses & sharper contact maps than the baseline.

We also show failure cases in Fig. 5. We observe that predictions are near-static and hand poses are inaccurate in some cases. Moreover, the MPJPE error is dominated by the translation components, indicating high error in the absolute hand position. These failure modes are visible in all models. This also highlights the challenging nature of the task since the model needs to predict the 3D trajectory from a single image, which is inherently ambiguous.

References

[1] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[2] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[3] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[4] Jaewoo Jeong, Daehee Park, and Kuk-Jin Yoon. Multi-agent long-term 3d human pose forecasting via interaction-aware trajectory conditioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3

[5] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 1

[6] Micrsoft. Hololens 2. https://learn.microsoft.com/en-us/hololens/hololens2-hardware, 2023. 1

[7] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[8] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[9] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 2017. 1

[10] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

	Method	Task-level			Object-level			Action-level			Scene-level		
		M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑
Hand visible	**Forecasting**												
	HCTFormer	7.63	2.48	0.73	7.72	2.51	0.75	7.89	2.60	0.75	7.79	2.55	0.74
	HCTDiff	7.85	**2.25**	0.73	11.58	**2.17**	0.75	8.06	**2.38**	0.75	8.43	**2.29**	0.73
	LatentAct (Ours)	**7.41**	2.43	**0.76**	**7.30**	2.43	**0.77**	**7.39**	2.62	**0.77**	**7.53**	2.53	**0.75**
	Interpolation												
	HCTFormer	7.11	2.40	0.78	7.25	2.48	0.78	7.20	2.58	0.78	7.21	2.49	**0.78**
	HCTDiff	7.20	**2.21**	0.78	7.38	**2.30**	0.78	7.36	**2.44**	0.79	7.56	**2.35**	0.77
	LatentAct (Ours)	**6.66**	2.34	**0.79**	**6.63**	2.40	**0.80**	**7.00**	2.52	**0.80**	**6.93**	2.48	**0.78**
No hands	**Forecasting**												
	HCTFormer	7.64	2.43	0.73	7.72	2.50	0.74	7.53	2.55	0.75	7.72	2.52	0.73
	HCTDiff	7.79	**2.17**	0.72	8.19	**2.19**	0.73	7.96	**2.33**	0.74	9.52	**2.24**	0.72
	LatentAct (Ours)	**7.25**	2.40	**0.76**	**7.41**	2.45	**0.77**	**7.60**	2.56	**0.77**	**7.43**	2.50	**0.75**
	Interpolation												
	HCTFormer	7.04	2.42	0.77	7.13	2.46	0.78	7.41	2.56	0.78	7.14	2.54	0.77
	HCTDiff	7.53	**2.27**	0.77	7.57	**2.26**	0.77	7.92	**2.38**	0.77	7.41	**2.32**	0.77
	LatentAct (Ours)	**6.74**	2.33	**0.79**	**6.82**	2.39	**0.79**	**7.04**	2.52	**0.80**	**7.04**	2.44	**0.78**

Table 2. **Generalization results**. We report MPJPE (M-PE: cm), MPJPE-PA (M-PA: cm) & F1-score in 4 generalization settings: novel tasks, objects, actions & scene, to measure the accuracy of the predicted trajectories. We adapt two recent methods from human pose literature to work with image inputs, *i.e.*, HCTFormer: ViT encoded image features passed to a transformer decoder (similar to pose decoder of T2P [4]), HCTDiff: MDM [11] modified to take image features as well. LatentAct leads to better absolute hand poses & contact maps. Here, we consider $T = 16$ timesteps.

	Method	Task-level			Object-level			Action-level			Scene-level		
		M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑
Hand visible	**Forecasting**												
	LatentAct	7.41	2.43	0.76	7.30	2.43	0.77	7.39	2.62	0.77	7.53	2.53	0.75
	LatentAct-Diff	6.85	2.37	0.73	7.17	2.40	0.61	7.40	2.64	0.75	7.63	2.47	0.74
	Interpolation												
	LatentAct	6.66	2.34	0.79	6.63	2.40	0.80	7.00	2.52	0.80	6.93	2.48	0.78
	LatentAct-Diff	6.55	2.17	0.78	7.06	2.26	0.78	6.91	2.44	0.79	7.57	2.36	0.78
No hands	**Forecasting**												
	LatentAct	7.25	2.40	0.76	7.41	2.45	0.77	7.60	2.56	0.77	7.43	2.50	0.75
	LatentAct-Diff	7.23	2.43	0.72	7.46	2.32	0.75	7.69	2.46	0.74	7.51	2.39	0.73
	Interpolation												
	LatentAct	6.74	2.33	0.79	6.82	2.39	0.79	7.04	2.52	0.80	7.04	2.44	0.78
	LatentAct-Diff	6.85	2.27	0.77	7.24	2.31	0.77	7.45	2.38	0.78	6.97	2.34	0.78

Table 3. **LatentAct-Diff trends**. Training InterPred with diffusion loss leads to better hand poses but worse contact maps than LatentAct.

[11] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 2, 3

[12] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan C. Russell, Max J. Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single RGB images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 1

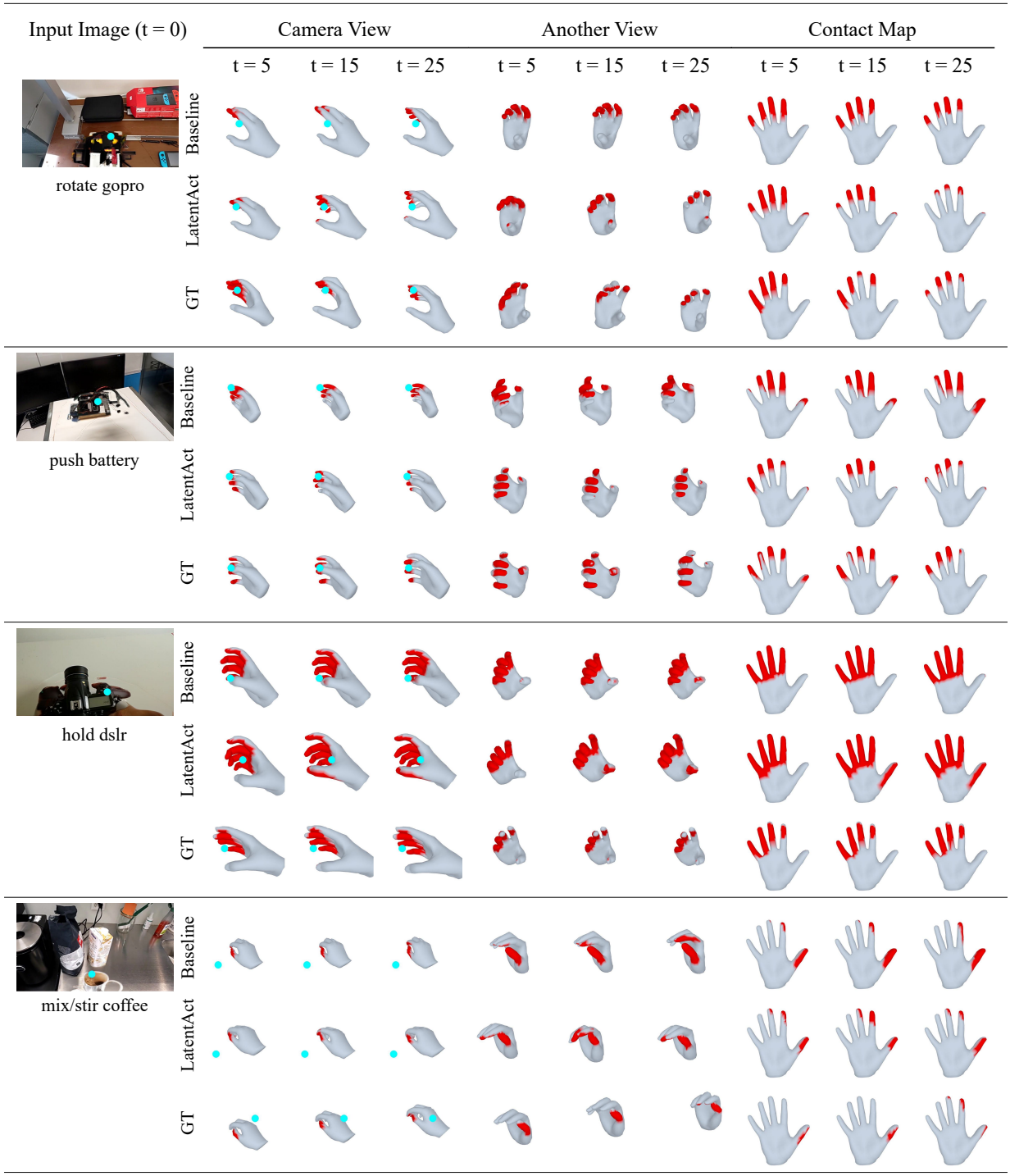

Figure 1. **Visualizations**. We compare the predictions of LatentAct & the best baseline on a few examples from forecasting (no hand) setting for 3 timesteps (t=5, 15, 25). The left column shows the input image with the contact point (projected in the image in cyan blob) & text describing the action. The other columns show: (a) Camera View: predicted hand in the camera view with the contact point, this captures the placement of the hand around the contact point, (b) Another View: this better visualizes the hand pose from a different camera viewpoint, (c) Contact Maps are shown as red parts of the hand mesh. LatentAct leads to better orientation of hand & sharper contact maps than the baseline.

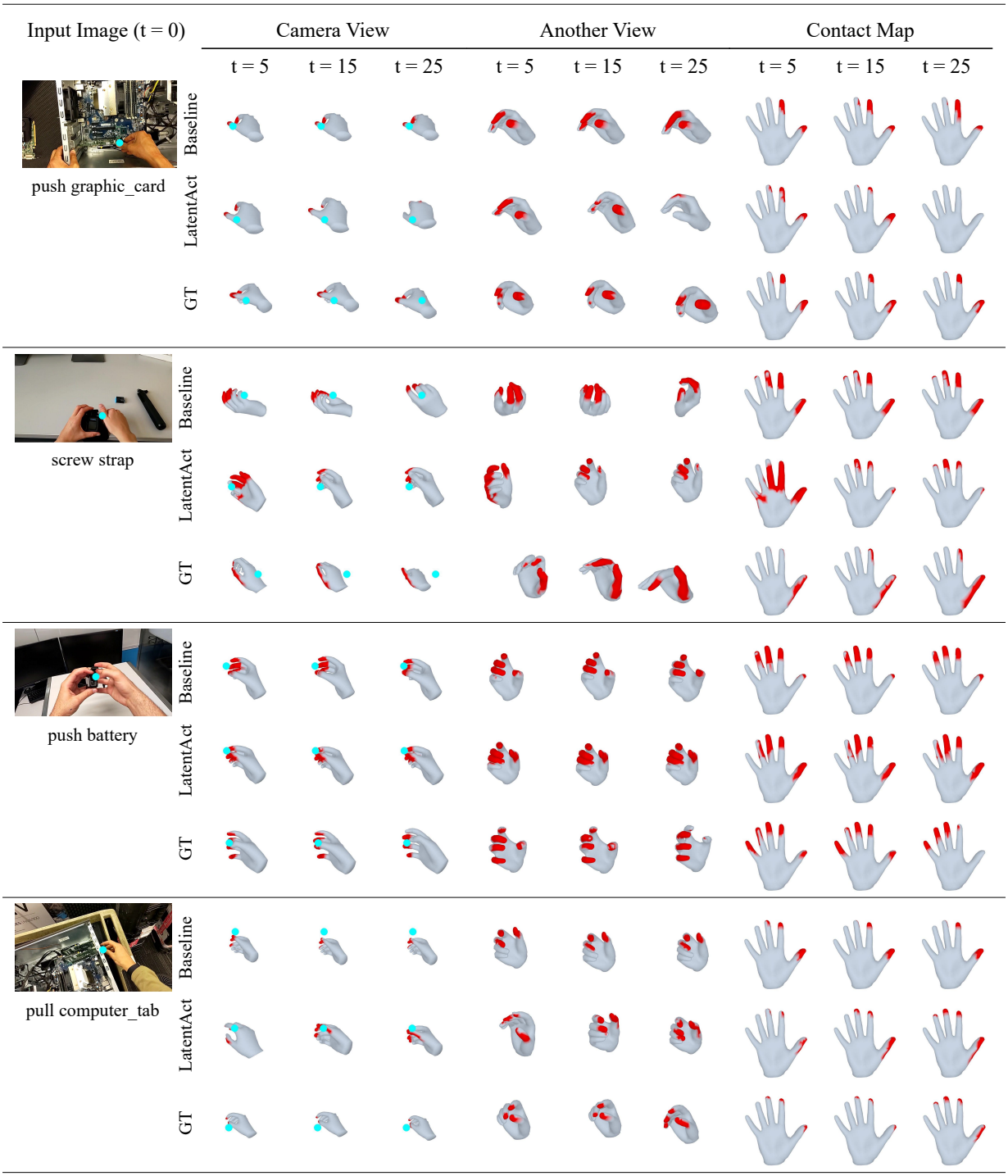

Figure 2. **Visualizations**. We compare the predictions of LatentAct & the best baseline on a few examples from forecasting (hand visible) setting for 3 timesteps (t=5, 15, 25). The left column shows the input image with the contact point (projected in the image in cyan blob) & text describing the action. The other columns show: (a) Camera View: predicted hand in the camera view with the contact point, this captures the placement of the hand around the contact point, (b) Another View: this better visualizes the hand pose from a different camera viewpoint, (c) Contact Maps are shown as red parts of the hand mesh. LatentAct leads to better orientation of hand & sharper contact maps than the baseline.

Figure 3. **Visualizations**. We compare the predictions of LatentAct & the best baseline on a few examples from interpolation (no hand) setting for 3 timesteps (t=5, 15, 25). The left column shows the input image with the contact point (projected in the image in cyan blob), goal image & text describing the action. The other columns show: (a) Camera View: predicted hand in the camera view with the contact point, this captures the placement of the hand around the contact point, (b) Another View: this better visualizes the hand pose from a different camera viewpoint, (c) Contact Maps are shown as red parts of the hand mesh. LatentAct leads to better orientation of hand & sharper contact maps than the baseline.

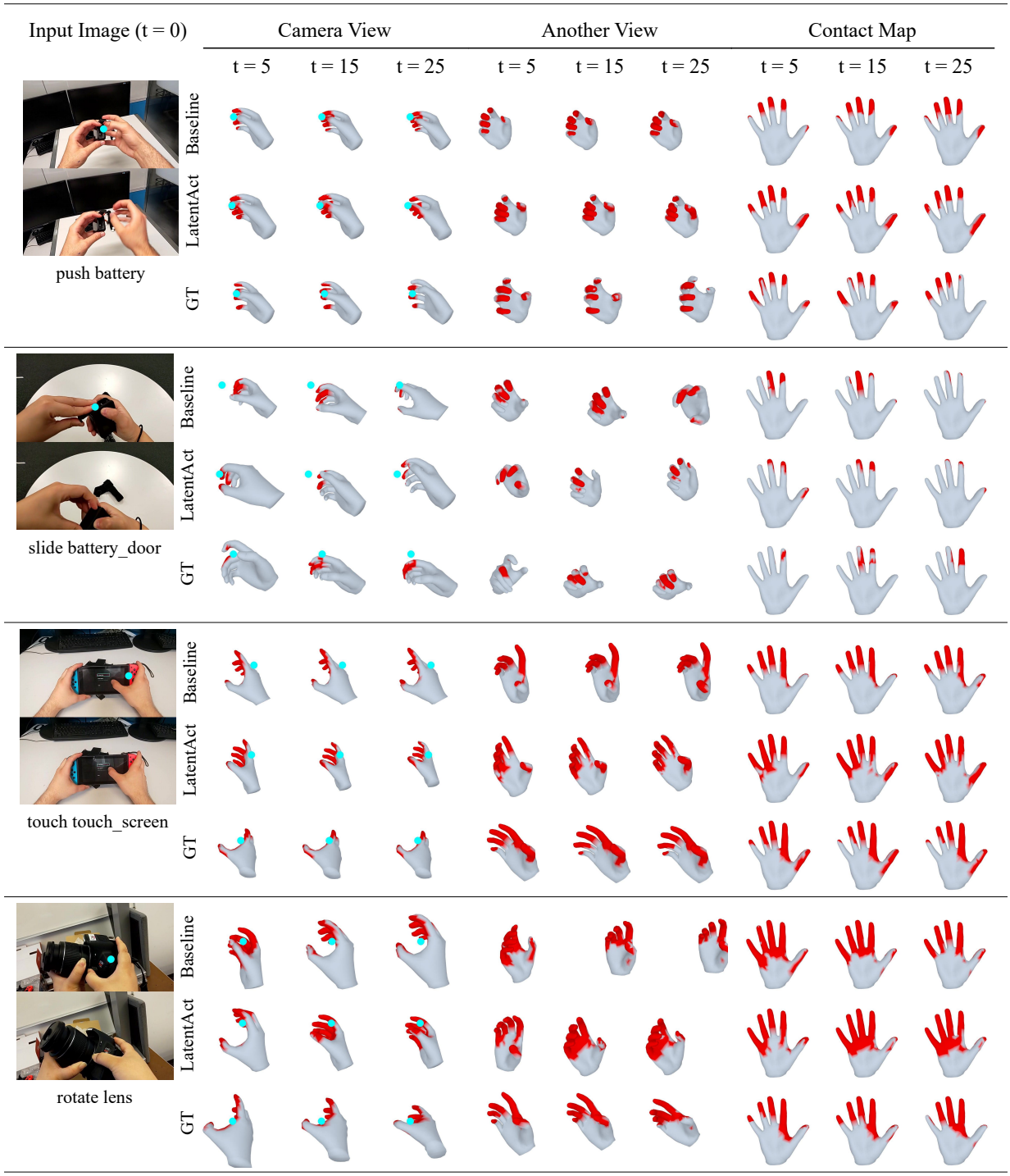

Figure 4. **Visualizations**. We compare the predictions of LatentAct & the best baseline on a few examples from interpolation (hand visible) setting for 3 timesteps (t=5, 15, 25). The left column shows the input image with the contact point (projected in the image in cyan blob), goal image & text describing the action. The other columns show: (a) Camera View: predicted hand in the camera view with the contact point, this captures the placement of the hand around the contact point, (b) Another View: this better visualizes the hand pose from a different camera viewpoint, (c) Contact Maps are shown as red parts of the hand mesh. LatentAct leads to better orientation of hand & sharper contact maps than the baseline.

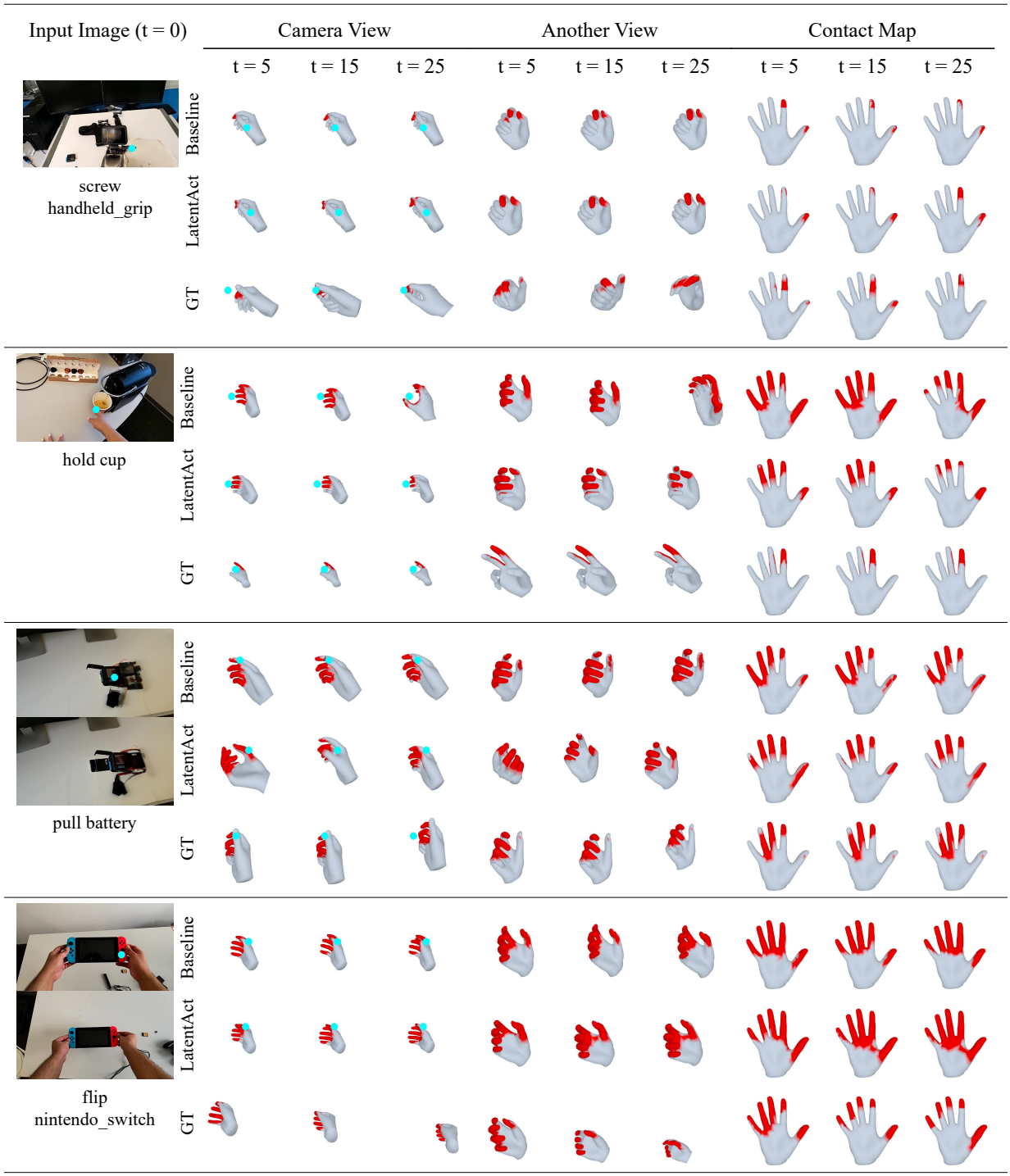

Figure 5. **Failure cases**. We show the predictions of LatentAct & the best baseline on a few examples from all settings for 3 timesteps (t=5, 15, 25). The left column shows the input image with the contact point (projected in the image in cyan blob), goal image (for interpolation task) & text describing the action. The other columns show: (a) Camera View: predicted hand in the camera view with the contact point, this captures the placement of the hand around the contact point, (b) Another View: this better visualizes the hand pose from a different camera viewpoint, (c) Contact Maps are shown as red parts of the hand mesh. We observe that predictions are near-static and hand poses are inaccurate in some cases. *Note that all the experiments are done in generalization settings.*