

# From Image to Video: An Empirical Study of Diffusion Representations

## Supplementary Material

This supplementary material provides further analysis of I-WALT, including ablation studies in Appendix A, using the same setup as V-WALT. Tables illustrating the performance metrics of the WALT model for image and video tasks are reported in Appendix B, which correspond to the same values used to calculate the relative performance metrics in Fig. 6 of the main paper. We also present visualizations of depth estimation and box tracking predictions for both models in Appendix C. Details of the datasets and readouts are explained in Appendix D, while training settings are described in Appendix E.

### A. I-WALT ablations

To evaluate the image-pretrained model, I-WALT (introduced in Sec. 3.3), we replicate the ablation study from Sec. 4.3 for the video-pretrained counterpart, V-WALT.

**Noise level.** Figure 1 (left) illustrates the readout performance of I-WALT at different noise levels. Following the noise level ablation for V-WALT, the block for feature extraction is set to  $l = 16$ . Similar to what was reported for V-WALT in Sec. 4.3, I-WALT benefits from addition of moderate noise levels (between 0 and 200 timesteps).

**Model block.** We analyzed the downstream task performance of different layers as shown in Fig. 1 (right). As in the model block ablation for V-WALT, the noise timestep is set to  $t = 200$ . Most layers exhibit performance comparable to V-WALT, but point tracking shows a notable difference. For this task, the earlier layers of the model achieve the highest accuracy. This suggests that the features relevant for point tracking are learned early in the I-WALT transformer.

**Training budget.** Figure 2 shows the training progress of I-WALT using a setup similar to the one described in Sec. 4.4. We observe a comparable behavior to V-WALT, except for camera pose estimation, where the performance of I-WALT improves with longer training. This suggests that I-WALT does not overfit in this setting, which is further supported by the moderate to strong positive correlations between performance, loss, and FID values over training time, as shown in Tab. 1. Additionally, we include the values for V-WALT in Tab. 2, which demonstrate a similar correlation pattern, except for camera pose estimation, as mentioned in Sec. 4.4.

### B. Baselines results

The performance metrics of I-WALT and V-WALT for downstream tasks are provided in Tab. 3 and Tab. 4, respectively. These tables correspond to the values used to generate the relative performance metrics presented in Fig. 6 of the main paper. For example, accuracy is used for SSv2 action recognition. See Sec. 3.4 for the full list of metrics used for each downstream task.

### C. Qualitative results

In order to qualitatively assess the performance of I-WALT compared to V-WALT, we include prediction outputs of all WALT models for the tasks of monocular depth prediction (Fig. 3), box tracking (Fig. 4) and action recognition.

	SSv2	K400	K700	PointT.	Cam. P.	Depth	Obj. T.
<i>Pearson coefficient</i>							
Loss	0.96	0.93	0.96	0.64	0.90	0.94	0.66
FID	0.96	0.91	0.95	0.69	0.90	0.94	0.72
<i>Spearman rank correlation coefficient</i>							
Loss	0.92	0.87	0.94	0.51	0.89	0.96	0.44
FID	0.93	0.86	0.94	0.50	0.91	0.96	0.47

Table 1. **Generation quality vs. performance for I-WALT** – The first row presents Pearson correlation coefficients between downstream performance metrics and training loss. The second row displays the same correlation metric, but calculated between downstream performance and generation quality, as measured by FID. The third and fourth row correspond to similar results with the Spearman correlation coefficient.

	SSv2	K400	K700	PointT.	Cam. P.	Depth	Obj. T.
<i>Pearson coefficient</i>							
Loss	0.94	0.93	0.95	0.62	0.11	0.76	0.84
FVD	0.45	0.42	0.41	0.20	-0.55	0.46	0.47
<i>Spearman rank correlation coefficient</i>							
Loss	0.94	0.89	0.97	0.27	-0.07	0.61	0.72
FVD	0.63	0.70	0.66	0.27	-0.50	0.45	0.52

Table 2. **Generation quality vs. performance for V-WALT** – The first row presents Pearson correlation coefficients between downstream performance metrics and training loss. The second row displays the same correlation metric, but calculated between downstream performance and generation quality, as measured by FVD [53]. The third and fourth row correspond to similar results with the Spearman correlation coefficient.

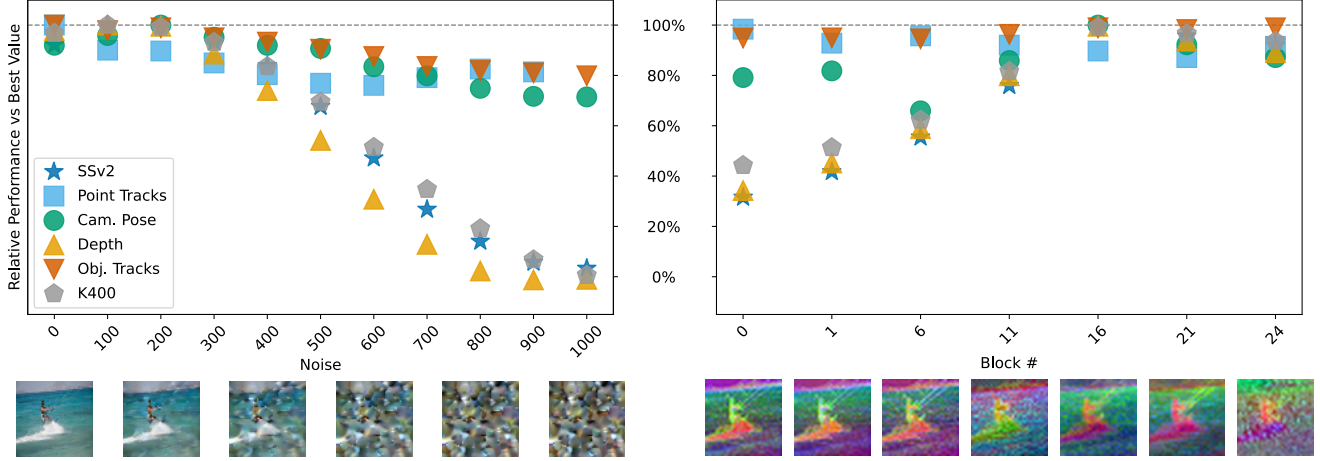


Figure 1. **Influence of Noise and Block Choice on Readout Performance of I-WALT** – Relative change in downstream task performance when probing different noise levels (left, fixed block  $l = 16$ ) and intermediate WALT blocks (right, fixed noise  $t = 200$ ). Values below -10% are excluded for clarity. Optimal performance is generally observed with noise timesteps between 0 and 200 and blocks 11-16. Example noisy images (left) and PCA visualizations (right) are shown below the plots.

tion (Fig. 5). As demonstrated in Sec. 4.2, V-WALT consistently outperforms I-WALT across all tasks, particularly those requiring spatiotemporal understanding. This is evident in the enhanced accuracy of V-WALT predictions, such as depth maps exhibiting greater pixel correspondence, even with the V-WALT 218M model, which has the same number of parameters as I-WALT. Similarly, V-WALT shows superior object tracking performance, while I-WALT struggles to accurately estimate the size and location of bounding

boxes. The temporally rich input data enables V-WALT to achieve higher accuracy in action recognition. For instance, V-WALT correctly classifies the action of “squeezing something” (first column), which occurs midway through the video clip, whereas I-WALT fails to do so.

## D. Datasets and readout details

A summary of the readout architectures and their corresponding parameter counts for the image and video tasks is provided in Tab. 5 and Tab. 6, respectively.

### D.1. Image classification

**Dataset.** Image classification results are reported on *ImageNet* [44], *Places365* [70], and *Inat2018* [55].

*ImageNet* [44] is a large-scale dataset containing 1,000 object and animal categories, with 1,281,167 images for training and 50,000 images for validation. For training on *ImageNet*, we follow the traditional augmentation scheme consisting of inception cropping and random horizontal flipping. For evaluation, a  $224 \times 224$  center crop is extracted from each image whose shorter edge is first resized to 256.

*Places365* is a scene classification dataset that contains images with buildings, landscapes, and other everyday scenarios. It includes 1.8 million training images and 36,500 validation images across 365 scene classes.

*Inat2018* contains 437,513 images for training and 24,426 images for validation, featuring 8,142 classes of visually similar species across a wide range of taxonomic groups, such as plants, animals, fungi, and insects. The dataset also exhibits heterogeneous image quality due to diverse camera sources and exhibits a substantial class imbalance.



Figure 2. **Impact of pre-training progress on downstream task performance of I-WALT** – Recognition tasks generally improve with longer training, while tasks like tracking and depth estimation show optimal performance at earlier stages. Performance is evaluated across a range of tasks and compared to training loss and Fréchet Inception Distance (FID).

Model	<i>Places365</i> ↑	<i>ImageNet</i> ↑	<i>Inat2018</i> ↑
<i>Methods pretrained on image tasks</i>			
I-JEPA-600M	0.514	0.732	0.421
ImageMAE-600M	0.488	0.749	0.479
SigLIP 1.7B	0.559	0.852	0.479
DinoV2-300M	0.567	0.854	0.726
I-WALT 284M	0.461	0.607	0.260
<i>Methods pretrained on video tasks</i>			
V-JEPA-300M	0.509	0.678	0.349
V-JEPA-600M	0.515	0.688	0.360
VideoMAEv1-600M	0.499	0.643	0.321
V-WALT 284M	0.464	0.618	0.290
V-WALT 1.9B	0.512	0.728	0.414

Table 3. **Comparison with state-of-the-art methods on image recognition tasks** – All results presented here were obtained using the same training and evaluation protocol with frozen backbones and trainable readouts.

Model	<i>K400</i> ↑	<i>K700</i> ↑	<i>Depth</i> ↓	<i>Obj. Tracks</i> ↑	<i>SSv2</i> ↑	<i>Cam. Pose</i> ↓	<i>PointTracks</i> ↑
<i>Methods pretrained on image tasks</i>							
I-JEPA-600M	0.617	0.485	0.147	0.483	0.451	2.299	0.515
ImageMAE-600M	0.612	0.496	0.117	0.501	0.458	2.197	0.566
SigLIP 1.7B	0.760	0.663	0.154	0.464	0.448	2.442	0.396
DinoV2-300M	0.702	0.604	0.108	0.502	0.507	2.307	0.526
I-WALT 284M	0.527	0.396	0.199	0.459	0.360	2.095	0.449
<i>Methods pretrained on video tasks</i>							
V-JEPA-300M	0.685	0.557	0.132	0.629	0.658	0.507	0.733
V-JEPA-600M	0.696	0.572	0.123	0.620	0.684	0.409	0.737
VideoMAEv1-600M	0.675	0.543	0.117	0.620	0.665	0.583	0.708
V-WALT 284M	0.552	0.414	0.185	0.586	0.510	0.826	0.756
V-WALT 724M	0.571	0.445	0.151	0.587	0.547	0.814	0.741
V-WALT 1.9B	0.615	0.488	0.124	0.597	0.597	0.617	0.735

Table 4. **Comparison with state-of-the-art methods on video recognition, depth estimation, tracking, and camera pose estimation tasks** – All results presented here were obtained using the same training and evaluation protocol with frozen backbones and trainable readouts.

The same data augmentation and preprocessing steps used for *ImageNet* are applied to both *Inat2018* and *Places365*. For all the image classification datasets, the original data splits are used for training and evaluating the classification readout.

**Readout.** In the case of V-WALT, an image is replicated across the 17 temporal input channels and the model features are averaged before passing them to the readout. I-WALT does not require any image replication and the model features are directly passed to the readout.

Adopting the approach of V-JEPA [5], we utilize a cross-attention block with a learnable query token to extract class information from the model features. This token attends to

the features within the cross-attention block, and its output is fed into a linear classifier for class prediction. The readout is trained with the softmax cross-entropy loss.

## D.2. Action recognition

**Dataset.** Something-Something-V2 (SSv2) [20] is a large-scale dataset consisting of short videos (2-6 seconds at 12 frames per second) depicting diverse human actions with everyday objects. The dataset is specifically designed for fine-grained understanding of human hand gestures, focusing on subtle actions, such as placing objects into containers. Something-Something-V2 encompasses 174 categories with 168,913 samples in the training set and 24,777 in the validation set.

Kinetics-400 (*K400*) is a large-scale dataset of YouTube

Task	Architecture	Number of parameters
<i>ImageNet</i>	CrossAttention( qkv_size=768, num_heads=12)	7,678,184
	Dense(output_size=1000)	
<i>Places365</i>	CrossAttention( qkv_size=768, num_heads=12)	7,189,869
	Dense(output_size=365)	
<i>Inat2018</i>	CrossAttention( qkv_size=768, num_heads=12)	13,170,382
	Dense(output_size=8142)	

Table 5. Architecture details and number of parameters for the image classification readouts.

videos designed for human action recognition, encompassing object manipulation, human-object interaction, and body movements. Kinetics-400 consists of 246,245 training videos and  $\sim 20K$  validation videos with an average duration of 10 seconds. All video clips are labeled into 400 classes.

Kinetics-700 (*K700*) is an extension of Kinetics-400. The data collection pipeline between the two datasets differs in how action classes are sourced, how videos are matched with classes, and the human verification process. Kinetics-700 contains 545,317 training videos and 35,000 validation videos across 700 fine-grained action classes.

For both training and evaluation of the readout, 17 frames are sampled from each video to generate the model input. A stride of 2 is used for *SSv2*, while a stride of 1 is used for *K400* and *K700*. For all the action recognition datasets, the original data splits are used for training and evaluating the readout.

**Readout.** The same attention readout used for image classification and described above is also used for action recognition.

### D.3. Monocular depth prediction

**Dataset.** For this work, we utilize the train and validation splits of the ScanNet dataset [15], comprising 1,201 and 312 videos respectively. The dataset offers high-resolution RGB frames (1296x968) in diverse indoor environments and corresponding depth frames (640x480) captured with an RGB-D system. The input to the WALT model is obtained by sampling 17 consecutive frames from the ScanNet videos. During training, the starting frame is chosen randomly. For evaluation, sampling begins at frame 0.

**Readout.** We use the readout from [45], which applies cross-attention with spatial coordinates as queries to each frame independently. The readout is trained using an L2 loss between predicted and ground truth depth maps.

### D.4. Relative camera pose estimation

**Dataset.** RealEstate10K [71] is a dataset of property walkthrough videos with intrinsic and extrinsic camera parameters using Structure from Motion (SfM). The clips were gathered from YouTube and typically feature smooth camera movement with minimal camera roll or pitch. The original splits of the dataset are used, which consist of roughly 10 million training frames from 6,500 videos and 1 million test frames from 696 videos.

**Readout.** The input of the readout is formed by concatenating the video representations of the first and last frame of the video sequences. These are then processed via cross-attention with learned latent vectors and a linear layer to produce 12-dimensional vectors representing  $SE(3)$  pose transformations, which correspond to a  $3 \times 3$  rotation matrix and a  $3 \times 1$  translation vector. The predicted rotation matrix is refined using the Procrustes algorithm [7] to ensure it represents a valid  $SO(3)$  rotation before metric evaluation. Training is performed by minimizing the L2 loss between predicted and ground-truth pose matrices.

### D.5. Visual correspondence – Point tracking

**Dataset.** The Perception Test dataset [38] was specifically designed to evaluate the perception and reasoning skills of multimodal video models. It was filmed by around 100 participants worldwide and contains perceptually interesting situations. In this paper, the Perception Test dataset is

used to evaluate the point tracking task. Specifically, the validation set is employed, which comprises 73 real-world videos, averaging 722 frames in length, with multiple point tracks annotated using the same protocol as in [56]. Each point is visible in approximately 480 frames. The first 17 frames of each video are sampled with a stride of 4 to generate the model input.

The point tracking readout head is trained with the training set of the Kubric MOVIE dataset [21], which contains 97,500 synthetic 24-frame videos, each depicting scenes with 10-20 static and 1-3 dynamic objects rendered against photorealistic backgrounds. The camera in these videos moves on a straight line at a constant velocity, always pointed towards the origin of the scene.

**Readout.** To build the readout input, pretrained model features are first interpolated in the temporal dimension to match the number of video frames. The interpolated features and a set of query points become the readout input. During training, 17 frames and 64 point tracks are randomly sampled from each video. Then, a random crop with an area between 30% and 100% of the original frame and an aspect ratio between 0.5 and 2.0 is extracted. The crops are then resized to  $128 \times 128$ . Query points are selected exclusively from the first frame.

For evaluation, we sample the first 17 frames with a stride of 4 from each video and use 64 point tracks. As in [56], our evaluation takes the first visible point track as the query, and discards frames preceding its appearance.

The readout head employs an iteratively applied cross-attention transformer, maintaining a 512-dimensional latent state for each point track between frames. As in [56], this state is initialized from query point positions using a Fourier positional encoding followed by a two-layer MLP. The transformer comprises three layers of cross-attention with eight heads and a key/value size of 512. At each step, it uses the latent state as queries to attend to the frame features generated by the pretrained model. A two-layer MLP predicts the position, visibility, and uncertainty of each point track in each frame using the corresponding latent states as input. The loss function is a combination of a Huber loss for location accuracy and Sigmoid Binary Cross Entropy losses for visibility and certainty. Points that have exited the scene contribute only to the visibility loss.

## D.6. Visual correspondence – Box tracking

**Dataset.** We leverage the Waymo Open Dataset [50], utilizing the high-resolution (1280x1920) RGB video data captured at 10 fps. This dataset, recorded from Waymo vehicles in urban and suburban settings, includes 2D and 3D bounding box annotations. We use the 2D bounding boxes for loss calculation and metric evaluation. The training and validation sets comprise 798 and 202 samples, respectively,

each 20 seconds in duration. The same data splits are used for training and evaluating the box tracking readout.

**Readout.** Consistent with prior work [56], for both training and evaluation, we downsample videos to  $256 \times 384$  resolution at 5 fps, and then extract a central  $256 \times 256$  spatial crop and a random 17-frame temporal crop. Bounding boxes smaller than 0.5% of the first sampled frame area are discarded, and a maximum of 25 boxes are retained per sample.

The same attention readout used for point tracking and described above is also used for box tracking, only differing in the predictions. The position  $x_{min}, x_{max}, y_{min}, y_{max}$  of query boxes is predicted for box tracking. The box tracking readout is trained using an L2 loss between the predicted and normalized box coordinates. As in point tracking, the pretrained model features are interpolated in the temporal dimension to match the number of video frames.

## D.7. Pretrained Model

**Datasets.** As specified in [22], the WALT pretrained model was trained jointly on text-image and text-video pairs. The dataset consists of  $\sim 970$ M text-image pairs and  $\sim 89$ M text-video pairs from the public internet and internal sources, with no overlap with our evaluation datasets.

## E. Training settings

Table 7 summarizes the hyperparameters used to train the readout heads for each task described in Appendix D. We use the AdamW optimizer with a cosine learning rate decay schedule, initialized with a linear warmup over 1,000 steps (from 0 to  $3e-4$ ), and subsequently decaying to  $1e-7$ . Batch sizes are 32 for video tasks, 512 for *Places365* and *Inat2018*, and 64 for *ImageNet*.

Task	Architecture	Number of parameters
SSv2 Action Recognition	CrossAttention( qkv.size=768, num.heads=12)  Dense(output.size=174)	7,042,990
K400 Action Recognition	CrossAttention( qkv.size=768, num.heads=12)  Dense(output.size=400)	11,975,056
K700 Action Recognition	CrossAttention( qkv.size=768, num.heads=12)  Dense(output.size=700)	12,282,556
Relative camera pose estimation	CrossAttention( qkv.size=256, num.heads=8)  Dense(output.size=12)	1,650,444
Monocular depth prediction	CrossAttentionTransformer( qkv.size=512, num.heads=2, mlp.size=512, num.layers=1)	3,284,353
Waymo Object Tracking	CrossAttentionTransformer( qkv.size=512, num.heads=8, mlp.size=2048, num.layers=3)  predictor=MLP( hidden.size=512, output.size=512)  output=MLP( hidden.size=512, output.size=6)	12,931,462
Point Tracking	CrossAttentionTransformer( qkv.size=512, num.heads=8, mlp.size=2048, num.layers=3)  predictor=MLP( hidden.size=512, output.size=512)  output=MLP( hidden.size=512, output.size=4)	12,897,668

Table 6. **Readout heads setup** – Architecture details and number of parameters for the readouts of the video tasks.



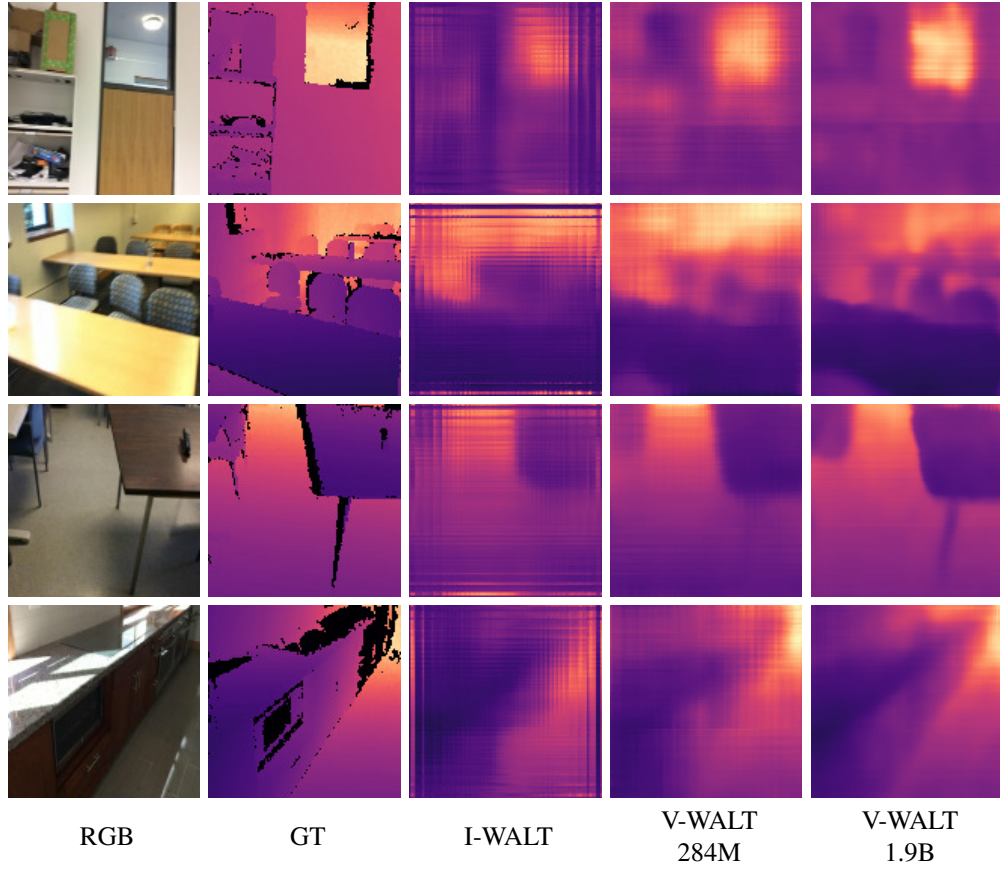


Figure 3. **Depth predictions from I-WALT and V-WALT** – RGB, Ground Truth, and Predictions of I-WALT and V-WALT (284M and 1.9B) models. The results of V-WALT 1.9B are resized to a square aspect ratio for visualization purposes.

	Places365	iNat2018	ImageNet	Video Datasets
batch size	512	512	64	32
training steps	10,000	10,000	200,182	40,000
optimizer	adamw	adamw	adamw	adamw
$\epsilon$	1e-8	1e-8	1e-8	1e-8
$\beta_1$	0.9	0.9	0.9	0.9
$\beta_2$	0.999	0.999	0.999	0.999
weight decay	1e-4	1e-4	1e-4	1e-4
lr schedule	cosine	cosine	cosine	cosine
init. lr	0	0	0	0
peak lr	3e-4	3e-4	3e-4	3e-4
warmup steps	1,000	1,000	1,000	1,000
lr end value	1e-7	1e-7	1e-7	1e-7

Table 7. **Readout hyperparameters** – Parameters used to train the image and video readout heads using a frozen pretrained WALT model.



Figure 4. **Object Tracks predictions from I-WALT and V-WALT** – Ground Truth, and Predictions of I-WALT and V-WALT (284M and 1.9B) models. The results of V-WALT 1.9B are resized to a square aspect ratio for visualization purposes.



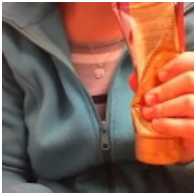



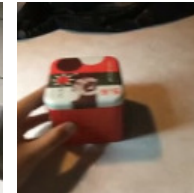
Video					
GT	Squeezing something	Pretending to pick something up	Dropping something next to something	Pretending to throw something	Pushing something from left to right
I-WALT	Holding something	Picking something up	Turning the camera upwards while filming something	Pretending to throw something	Pushing something so that it slightly moves
V-WALT 284M	Squeezing something	Pretending to pick something up	Dropping something next to something	Pretending to throw something	Pushing something from left to right
V-WALT 1.9B	Squeezing something	Pretending to pick something up	Dropping something next to something	Pretending to throw something	Pushing something from left to right

Figure 5. **Action recognition predictions from SSv2 using I-WALT and V-WALT** – The top row displays a single frame from a dataset sample. Frames were manually selected to best showcase the corresponding label.