

Unified Dense Prediction of Video Diffusion

Supplementary Material

This supplementary material provides additional visualization results, details about the Panda-Dense video dense prediction dataset, and thorough descriptions of the training and inference processes for video generation and dense prediction, which includes segmentation and depth estimation. The contents of this supplementary material are organized as follows:

- Panda-Dense Dataset details.
- Detailed training and inference information for two tasks.
- More ablation studies on the model.
- Additional visualization results, including examples of video generation and the dense prediction.

Additionally, please see the mp4 file in our supplementary material to view the [recorded video](#) that provides a concise overview of our paper.

1. Panda-Dense Dataset Details

In Fig. 1, we present the distribution of the number of entities in our video segmentation dataset. The data shows a broad distribution of entities, with most concentrated in the range of 1 to 20. Additionally, our dataset includes numerous dense segmentation samples that contain multiple entities in complex scenes.

2. Training / Inference Details

In our work, we train our model using the Panda-Dense data set and initialize it with the weights of the CogVideoX 5B model. We duplicate the weights from the original 16 channels to accommodate the added input and output channels.

This adjustment modifies the original layers instead of adding extra branches, which would disrupt the existing weights. In other words, this set of weights cannot be expected to directly output two feasible and identical video sequences after adding channels, so complete fine-tuning is required.

Regarding prompts, long captions offer more detailed descriptions and greater control, while short captions provide users with flexibility. We demonstrate the comparison between Panda-70M original captions and our Panda-Dense captions in Fig. 2. To balance strong scene description capabilities with user input convenience, we utilize long captions from our dataset with a probability of 0.8 and short captions from Panda-70M with a probability of 0.2 for each video.

Training is conducted under these conditions for 40000 steps. During inference, we employ DPM++ to execute 50 inference steps, generating 49 video frames along with their

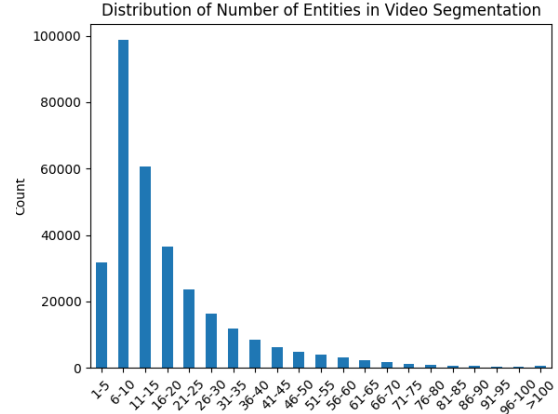


Figure 1. **Distribution of entity number in the video segmentation dataset.** The X-axis represents the groups of the total number of entities in a single video, while the Y-axis represents the total number of videos appearing in this group.

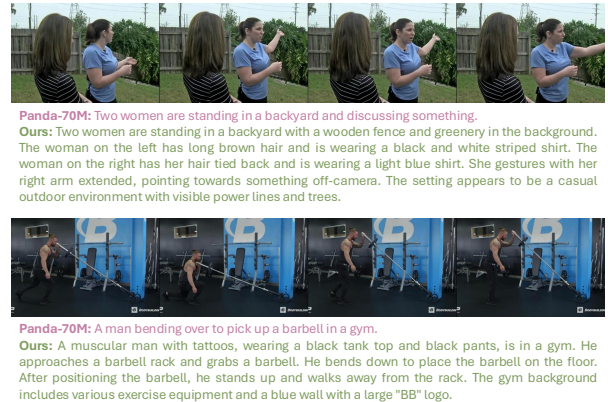


Figure 2. **Caption comparison.** Refer to the same video, compare the original Panda-70M caption and our Panda-Dense caption.

corresponding dense prediction maps.

Video Depth Estimation. During inference, the video and the corresponding depth estimation map are produced, with the depth map appearing as an RGB value map similar to a visualization. Therefore, we should convert the depth map to a single-channel depth value between 0 and 1. Since the depth-to-color projection P_d is a unidirectional function, we can not project the RGB colors back to depth values. To project it back, we sampled 256 RGB values with equal intervals in the 0-1 depth range, as: $P_c = P_d\left(\frac{k}{256}\right)$, for $k = 1, 2, \dots, 256$. Thus, we have the discrete color-to-depth projection P_c . For each pixel, we cal-

Strategy	SC (\uparrow)	BC (\uparrow)	MS (\uparrow)
Random initialization	95.15	95.48	97.99
Duplicate initialization	97.07	96.89	99.23

Table 1. **Ablation study on initialization strategy on additional channel weight.** Different additional channel weight initialization strategy for input and output layers. ‘SC’, ‘BC’ and ‘MS’ stand for subject consistency, background consistency, and motion smoothness.

Strategy	Multi-task	SC (\uparrow)	BC (\uparrow)	MS (\uparrow)
Scratch	-	81.28	76.39	84.46
Scratch	✓	84.32	81.63	89.31
CogVideoX	✓	97.07	96.89	99.23

Table 2. **Ablation study on initialization strategy on whole model weight.** Different model-level weight initialization methods for UDPDiff, comparing initialize from CogVideoX and train from scratch. ‘SC’, ‘BC’ and ‘MS’ stand for subject consistency, background consistency, and motion smoothness.

culate the distance between its RGB value and the available discrete values in P_c , and then project back to the depth value. After propagating the depth for all frames in the video, we obtain a distance-based video depth estimation map.

Multi-task. The multi-task model defines a dictionary mapping different tasks to specific IDs, with segmentation assigned as 0 and depth estimation as 1. This ID serves as the input for the task embedding. During training, we randomly sample different tasks, corresponding to different task IDs and their respective dense prediction maps. During inference, in addition to the caption, a specific task ID must be provided as input. The video channel consistently outputs video, while the dense prediction channel outputs either segmentation or depth estimation under the guidance of task embedding.

3. Ablation Studies

Additional Channel Weight Initialization Method. In our work, we augment the input and output layers with an additional 16 channels to accommodate the inputs and outputs of the dense prediction channels. Specifically, we duplicate the original 16 channels of the input and output layers of CogVideoX and concatenate the weights of the two sets of 16 channels to form 32-channel inputs and outputs. We conduct ablation experiments using the multitask model inference depth estimation to explore different initialization methods for the weights of the additional channels, including using the pre-trained weights from CogVideoX and random initialization. As shown in Tab 1, utilizing the weights



Figure 3. **Converge speed comparison.** Train from scratch, comparing single-task training and multi-task training converge speed. ‘SC’, ‘BC’ and ‘MS’ stand for subject consistency, background consistency, and motion smoothness.

from CogVideoX’s input and output layers as duplicate initialization outperforms random initialization. Although increasing the number of channels and concatenating them disrupts the original weight distribution, using pre-trained weights for dense prediction features still results in better adaptation performance.

Pretraining. Since our method is based on CogVideoX 5B, we directly utilize the weights of CogVideoX 5B’s text-to-video model to initialize all layers except for the input and output layers. For the input and output layers, we continue to adopt the duplicating channel approach by concatenating the original weights of CogVideoX. In this experiment, we compare this strategy with an entirely random initialized multi-task model, both trained for 40000 steps. As shown in Tab 2, the method initialized with CogVideoX outperforms the training from scratch. This is because CogVideoX was trained on a dataset much larger than ours and employed various training tricks. Our fine-tuning approach cannot achieve good results on datasets of this scale. Using pre-trained weights allows the model to learn from an appropriate distribution, resulting in better performance. To further illustrate the advantages of our multi-task training strategy beyond the fine-tuning stage, we compare the performance of a single-task model trained from scratch with that of a multi-task model, as shown in Tab. 2. Multi-task training delivers better performance under the same conditions. Additionally, we present in Fig. 3 the validation scores of the single-task and multi-task models at different steps. The multi-task model achieves higher scores at each step and exhibits a smoother increase in the later stages, converging more rapidly. Therefore, our multi-task training strategy also accelerates the training process.

Model Parameter. The original CogVideoX model is available in 2B and 5B. Our initial experiments utilize the 2B model. Although the 2B model is more efficient, it has a

Model Size	SC (\uparrow)	BC (\uparrow)	MS (\uparrow)
2B	87.29	89.19	91.76
5B	94.98	95.92	98.62

Table 3. **Ablation study on model size.** Compare the performance of our method using CogVideoX at 2B and 5B model sizes. ‘SC’, ‘BC’ and ‘MS’ stand for subject consistency, background consistency, and motion smoothness.

smaller capacity, which may result in insufficient performance compared to the larger model. For our experiments with the 2B model, we employed the DDIM sampler, which is also used in the 2B version of CogVideoX. We compared the performance differences between the 2B and 5B models within our method, specifically focusing on the single-task segmentation model, as presented in Tab. 3. The increased number of parameters in the 5B model led to significant performance improvements. This larger model is better at incorporating segmentation guidance and can generate more realistic and higher-quality videos.

4. More Visualization Results

Fig. 4 presents additional videos generated by our model, showcasing its capabilities in diverse scenes. These videos include subjects such as portraits of people, animals in motion, and dynamic landscapes. In these examples, we demonstrate high-quality generation results, producing videos that are consistent, smooth, and aesthetically pleasing. The model effectively maintains temporal coherence and visual fidelity throughout the frames, capturing fine details and natural movements inherent in different scenes.

Fig. 5 provides more videos and their segmentation results, where the multitask model performs inference in segmentation mode. The guidance provided by segmentation improves the consistency and quality of video generation and produces the corresponding high-quality segmentation masks. Our segmentation is very detailed in both simple and complex, densely populated scenes. Fig. 6 displays videos and their corresponding depth maps, with inference performed in depth estimation mode by the multitask model. Depth estimation also improves consistency and quality while generating accurate depth maps.

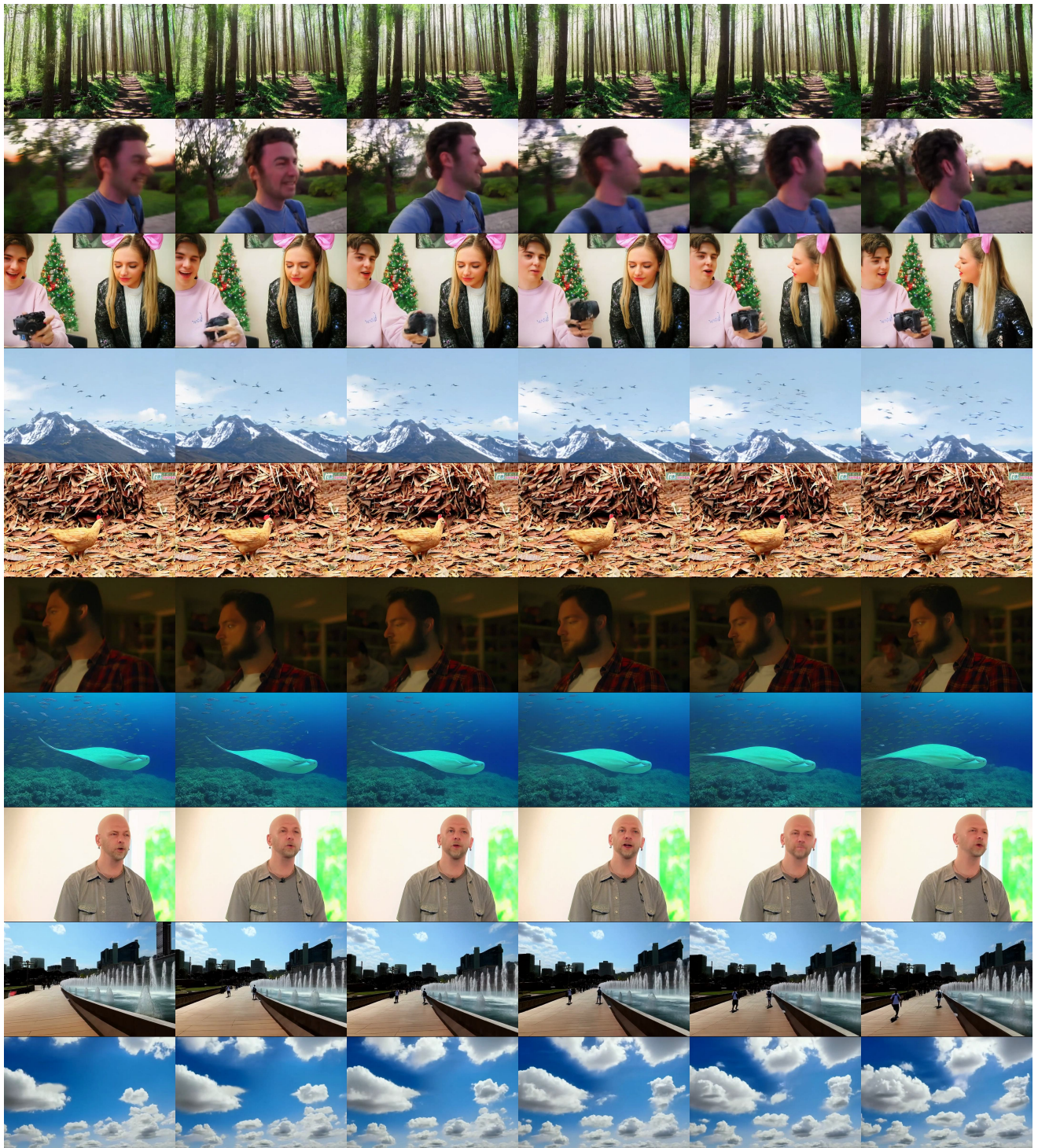


Figure 4. **More examples on video generation quality.** Six frames are evenly sampled from the entire 49-frame video generated. Having the sample in both portraits, animals, and landscapes.

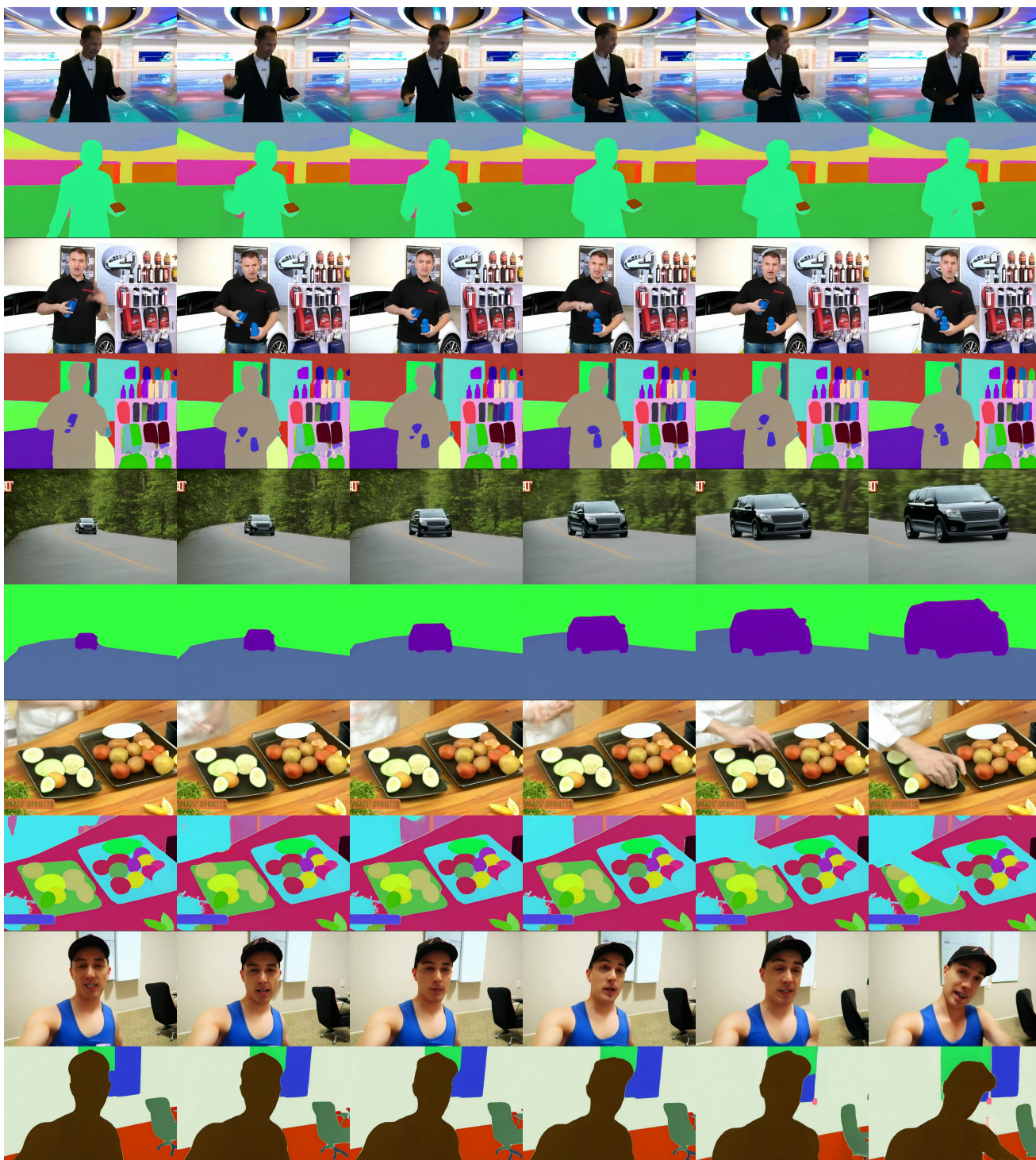


Figure 5. **More examples on video generation with segmentation.** Each pair consists of two rows: the first row displays the generated video, while the second row shows the corresponding segmentation. Six frames are evenly sampled from the total of 49 frames in the generated video. Our results demonstrate excellent performance in both simple scenes and those with densely packed objects, highlighting our ability to effectively segment dense entities.

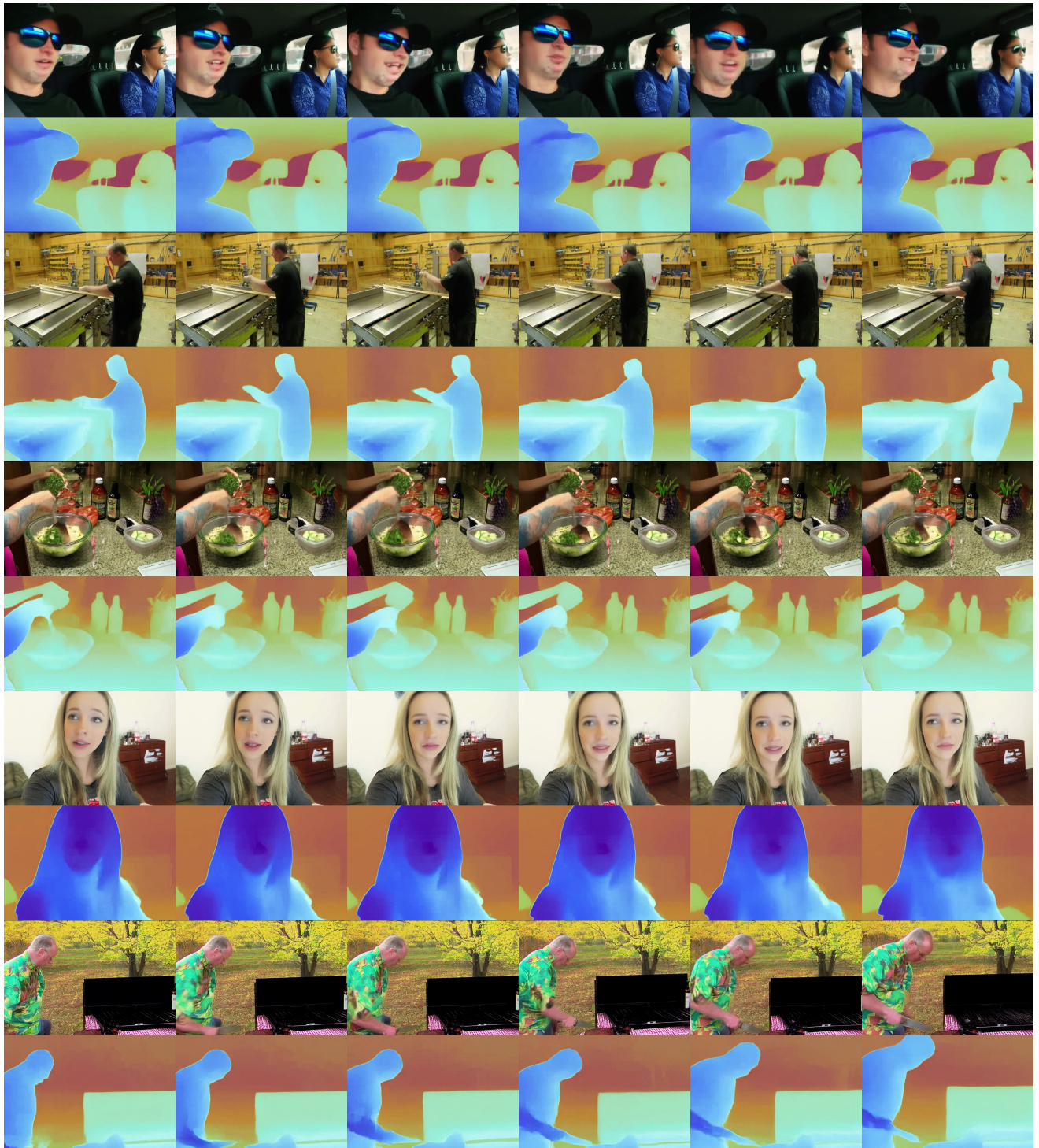


Figure 6. **More examples on video generation with depth estimation.** Each pair has two rows: the first row displaying the generated video, and the second row presenting the corresponding depth estimation map. Six frames are evenly sampled from the total of 49 frames in the generated video.