A Details on the CrossTask Dataset

In this appendix, we provide detailed statistics for the 18 primary tasks in the CrossTask dataset (Zhukov et al. 2019). This dataset contains 2750 videos, each demonstrating one of its 18 procedural tasks; e.g., "Make a latte", "Change a tire", or "Make pancakes". The average video length is about 5 minutes, with a total of 212 hours of recorded videos. These tasks are fairly complex and each task takes on average 7.4 actions to complete. Simpler tasks like "Jack up a car" take about 3 actions to finish, while more complicated ones like "Change a tire" can take as many as 11 actions.

Table 6 shows a complete list of the 18 primary tasks, their average video lengths, sizes of the action spaces, average step lengths, and percentage of "null action" steps. The action space of a task is the set of all candidate actions that can be taken when performing the task, although some videos skip certain actions and do not cover the complete action space. The average step length measures the average number of steps taken in the actual video demonstrations to complete a task. Due to certain actions being skipped in some videos and others being repeated, the average step length of a task is not always equal to its action space size. "Null actions" are used to refer to the video segments that do not have actual actions happening, such as the introduction part of the video or transitioning scenes from one action to another. In Figure 2, we also illustrate the Action Dynamics Task Graphs for several tasks from the CrossTask dataset.

For reproducibility and fair comparison with existing methods, we leverage the pre-computed video features provided along with the CrossTask dataset. For each one-second segment of the video, a 3200-dimensional feature vector is provided and contains a concatenation of 1024-D RGB I3D features (Carreira and Zisserman 2017), 2048-D Resnet-152 features (He et al. 2016), and 128-D audio VGG features (Hershey et al. 2017).

B Details for the CrossTask Baseline

The CrossTask baseline refers to the solution proposed in the same work (Zhukov et al. 2019) along with the CrossTask dataset. The CrossTask baseline is a weakly-supervised approach for learning from instructional videos. It does not rely on the strong supervisions via temporal annotations of the action boundaries, but instead only use the temporal constraints generated from the instructional narrations and an ordered list of the action steps. The CrossTask approach is built upon the idea that the learning model should share certain components (e.g., verbs or nouns) while learning different steps across multiple tasks. For example, the action "pour egg" should be trained jointly with other tasks involving the components "pour" or "egg". Following this idea, CrossTask proposes to use component models to represent each step as its constituent components instead of as a monolithic entity. The step assignment objective in CrossTask essentially corresponds to our task tracking module, yet CrossTask does not support next action recommendation or plan generation.

Since the original CrossTask approach is a weakly-supervised learning method, for fair comparisons, in our experiments we also consider a supervised-learning variant of CrossTask that adopts the same linear classifier as (Zhukov et al. 2019), but further uses the annotated action segmentation boundaries for training.

C Details for the Neural Task Graphs Baseline

Similar to ours, the NTG approach is a modularized method that uses (a simplified variant of) the conjugate task graphs as intermediate representations. NTG focuses on generalizing to unseen tasks from a single video demonstration in the same domain. It uses the CTG representations to explicitly modularize the video demonstration and the derived policy, so as to incorporate the compositional structure of the tasks into the NTG model. Specifically, NTG consists of a generator that builds a conjugate task graph from video demonstrations, and an execution engine that uses the learned tasks graphs to perform task tracking. In particular, the NTG generator itself can be decomposed into two parts: a demo interpreter that is used to obtain a single action path traversing the CTG by observing the action sequence in the video demonstration, and a graph completion network that adds the edges that are not observed in the single demonstration to capture the potential interchangeability of the action ordering. The NTG execution engine also consists of two parts: A node localizer that tries to localize the current action node in the CTG based on the visual observation (i.e., task tracking), and an edge classifier that checks the precondition of each possible outgoing edge from the localized node to decide the next action (i.e., next action recommendation). Since the edge classifier in NTG relies on visual observations as input, in the absence of an interactive environment, it cannot generate a full plan in an autoregressive way as we do.

D Plan Visualization

In Figure 5, we visualize the planned action sequences generated by ADTG on a few testing videos, and compare them with the ground-truth (GT) plans. In the first example, ADTG successfully generates the correct sequence of actions for the task "Make jello shots". In the second example, the task tracking module of ADTG fails to recognize the first step ("pour jello powder") of the video and misclassifies it as "stir mixture". Since ADTG generates plans by recursively invoking the next action recommendation module, it is not able to correct such a mistake and hence diverges from the ground-truth action sequence afterward. In the last example (on the task "Make pancakes"), even though the action sequence planned by ADTG does not exactly match the ground-truth plan, it still forms a semantically reasonable plan to complete the task. This is because the ADTG generated plan simply switches the order of the actions "pour milk" and "whisk mixture" compared to the ground-truth and removes the repeated "whisk mixture" steps, which makes sense in the given task. This also suggests that we might need better ways to evaluate plans in such datasets that do not have an interactive environment and we leave this to future work.

Table 6: Statistics of the CrossTask dataset.

Task	Number of videos	Action space size	Average step length	Percentage of null action
Make Jello Shots	182	6	7.90	72%
Build Simple Floating Shelves	153	5	5.54	58%
Make Taco Salad	170	8	6.34	79%
Grill Steak	228	11	8.54	75%
Make Kimchi Fried Rice	120	6	8.66	70%
Make Meringue	154	6	6.72	67%
Make a Latte	157	6	5.06	71%
Make Bread and Butter Pickles	106	11	6.44	75%
Make Lemonade	131	8	8.28	69%
Make French Toast	252	10	9.10	68%
Jack Up a Car	89	3	3.38	81%
Make Kerala Fish Curry	149	7	10.02	69%
Make Banana Ice Cream	170	5	4.52	80%
Add Oil to Your Car	137	8	8.04	85%
Change a Tire	99	11	9.84	62%
Make Irish Coffee	185	5	4.94	74%
Make French Strawberry Cake	86	9	11.56	63%
Make Pancakes	182	8	10.54	70%
Average	153	7.4	7.84	72%

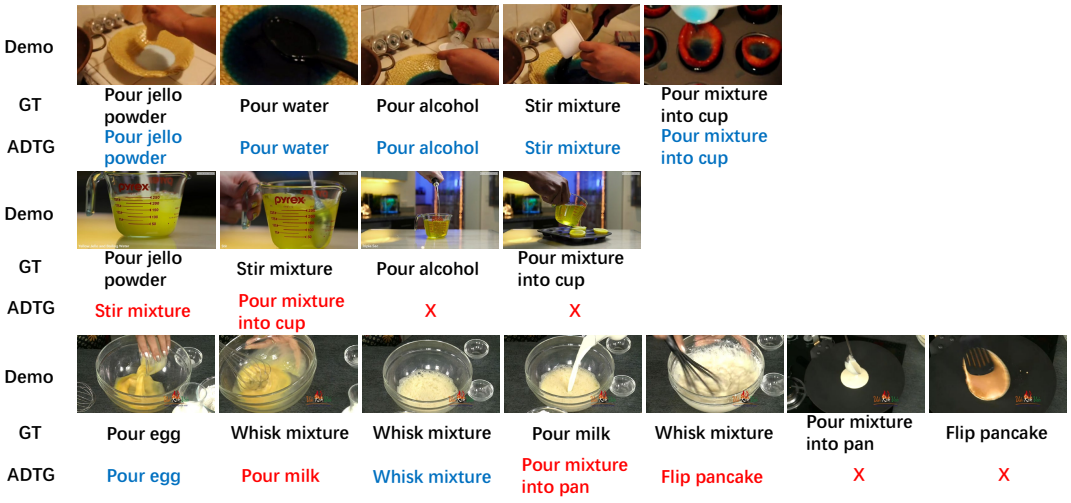

Figure 5: Visualization of ground-truth plans (GT) vs. ADTG generated plans.