Learning to Segment Actions from Observation and Narration

Daniel Fried[‡] **Jean-Baptiste Alayrac**[†] **Phil Blunsom**[†]
Chris Dyer[†] **Stephen Clark**[†] **Aida Nematzadeh**[†]
[†]DeepMind, London, UK
[‡]Computer Science Division, UC Berkeley
dfried@berkeley.edu, {jalayrac,pblunsom,cdyer,clarkstephen,nematzadeh}@google.com

Abstract

We apply a generative segmental model of task structure, guided by narration, to action segmentation in video. We focus on unsupervised and weakly-supervised settings where no action labels are known during training. Despite its simplicity, our model performs competitively with previous work on a dataset of naturalistic instructional videos. Our model allows us to vary the sources of supervision used in training, and we find that both task structure and narrative language provide large benefits in segmentation quality.

1 Learning to Segment Actions

Finding boundaries in a continuous stream is a crucial process for human cognition (Martin and Tversky, 2003; Zacks and Swallow, 2007; Levine et al., 2019; Ünal et al., 2019). To understand and remember what happens in the world around us, we need to recognize the action boundaries as they unfold and also distinguish the important actions from the insignificant ones. This process, referred to as *temporal action segmentation*, is also an important first step in systems that ground natural language in videos (Hendricks et al., 2017). These systems must identify which frames in a video depict actions – which amounts to distinguishing these frames from background ones – and identify which actions (*e.g.*, boiling potatoes) each frame depicts. Despite recent advances (Miech et al., 2019; Sun et al., 2019), unsupervised action segmentation in videos remains a challenge.

The recent availability of large datasets of naturalistic instructional videos provides an opportunity for modeling of action segmentation in a rich task context (Yu et al., 2014; Zhou et al., 2018; Zhukov et al., 2019; Miech et al., 2019; Tang et al., 2019); in these videos, a person teaches a specific high-level *task* (*e.g.*, making croquettes) while describing the lower-level *steps* involved in that task (*e.g.*, boiling potatoes). However, the real-world nature of these datasets introduces many challenges. For example, more than 70% of the frames in one of the YouTube instructional video datasets, CrossTask (Zhukov et al., 2019), consist of *background* regions (*e.g.*, the video presenter is thanking their viewers), which do not correspond to any of the steps for the video's task.

These datasets are interesting because they provide (1) narrative language that roughly corresponds to the activities demonstrated in the videos and (2) structured task scripts that define a strong signal of the order in which steps in a task are typically performed. As a result, these datasets provide an opportunity to study the extent to which task structure and language can guide action segmentation. Interestingly, young children can segment actions without any explicit supervision (Baldwin et al., 2001; Sharon and Wynn, 1998), by tapping into similar cues – action regularities and language descriptions (*e.g.*, Levine et al., 2019).

While previous work mostly focuses on building action segmentation models that perform well on a few metrics (Richard et al., 2018; Zhukov et al., 2019), we aim to provide insight into how various modeling choices impact action segmentation. How much do unsupervised models improve when given implicit supervision from task structure and language, and which types of supervision help most? Are discriminative or generative models better suited for the task? Does explicit structure modeling improve the quality of segmentation? To answer these questions, we compare two existing models with a generative hidden semi-Markov model, varying the degree of supervision.

Work begun while DF was interning at DeepMind. Code is available at https://github.com/dpfried/action-segmentation.

On a challenging and naturalistic dataset of instructional videos (Zhukov et al., 2019), we find that our model and models from past work both benefit substantially from the weak supervision provided by task structure and narrative language, even on top of rich features from state-of-the-art pretrained action and object classifiers. Our analysis also shows that: (1) Generative models tend to do better than discriminative models of the same or similar model class at learning the full range of step types, which benefits action segmentation; (2) Task structure affords strong, feature-agnostic baselines that are difficult for existing systems to surpass; (3) Reporting multiple metrics is necessary to understand each model's effectiveness for action segmentation; we can devise feature-agnostic baselines that perform well on single metrics despite producing low-quality action segments.

2 Related Work

Typical methods (Rohrbach et al., 2012; Singh et al., 2016; Xu et al., 2017; Zhao et al., 2017; Lea et al., 2017; Yeung et al., 2018; Farha and Gall, 2019) for temporal action segmentation consist of assigning action classes to intervals of videos and rely on manually-annotated supervision. Such annotation is difficult to obtain at scale. As a result, recent work has focused on training such models with less supervision: one line of work assumes that only the order of actions happening in the video is given and use this weak supervision to perform action segmentation (Bojanowski et al., 2014; Huang et al., 2016; Kuehne et al., 2017; Richard et al., 2017; Ding and Xu, 2018; Chang et al., 2019). Other approaches weaken this supervision and use only the set of actions that occur in each video (Richard et al., 2018), or are fully unsupervised (Sener and Yao, 2018; Kukleva et al., 2019).

Instructional videos have gained interest over the past few years (Yu et al., 2014; Sener et al., 2015; Malmaud et al., 2015; Alayrac et al., 2016; Zhukov et al., 2019) since they enable weakly-supervised modeling: previous work most similar to ours consists of models that localize actions in narrated videos with minimal supervision (Alayrac et al., 2016; Sener et al., 2015; Elhamifar and Naing, 2019; Zhukov et al., 2019).

We present a generative model of action segmentation that incorporates duration modeling, narration and ordering constraints, and can be trained in all of the above supervision conditions by maximizing the likelihood of the data; while these past works have had these individual components, they have not yet all been combined.

3 The CrossTask Dataset

We use the recent CrossTask dataset (Zhukov et al., 2019) of instructional videos. To our knowledge, CrossTask is the only available dataset that has tasks from more than one domain, includes background regions, provides step annotations and naturalistic language. Other datasets lack one of these; *e.g.*they focus on one domain (Kuehne et al., 2014) or do not have natural language (Tang et al., 2019) or step annotations (Miech et al., 2019). An example instance from the dataset is shown in Figure 1, and we describe each aspect below.

Tasks Each video comes from a *task*, e.g. *make a latte*, with tasks taken from the titles of selected WikiHow articles, and videos curated from YouTube search results for the task name. We focus on the *primary* section of the dataset, containing 2,700 videos from 18 different tasks.

Steps and canonical order Each task has a set of *steps*: lower-level action *step* types, *e.g.*, *steam milk* and *pour milk*, which are typically completed when performing the task. Step names consist of a few words, typically naming an action and an object it is applied to. The dataset also provides a *canonical step order* for each task: an ordering, like a script (Schank and Abelson, 1977; Chambers and Jurafsky, 2008), in which a task's steps are typically performed. For each task, the set of step types and their canonical order were hand-constructed by the dataset creators based on section headers in the task's WikiHow article.

Annotations Each video in the primary section of the dataset is annotated with labeled temporal segments identifying where steps occur. (In the weak supervision setting, these step segment labels are used only in evaluation, and never in training.) A given step for a task can occur multiple times, or not at all, in any of the task's videos. Steps in a video also *need not* occur in the task's canonical ordering (although in practice our results show that this ordering is a helpful inductive bias for learning). Most of the frames in videos (72% over the entire corpus) are *background* – not contained in any step segment.

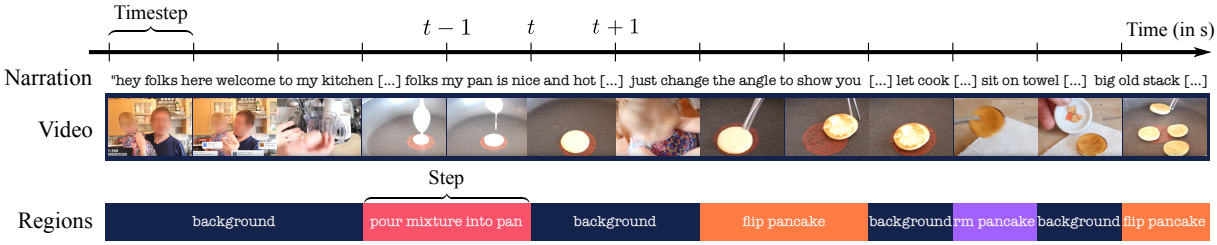


Figure 1: An example video instance from the CrossTask dataset (Sec. 3). The video depicts a task, *make pancakes*, and is annotated with *region* segments, which can be either action steps (*e.g.*, *pour mixture into pan*) or background regions. Videos also are temporally-aligned with transcribed narration. We learn to segment the video into these regions and label them with the action steps (or background), without access to region annotations during training.

Narration Videos also have narration text (transcribed by YouTube's automatic speech recognition system) which typically consists of a mix of the task demonstrator describing their actions and talking about unrelated topics. Although narration is temporally aligned with the video, and steps (*e.g.*, *pour milk*) are sometimes mentioned, these mentions often do not occur at the same time as the step they describe (*e.g.*, "let the milk cool before *pouring it*"). Zhukov et al. (2019) guide weakly-supervised training using the narration by defining a set of *narration constraints* for each video, which identify where in the video steps are likely to occur, using similarity between the step names and temporally-aligned narration (see Sec. 6.1).

4 Model

Our generative model of the video features and labeled task segments is a first-order semi-Markov model. We use a semi-Markov model for the action segmentation task because it explicitly models temporal regions of the video, their duration, their probable ordering, and their features.[1] It can be trained in an unsupervised way, without labeled regions, to maximize the likelihood of the features.

Timesteps Our atomic unit is a one-second region of the video, which we refer to as a *timestep*. A video with T timesteps has feature vectors $x_{1:T}$. The features x_t at timestep t are derived from the video, its narration, or both, and in our work (and past work on the dataset) are produced by pre-trained neural models which summarize some non-local information in the region containing each timestep, which we describe in Sec. 6.3.

Regions Our model segments a video with T timesteps into a sequence of regions, each of which consists of a consecutive number of timesteps (the region's *duration*). The number of regions K in a video and the duration d_k of each region can vary; the only constraint is that the sum of the durations equals the video length: $\sum_{k=1}^{K} d_k = T$. Each region has a label r_k, which is either one of the task's step labels (*e.g.*, *pour milk*) or a special label BKG indicating the region is background. In our most general, unconstrained model, a given task step can occur multiple times (or not at all) as a region label in any video for the task, allowing step repetitions, dropping, and reordering.

Structure We define a first-order Markov (bigram) model over these region labels:

$$P(r_{1:K}) = P(r_1) \prod_{k=2}^{K} P(r_k \mid r_{k-1}) \qquad (1)$$

with tabular conditional probabilities. While region labels are part of the dataset, they are primarily used for evaluation: we seek models that can be trained in the unsupervised and weakly-supervised conditions where labels are unavailable. This model structure, while simple, affords a dynamic program allowing efficient enumeration over both all possible segmentations of the video into regions and assignments of labels to the regions, allowing unsupervised training (Sec. 4.1).

Duration Our model, following past work (Richard et al., 2018), parameterizes region durations using Poisson distributions, where each label type r has its own mean duration λ_r: $d_k \sim \text{Poisson}(\lambda_{r_k})$. These durations are constrained so that they partition the video: *e.g.*, region r_2 begins at timestep d_1 (after region r_1), and the final region r_K ends at the final timestep T.

Timestep labels The region labels $r_{1:K}$ (step, or background) and region durations $d_{1:K}$ together give a sequence of *timestep labels* $l_{1:T}$ for all

[1] Semi-Markov models are also shown to be successful in the similar domain of speech recognition (*e.g.*, Pylkkonen and Kurimo, 2004).

timesteps, where a timestep's label is equal to the label for the region it is contained in.

Feature distribution Our model's feature distribution $p(x_t|l_t)$ is a class-conditioned multivariate Gaussian distribution: $x_t \sim \text{Normal}(\mu_{l_t}, \Sigma)$, where l_t is the step label at timestep t. (We note that the assignment of labels to steps is latent and unobserved during unsupervised and weakly-supervised training.) We use a separate learned mean μ_l for each label type l, both steps and background. Labels are atomic and task-specific, *e.g.*, the step type *pour milk* when it occurs in the task *make a latte* does not share parameters with the step *add milk* when it occurs in the task *make pancakes*.[2] We use a diagonal covariance matrix Σ which is fixed to the empirical covariance of each feature dimension.[3]

4.1 Training

In the unsupervised setting, labels l are unavailable at training (used only in evaluation). We describe training in this setting, as well as two supervised training methods which we use to analyze properties of the dataset and compare model classes.

Unsupervised We train the generative model as a *hidden* semi-Markov model (HSMM). We optimize the model's parameters to maximize the log marginal likelihood of the features for all video instance features $x^{(i)}$ in the training set:

$$\mathcal{ML} = \sum_{i}^{N} \log P(x^{(i)}_{1:T_i}) \tag{2}$$

Applying the semi-Markov forward algorithm (Murphy, 2002; Yu, 2010) allows us to marginalize over all possible sequences of step labels to compute the log marginal likelihood for each video as a function of the model parameters, which we optimize directly using backpropagation and mini-batched gradient descent with the Adam (Kingma and Ba, 2015) optimizer.[4] See Appendix A for optimization details.

Generative supervised Here the labels l are observed; we train the model as a generative semi-Markov model (SMM) to maximize the log joint likelihood:

$$\mathcal{JL} = \sum_{i}^{N} \log P(l^{(i)}_{1:T_i}, x^{(i)}_{1:T_i}) \tag{3}$$

We maximize this likelihood over the entire training set using the closed form solution given the dataset's sufficient statistics (per-step feature means, average durations, and step transition frequencies).

Discriminative supervised To train the SMM model discriminatively in the supervised setting, we use gradient descent to maximize the log conditional likelihood:

$$\mathcal{CL} = \sum_{i}^{N} \log P(l^{(i)}_{1:T} \mid x^{(i)}_{1:T}) \tag{4}$$

	Richard et al. (2018)	Zhukov et al. (2019)	Ours
step reordering	✓		✓
step repetitions	✓		✓
step duration	✓		✓
language		✓	✓
generative model	✓		✓

Table 1: Characteristics of each model we compare.

5 Benchmarks

We identify five modeling choices made in recent work: imposing a fixed ordering on steps (not allowing step reordering); allowing for steps to repeat in a video; modeling the duration of steps; using the language (narrations) associated with the video; and using a discriminative/generative model. We picked the recent models of Zhukov et al. (2019) and Richard et al. (2018) since they have non-overlapping strengths (see Table 1).

OrderedDiscrim (Zhukov et al., 2019). This work uses a discriminative classifier which gives a probability distribution over labels at each timestep: $p(l_t \mid x_t)$. Inference finds an assignment of steps to timesteps that maximizes $\sum_t \log p(l_t|x_t)$ subject to the constraints that: all steps are predicted exactly once; steps occur in the fixed canonical ordering defined for the task; one background region occurs between each step. Unsupervised training of the model alternates between inferring labels using the dynamic program, and updating the classifier to maximize the probability of these inferred labels.[5]

[2] We experimented with sharing steps, or step components, across tasks in initial experiments, but found that it was helpful to have task-specific structural probabilities.

[3] We found that using a shared diagonal covariance matrix outperformed using full or unshared covariance matrices.

[4] This is the same as mini-batched Expectation Maximization using gradient descent on the M-objective (Eisner, 2016).

[5] To allow the model to predict step regions with duration longer than a single timestep, we modify this classifier to also predict a background class, and incorporate the scores of the background class into the dynamic program.

ACTIONSETS (Richard et al., 2018). This work uses a generative model which has structure similar to ours, but uses dataset statistics (*e.g.*, average video length and number of steps) to learn the structure distributions, rather than setting parameters to maximize the likelihood of the data. As in our model, region durations are modeled using a class-conditional Poisson distribution. The feature distribution is modeled using Bayesian inversion of a discriminative classifier (a multi-layer perceptron) with an estimated label prior. The structural parameters of the model (durations and class priors) are estimated using the length of each video, and the number of possible step types. As originally presented, this model depends on knowing which steps occur in a video at training time; for fair comparison, we adapt it to the same supervision conditions of Zhukov et al. (2019) by enforcing the canonical step ordering for the task during both training and evaluation.

6 Experimental Setting

We compare models on the CrossTask dataset across supervision conditions. We primarily evaluate the models on action segmentation (Sec. 1). Past work on the dataset (Zhukov et al., 2019) has focused on a *step recognition task*, where models identify individual timesteps in videos that correspond to possible steps; for comparison, we also report performance for all models on this task.

6.1 Supervision Conditions

In all settings, the task for a given video is known (and hence the possible steps), but the settings vary in the availability of other sources of supervision: step labels for each timestep in a video, and constraints from language and step ordering. Models are trained on a training set and evaluated on a separate held-out testing set, consisting of different videos (from the same tasks).

Supervised Labels for all timesteps $l_{1:T}$ are provided for all videos in the training set.

Fully unsupervised No labels for timesteps are available during training. The only supervision is the number of possible step types for each task (and, as in all settings, which task each video is from). In evaluation, the task for a given video (and hence the possible steps, but not their ordering) are known. We follow past work in this setting (Sener et al., 2015; Sener and Yao, 2018) by finding a mapping from model states to region labels that maximizes label accuracy, averaged across all videos in the task. See Appendix C for details.

Weakly supervised No labels for timesteps are available, but two supervision types are used in the form of constraints (Zhukov et al., 2019):

(1) *Step ordering constraints*: Step regions are constrained to occur in the canonical step ordering (see Sec. 3) for the task, but steps may be separated by background. We constrain the structure prior distributions $p(r_1)$ and transition distributions $p(r_{k+1}|r_k)$ of the HSMM to enforce this ordering. For $p(r_1)$, we only allow non-zero probability for the background region, BKG, and for the first step in the task's ordering. $p(r_k \mid r_{k-1})$ constrains each step type to only transition to the next step in the constrained ordering, or to BKG.[6] As step ordering constraints change the parameters of the model, when we use them we enforce them during both training and testing. While this obviates most of the learned structure of the HSMM, the duration model (as well as the feature model) is still learned.

(2) *Narration constraints*: These give regions in the video where each step type is likely to occur. Zhukov et al. (2019) obtained these using similarities between word vectors for the transcribed narration and the words in the step labels, and a dynamic program to produce constraint regions that maximize these similarities, subject to the step ordering matching the canonical task ordering. See Zhukov et al. for details. We enforce these constraints in the HSMM by penalizing the feature distributions to prevent any step labels that occur outside of one of the allowed constraint regions for that step. Following Zhukov et al., we only use these narration constraints during training.[7]

6.2 Evaluation

We use three metrics from past work, outlined here and described in more detail in Appendix D. To evaluate action segmentation, we use two varieties of the standard label accuracy metric (Sener and Yao, 2018; Richard et al., 2018): **all label accuracy**, which is computed on all timesteps, including background and non-background, as well as

[6] To enforce ordering when steps are separated by BKG, we annotate BKG labels with the preceeding step type (but all BKG labels for a task share feature and duration parameters, and are merged for evaluation).

[7] We also experiment with using features derived from transcribed narration in Appendix G.

step label accuracy: accuracy only for timesteps that occur in a non-background region (according to the ground-truth annotations). Since these two accuracy metrics are defined on individual frames, they penalize models if they don't capture the full temporal extent of actions in their predicted segmentations. Our third metric is **step recall**, used by past work on the CrossTask dataset (Zhukov et al., 2019) to measure *step recognition* (defined in Sec. 6). This metric evaluates the fraction of step types which are correctly identified by a model when it is allowed to predict only one frame per step type, per video. A high step recall indicates a model can accurately identify at least one representative frame of each action type in a video.

We also report three other statistics to analyze the predicted segmentations: (1) Sequence similarity: the similarity of the sequence of region labels predicted in the video to the groundtruth, using inverse Levenshtein distance normalized to be between 0 and 100. See Appendix D for more details. (2) Predicted background percentage: the percentage of timesteps for which the model predicts the background label. Models with a higher percentage than the ground truth background percentage (72%) are overpredicting background. (3) Number of segments: the number of step segments predicted in a video. Values higher than the ground truth average (7.7) indicate overly-fragmented steps. Sequence similarity and number of segments are particularly relevant for measuring the effects of structure, as they do not factor over individual timesteps (as do the all label and step label accuracies and step recall).

We average values across the 18 tasks in the evaluation set (following Zhukov et al. 2019).

6.3 Features

For our features $x_{1:T}$, we use the same base features as Zhukov et al. (2019), which are produced by convolutional networks pre-trained on separate activity, object, and audio classification datasets. See Appendix B for details. In our generative models, we apply PCA (following Kuehne et al., 2014 and Richard et al., 2018) to project features to 300 dimensions and decorrelate dimensions (see Appendix B for details).[8]

[8]This reduces the number of parameters that need to be learned in the emission distributions, both by reducing the dimensionality and allowing a diagonal covariance matrix. In early experiments we found PCA improved performance.

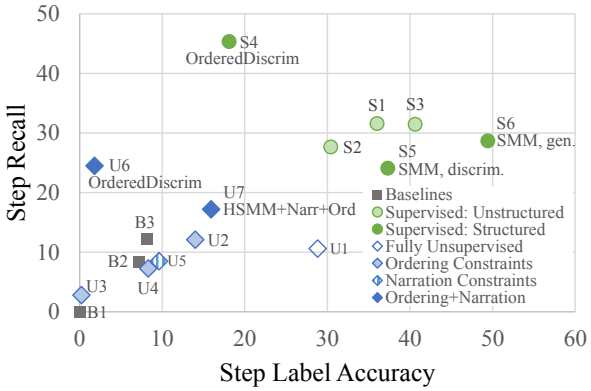


Figure 2: Baseline and model performance on two key metrics: step label accuracy and step recall. Points are colored according to their supervision type, and labeled with their row number from Table 2. We also label particular important models.

7 Results

We first define several baselines based on dataset statistics (Sec. 7.1), which we will find to be strong in comparison to past work. We then analyze each aspect of our proposed model on the dataset in a supervised training setting (Sec. 7.2), removing some error sources of unsupervised learning and evaluating whether a given model fits the dataset (Liang and Klein, 2008). Finally, we move to our main setting, the weakly-supervised setting of past work, incrementally adding step ordering and narration constraints (see Sec. 6.1) to evaluate the degree to which each helps (Sec. 7.3).

Results are given in Table 2 for models trained on the CrossTask training set of primary tasks, and evaluated on the held-out validation set. We will describe and analyze each set of results in turn. See Figure 2 for a plot of models' performance on two key metrics, and Appendix I for example predictions.

7.1 Dataset Statistic Baselines

Table 2 (top block) shows baselines that *do not* use video (or narration) features, but predict steps according to overall statistics of the training data. These demonstrate characteristics of the data, and the importance of using multiple metrics.

Predict background (B1) Since most timesteps are background, a model that predicts background *everywhere* can obtain high overall label accuracy, showing the importance of also using step label accuracy as a metric for action segmentation.

Sample from the training distribution (B2) For each timestep in each video, we sample a label

#	Model	All Label Accuracy	Step Label Accuracy	Step Recall	Sequence Similarity	Predicted Bkg. %	Num. Segments.
Dataset Statistic Baselines (Sec. 7.1)							
GT	Ground truth	100.0	100.0	100.0	100.0	71.9	7.7
B1	Predict background	71.9	0.0	0.0	9.0	100.0	0.0
B2	Sample from train distribution	54.6	7.2	8.3	12.8	72.4	69.5
B3	Ordered uniform	55.6	8.1	12.2	55.0	73.0	7.4
Supervised (Sec. 7.2)							
	Unstructured						
S1	Discriminative linear	71.0	36.0	31.6	30.7	73.3	27.1
S2	Discriminative MLP	**75.9**	30.4	27.7	41.1	82.8	13.0
S3	Gaussian mixture	69.4	40.6	31.5	33.3	68.9	23.9
	Structured						
S4	OrderedDiscrim	75.2	18.1	**45.4**	54.4	90.7	7.4
S5	SMM, discriminative	66.0	37.3	24.1	50.5	65.9	8.5
S6	SMM, generative	60.5	**49.4**	28.7	46.6	52.4	10.6
Un- and Weakly-Supervised (Sec. 7.3)							
	Fully Unsupervised						
U1	HSMM (with opt. acc. assignment)	31.8	28.8	10.6	31.0	31.1	15.4
	Ordering Supervision						
U2	ActionSets	40.8	14.0	12.1	55.0	49.8	7.4
U3	OrderedDiscrim (without Narr.)	69.5	0.2	2.8	55.0	97.2	7.4
U4	HSMM + Ord	55.5	8.3	7.3	55.0	70.6	7.4
	Narration Supervision						
U5	HSMM + Narr	65.7	9.6	8.5	35.1	84.6	4.5
	Ordering + Narration Supervision						
U6	OrderedDiscrim	**71.0**	1.8	**24.5**	55.0	97.2	7.4
U7	HSMM + Narr + Ord	61.2	**15.9**	17.2	55.0	73.7	7.4

Table 2: Model comparison on the CrossTask validation data. We evaluate primarily using all label accuracy and step label accuracy to evaluate action segmentation, and step recall to evaluate step recognition.

from the empirical distribution of step and background label frequencies for the video's task in the training data.

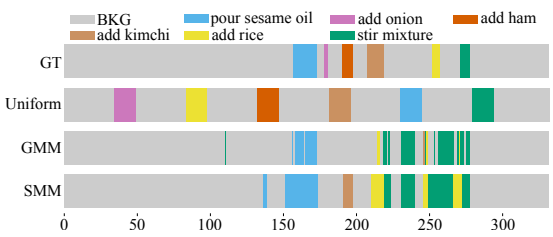
Ordered uniform (B3) For each video, we predict step regions in the canonical step order, separated by background regions. The length of each region is set so that all step regions in a video have equal duration, and the percentage of background timesteps is equal to the corpus average. See *Uniform* in Figure 3a for sample predictions.

Sampling each timestep label independently from the task distribution (row B2), and using a uniform step assignment in the task's canonical ordering with background (B3) both obtain similar step label accuracy, but the ordered uniform baseline improves substantially on the step recall metric, indicating that step ordering is a useful inductive bias for *step recognition*.

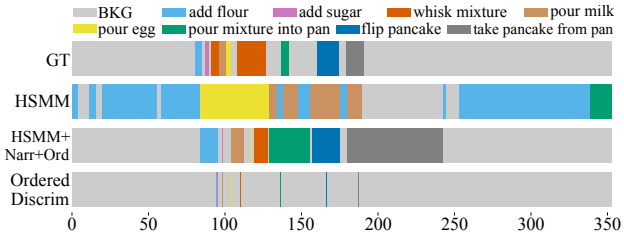
7.2 Full Supervision

Models in the **unstructured** block of Table 2 are classification models applied independently to all timesteps, allowing us to compare the performance of the feature models used as components in our structured models. We find that a Gaussian mixture model (row S3), which is used as the feature model in the HSMM, obtains comparable step recall and substantially higher step label accuracy than a discriminative linear classifer (row S1) similar to the one used in Zhukov et al. (2019), which is partially explained by the discriminative classifier overpredicting the background class (comparing Predicted Background % for those two rows). Using a higher capacity discriminative classifier, a neural net with a single hidden layer (MLP), improves performance over the linear model on several metrics (row S2); however, the MLP still overpredicts background, substantially underperforming the Gaussian mixture on the step label accuracy metric.

In the **structured** block of Table 2, we compare the full models which use step constraints (Zhukov et al., 2019) or learned transition distributions (the SMM) to model task structure. The structured models learn (or in the case of Zhukov et al., enforce) orderings over the steps, which greatly improve their sequence similarity scores when compared to the unstructured models, and decrease step fragmentation (as measured by num. segments). Figure 3a shows predictions for a typical video, demonstrat-



(a) Step segmentations in the full supervision condition for a video from the *make kimchi fried rice* task, comparing the ground truth (GT), ordered uniform baseline (Uniform), and predictions from the Gaussian mixture (GMM) and semi-Markov (SMM) models.



(b) Step segmentations in the no- or weak-supervision conditions for a video from the *make pancakes* task, comparing the ground truth (GT) to predictions from our model without (HSMM) and with constraint supervision (HSMM+Narr+Ord) and from Zhukov et al. (2019) (ORDEREDDISCRIM).

Figure 3: Step segmentation visualizations for two sample videos in supervised (left) and unsupervised (right) conditions. The x-axes show timesteps, in seconds. See Appendix I for more visualizations.

ing this decreased fragmentation.[9]

We see two trends in the supervised results:

(1) Generative models obtain substantially higher step label accuracy than discriminative models of the same or similar class. This is likely due to the fact that the generative models directly parameterize the step distribution. (See Appendix E.)

(2) Structured sequence modeling naturally improves performance on sequence-level metrics (sequence similarity and number of segments predicted) over the unstructured models. However, none of the learned structured models improve on the strong ordered uniform baseline (B3) which just predicts the canonical ordering of a task's steps (interspersed with background regions). This will motivate using this canonical ordering as a constraint in unsupervised learning.

Overall, the SMM models obtain strong action segmentation performance (high step label accuracy without fragmenting segments or overpredicting background).

7.3 No or Weak Supervision

Here models are trained without supervision for the labels $l_{1:T}$. We compare models trained without any constraints, to those that use constraints from step ordering and narration, in the **Un- and Weakly Supervised** block of Table 2. Example outputs are shown in Appendix I.

Our generative HSMM model affords training without any constraints (row U1). This model has high step label accuracy (compared to the other unsupervised models) but low all label accuracy, and similar scores for both metrics. This hints, and other metrics confirm, that the model is not adequately distinguishing steps from background: the percentage of predicted background is very low (31%) compared to the ground truth (72%, row GT). See *HSMM* in Figure 3b for predictions for a typical video. These results are attributable to features within a given video (even across step types) being more similar than features of the same step type in different videos (see Appendix H for feature visualizations). The induced latent model states typically capture this inter-video diversity, rather than distinguishing steps across tasks.

We next add in constraints from the canonical step **ordering**, which our supervised results showed to be a strong inductive bias. Unlike in the fully unsupervised setting, the HSMM model with ordering (HSMM+Ord, row U4) *learns* to distinguish steps from background when constrained to predict each step region once in a video, with predicted background timesteps (70.6%) close to the ground-truth (72%). However, performance of this model is still very low on the task metrics – comparable to or underperforming the ordered uniform baseline with background (row B3) on all metrics.

This constrained step ordering setting also allows us to apply ACTIONSETS (Richard et al., 2018) and ORDEREDDISCRIM (Zhukov et al., 2019). ACTIONSETS obtains high step label accuracy, but substantially underpredicts background, as evidenced by both the all label accuracy and the low predicted background percentage. The tendency of ORDEREDDISCRIM to overpredict background which we saw in the supervised setting (row S4) is even more pronounced in this weakly-supervised setting (row U3), resulting in scores very close to the predict background baseline (B1).

Next, we use **narration** constraints (U5), which

[9] We also perform an ablation study to understand the effect of the duration model. See Appendix F for details.

	All Label Acc.	Step Label Acc.	Step Recall
OrderedDiscrim	71.3	1.2	17.9
HSMM+Narr+Or	66.0	5.6	14.2

Table 3: Unsupervised and weakly supervised results in the cross-validation setting.

are enforced only during training time, following Zhukov et al. (2019). Narration constraints substantially improve all label accuracy (comparing U1 and U5). However, the model overpredicts background, likely because it doesn't enforce each step type to occur in a given video. Overpredicting background causes step label accuracy and step recall to decrease.

Finally, we compare the HSMM and OrderedDiscrim models when using both narration constraints (in training) and ordering constraints (in training and testing) in the **ordering + narration** block. Both models benefit substantially from narration on all metrics when compared to using only ordering supervision, more than doubling their performance on step label accuracy and step recall (comparing U6 and U7 to U3 and U4).

Our weakly-supervised results show that:

(1) *Both* action segmentation metrics – all label accuracy and step label accuracy – are important to evaluate whether models adequately distinguish meaningful actions from background.

(2) Step constraints derived from the canonical step ordering provide a strong inductive bias for unsupervised step induction. Past work requires these constraints and the HSMM, when trained without them, does poorly, learning to capture diversity across videos rather than to identify steps.

(3) However, ordering supervision alone is not sufficient to allow these models to learn better segmentations than a simple baseline that just uses the ordering to assign labels (*ordered uniform*); narration is also required.

7.4 Comparison to Past Work

Finally, we compare our full model to the OrderedDiscrim model of Zhukov et al. (2019) in the primary data evaluation setup from that work: averaging results over 20 random splits of the primary data (Table 3). This is a low data setting which uses only 30 videos per task as training data in each split.

Accordingly, both models have lower performance, although the relative ordering is the same: higher step label accuracy for the HSMM, and higher all label accuracy and step recall for OrderedDiscrim. Although in this low-data setting, models overpredict background even more, this problem is less pronounced for the HSMM: 97.4% of timesteps for OrderedDiscrim are predicted background (explaining its high all label accuracy), and 87.1% for HSMM.

8 Discussion

We find that unsupervised action segmentation in naturalistic instructional videos is greatly aided by the inductive bias given by typical step orderings within a task, and narrative language describing the actions being done. While some results are more mixed (with the same supervision, different models are better on different metrics), we do observe that across settings and metrics, step ordering and narration increase performance. Our results also illustrate the importance of strong baselines: without weak supervision from step orderings and narrative language, even state-of-the-art unsupervised action segmentation models operating on rich video features underperform feature-agnostic baselines. We hope that future work will continue to evaluate broadly.

While action segmentation in videos from diverse domains remains challenging – videos contain both a large variety of types of depicted actions, and high visual variety in how the actions are portrayed – we find that structured generative models provide a strong benchmark for the task due to their abilities to capture the full diversity of action types (by directly modeling distributions over action occurrences), and to benefit from weak supervision. Future work might explore methods for incorporating richer learned representations both of the diverse visual observations in videos, and the narration that describes them, into such models.

Acknowledgments

Thanks to Dan Klein, Andrew Zisserman, Lisa Anne Hendricks, Aishwarya Agrawal, Gábor Melis, Angeliki Lazaridou, Anna Rohrbach, Justin Chiu, Susie Young, the DeepMind language team, and the anonymous reviewers for helpful feedback on this work. DF is supported by a Google PhD Fellowship.

References

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. 2016. YouTube-8M: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*.

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. 2016. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dare A Baldwin, Jodie A Baird, Megan M Saylor, and M Angela Clark. 2001. Infants parse dynamic action. *Child Development*, 72(3):708–717.

Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. 2014. Weakly supervised action labeling in videos under ordering constraints. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the Kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. 2019. D3TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li Ding and Chenliang Xu. 2018. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jason Eisner. 2016. Inside-outside and forward-backward algorithms are just backprop (tutorial paper). In *Proceedings of the Workshop on Structured Prediction for NLP*.

Ehsan Elhamifar and Zwe Naing. 2019. Unsupervised procedure learning via joint dynamic summarization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Yazan Abu Farha and Jurgen Gall. 2019. MS-TCN: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. CNN architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. 2016. Connectionist temporal modeling for weakly supervised action labeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The Kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Klein and Christopher D. Manning. 2002. Conditional structure versus conditional estimation in NLP models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hilde Kuehne, Ali Arslan, and Thomas Serre. 2014. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hilde Kuehne, Alexander Richard, and Juergen Gall. 2017. Weakly supervised learning of actions from transcripts. In *CVIU*.

Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. 2019. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. 2017. Temporal convolutional networks for action segmentation and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dani Levine, Daphna Buchsbaum, Kathy Hirsh-Pasek, and Roberta M Golinkoff. 2019. Finding events in a continuous world: A developmental account. *Developmental Psychobiology*, 61(3):376–389.

Percy Liang and Dan Klein. 2008. Analyzing the errors of unsupervised learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605.

Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nicholas Johnston, Andrew Rabinovich, and Kevin Murphy. 2015. What's cookin'? interpreting cooking videos using text, speech and vision. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Bridgette A Martin and Barbara Tversky. 2003. Segmenting ambiguous events. In *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Kevin Murphy. 2002. Hidden semi-markov models. *Unpublished tutorial*.

Janne Pylkkonen and Mikko Kurimo. 2004. Duration modeling techniques for continuous speech recognition. In *Eighth International Conference on Spoken Language Processing*.

Alexander Richard, Hilde Kuehne, and Juergen Gall. 2017. Weakly supervised action learning with RNN based fine-to-coarse modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alexander Richard, Hilde Kuehne, and Juergen Gall. 2018. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. 2012. A database for fine grained activity detection of cooking activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252.

Roger C Schank and Robert P Abelson. 1977. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press.

Fadime Sener and Angela Yao. 2018. Unsupervised learning and segmentation of complex activities from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

O. Sener, A. Zamir, S. Savarese, and A. Saxena. 2015. Unsupervised semantic parsing of video collections. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Tanya Sharon and Karen Wynn. 1998. Individuation of actions from continuous motion. *Psychological Science*, 9(5):357–362.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*.

Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. 2016. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. 2019. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*.

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. 2019. COIN: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ercenur Ünal, Yue Ji, and Anna Papafragou. 2019. From event representation to linguistic meaning. *Topics in Cognitive Science*.

Huijuan Xu, Abir Das, and Kate Saenko. 2017. R-C3D: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. 2018. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision (IJCV)*, 126(2-4):375–389.

Shoou-I Yu, Lu Jiang, and Alexander Hauptmann. 2014. Instructional videos for unsupervised harvesting and learning of action examples. In *Proceedings of the ACM International Conference on Multimedia (MM)*.

Shun-Zheng Yu. 2010. Hidden semi-markov models. *Artificial intelligence*, 174(2):215–243.

Jeffrey M Zacks and Khena M Swallow. 2007. Event segmentation. *Current Directions in Psychological Science*, 16(2):80–84.

Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. 2017. Temporal action

detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Luowei Zhou, Xu Chenliang, and Jason J. Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*.

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. 2019. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

A Optimization

For both training conditions for our semi-Markov models that require gradient descent (generative unsupervised and discriminative supervised), we initialize parameters randomly and use Adam (Kingma and Ba, 2015) with an initial learning rate of 5e-3, a batch size of 5 videos, and decay the learning rate when training log likelihood does not decrease for more than one epoch.

B Features

For our features $x_{1:T}$, we use the same base features as Zhukov et al. (2019). There are three feature types: activity recognition features, produced by an I3D model (Carreira and Zisserman, 2017) trained on the Kinetics-400 dataset (Kay et al., 2017); object classification features, from a ResNet-152 (He et al., 2016) trained on ImageNet (Russakovsky et al., 2015), and audio classification features[10] from the VGG model (Simonyan and Zisserman, 2015) trained by Hershey et al. (2017) on a preliminary version of the YouTube-8M dataset (Abu-El-Haija et al., 2016).[11]

For the generative mdoels which use Gaussian emission distributions, we apply PCA to the base features above to reduce the feature dimensionality and decorrelate dimensions. We perform PCA separately for features within task and within each feature group (I3D, ResNet, and audio features), but on features from all videos within that task. We use 100 components for each feature group, which explained roughly 70-100% of the variance in the features, depending on the task and feature group. The 100-dimensional PCA representations for the I3D, ResNet, and audio features for each frame, at timestep t, are then concatenated to give a 300-dimensional vector for the frame, x_t.

C Unsupervised Evaluation

The HSMM model, when trained in a fully unsupervised setting, induces class labels for regions in the video; however while these class labels are distinct, they do not correspond *a priori* to any of the actual region labels (which can be step types, or background) for our task. Just as with other unsupervised tasks and models (*e.g.*, part-of-speech induction), we need a mapping from these classes to step types (and background) in order to evaluate the model's predictions. We follow the evaluation procedure of past work (Sener and Yao, 2018; Sener et al., 2015) by finding the mapping from model states to region labels that maximizes label accuracy, averaged across all videos in the task, using the Hungarian method (Kuhn, 1955). This evaluation condition is only used in the "Unsupervised" section of Table 2 (in the rows marked with *optimal accuracy assignment*).

D Evaluation Metrics

Label accuracy The standard metric for action segmentation (Sener and Yao, 2018; Richard et al., 2018) is timestep label accuracy, in datasets with a large amount of background, label accuracy on non-background timesteps. The CrossTask dataset has multiple reference step labels in the groundtruth for around 1% of timesteps, due to noisy region annotations that overlap slightly. We obtain a single reference label for these timesteps by taking the step that appears first in the canonical step ordering for the task. We then compute accuracy of the model predictions against these reference labels across all timesteps and all videos for a task (in the *all label accuracy* condition), or by filtering to those timesteps which have a step label (non-background) in the reference (to focus on the model's ability to accurately predict step labels), in the *step label accuracy* condition.

Step recall This metric (Zhukov et al., 2019) measures a model's ability to pick out instants for each of the possible step types for a task, if they occur in a video. The model of Zhukov et al. (2019) predicted a single frame for each step type; while our extension of their model, OrderedDiscrim, and our HSMM model can predict multiple, when computing this metric we obtain a single frame for each step type to make the numbers comparable to theirs.When a model predicts multiple frames per step type, we obtain a single one by taking the one closest to the middle of the temporal extent of the predicted frames for that step type. We then apply their recall metric: First, count the number of *recovered steps*, step types from the true labels for the video that were identified by one of the predicted labels (have a predicted label of the same type at one of the true label's frames). These recovered step counts are summed across videos in the evaluation set for a given task, and normalized by

[10] https://github.com/tensorflow/models/tree/master/research/audioset/vggish

[11] We also experiment with using features derived from transcribed narration in Appendix G.

the maximum number of possible recovered steps (the number of step types in each video, summed across videos) to produce a step recall fraction for the task.

Sequence similarity This measures the similarity of the predicted sequence of regions in a video against the true sequence of regions. As in speech recognition, we are interested in the high-level sequence of steps recognized in a video (and wish to abstract away from noise in the boundaries of the annotated regions). We first compute the negated Levenshtein distance between the true sequence of steps and background $r_1, \ldots, r_K$ for a video and the and predicted sequence $\hat{r}_1, \ldots, \hat{r}'_K$. The negated distance for the sequence pairs for a given video are scaled to be between 0 and 100, where 0 indicates the Levenshtein distance is the maximum possible between two sequences of their respective lengths, and 100 corresponds to the sequences being identical. These similarities are then averaged across all videos in a task.

E Comparing Generative and Discriminative Models

We observe that the generative models tend to obtain higher performance on the action segmentation task, as measured by step label accuracy, than discriminative models of the same or similar class. We attribute this finding to two factors: first, the generative models explicitly parameterize probabilities for the steps, allowing better modeling of the full distribution of step labels. Second, the discriminative models are trained to optimize $p(l_t \mid x_t)$ for all timesteps t. We would expect that this would produce better accuracies on metrics aligned with this objective (Klein and Manning, 2002) – and indeed the all timestep accuracy is higher for the discriminative models. However, the discriminative models' high accuracy often comes at the expense of predicting background more frequently, leading to lower performance on step label accuracy.

F Duration Model Ablation

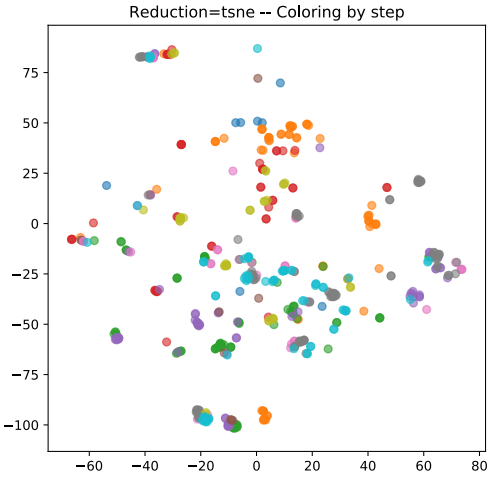
We examine the effect of the (hidden) semi-Markov model's Poisson duration model by comparing to a (hidden) Markov model (HMM in the unsupervised/weakly-supervised settings, or MM in the supervised setting). We use the model as described in Sec. 4 except for fixing all durations to be a single timestep. We then train as described in Sec. 4.1. While this does away with explicit modeling of duration, the transition distribution still allows the model to learn expected durations for each region type by implicitly parameterizing a geometric distribution over region length. Results are shown in 4. We observe that results are overall very similar, with the exceptions that removing the duration model decreases performance substantially on all metrics in the discriminative supervised setting, and increases performance on step label accuracy and step recall in the constrained unsupervised setting (HSMM+Ord+Narr and HMM+Ord+Narr). This suggests that the HMM transition distribution is able to model region duration as well as the HSMM's explicit duration model, or that duration overall plays a small role in modeling in most settings relative to the importance of the features.

Model	All Label Acc.	Step Label Acc.	Step Recall	Seq. Sim.
Supervised				
SMM, gen.	60.5	49.4	28.7	46.6
MM, gen.	60.1	48.6	28.2	46.8
SMM, disc.	66.0	37.3	24.1	50.5
MM, disc.	62.8	32.2	20.1	41.8
Weakly-Supervised				
HSMM	31.8	28.8	10.6	31.0
HMM	28.8	30.8	10.3	29.9
HSMM+Ord+Narr	61.2	15.9	17.2	55.0
HMM+Ord+Narr	60.6	17.0	20.0	55.0

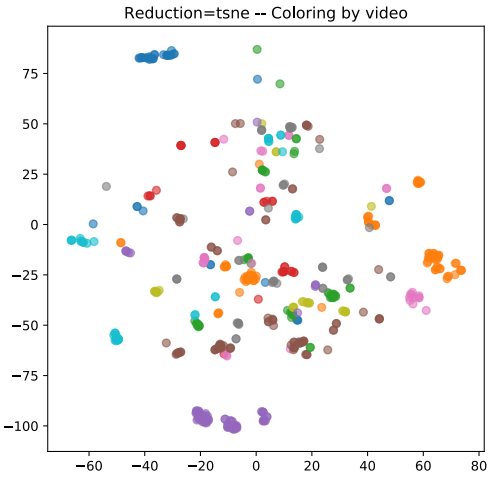
Table 4: Comparison between the semi-Markov and hidden semi-Markov models (SMM and HSMM) with the Markov and hidden Markov (MM and HMM) models, which ablate the semi-Markov's duration model.

G Narration Features

The benefit of narration-derived hard constraints on labels (following past work by Zhukov et al. 2019) raises the question of how much narration would help when used to provide features for the models. We obtain narration features for each video using FastText word embeddings (Mikolov et al., 2018) for the video's time-aligned transcribed narration (see Zhukov et al. 2019 for details on this transcription), pooled within a sliding window to allow for imperfect alignment between activities mentioned in the narration and their occurrence in the video. The features for a given timestep t are produced by a weighted sum of embeddings for all the words in the transcribed narration within a 5-second window of t (*i.e.*from $t-2$ to $t+2$), weighted using



(a) Feature vectors colored by their step label in the reference annotations.



(b) Feature vectors colored by the id of the video they occur in.

a Hanning window[12] (so that words in the center of each window are most heavily weighted for that window). We did not tune the window size, or experiment with other weighting functions. The word embeddings are pretrained on Common Crawl, and are not fine-tuned with the rest of the model parameters.

Once these narration features are produced, as above, we treat them in the same way as the other feature types (activity recognition, object classification, and audio) described in Appendix B: reducing their dimensionality with PCA, and concatenating them with the other feature groups to produce the features x_t.

In Table 5, we show performance of key supervised and weakly-supervised models on the validation set, when using these narration features in addition to activity recognition, object detection, and audio features. Narration features improve performance over the corresponding systems from Table 2 (differences are shown in parentheses) in 13 out of 15 cases, typically by 1-4%.

H Feature Visualizations

To give a sense for feature similarities both within step types and within a video, we visualize feature vectors for 20 videos randomly chosen from the *change a tire* task, dimensionality-reduced using t-SNE (Maaten and Hinton, 2008) so that similar feature vectors are close in the visualization.

Figure 4a shows feature vectors colored by step type: we see little consistent clustering of feature vectors by step. On the other hand, we observe a great deal of similarity across step types within a video (see Figure 4b); when we color feature vectors by video, different steps from the *same* video are close to each other in space. These together suggest that better featurization of videos can improve action segmentation.

[12] https://docs.scipy.org/doc/numpy/reference/generated/numpy.hanning.html

	All Label Acc.	Step Label Acc.	Step Recall
Supervised			
Gaussian mixture	70.4 (+1.0)	43.7 (+3.1)	34.9 (+3.4)
SMM, generative	63.3 (+2.8)	53.2 (+3.8)	32.1 (+3.4)
Weakly-Supervised			
HSMM+Ord	53.6 (-1.9)	9.5 (+1.2)	8.5 (+1.2)
HSMM+Narr	68.9 (+3.2)	8.0 (-1.6)	12.6 (+4.1)
HSMM+Narr+Ord	64.3 (+3.1)	17.9 (+2.0)	21.9 (+4.7)

Table 5: Performance of key supervised and weakly-supervised models on the validation data when adding narration vectors as features. Numbers in parentheses give the change from adding narration vectors to the systems from Table 2.

I Segmentation Visualizations

In the following pages, we show example segmentations from the various systems. Figure 5 and 6 visualize predicted model segmentations for the unstructured Gaussian mixture and structured semi-Markov model in the supervised setting, in comparison to the ground-truth and the ordered uniform baseline. We see that while both models typically make similar predictions in the same temporal regions of the video, the structured model produces steps that are much less fragmented.

Figure 7 and 8 visualize segmentations in the

unsupervised and weakly-supervised settings for the HSMM model and OrderedDiscrim of Zhukov et al. (2019). The unsupervised HSMM has difficulty distinguishing steps from background (see Appendix H), while the model trained with weak supervision from ordering and narration (HSMM+Ord+Narr) is better able to induce meaningful steps. The OrderedDiscrim model, although it has been modified to allow predicting multiple timesteps per step, collapses to predicting a single label, background, nearly everywhere, which we conjecture is because the model is discriminatively trained: jointly inferring labels that are easy to predict, and the model parameters to predict them.

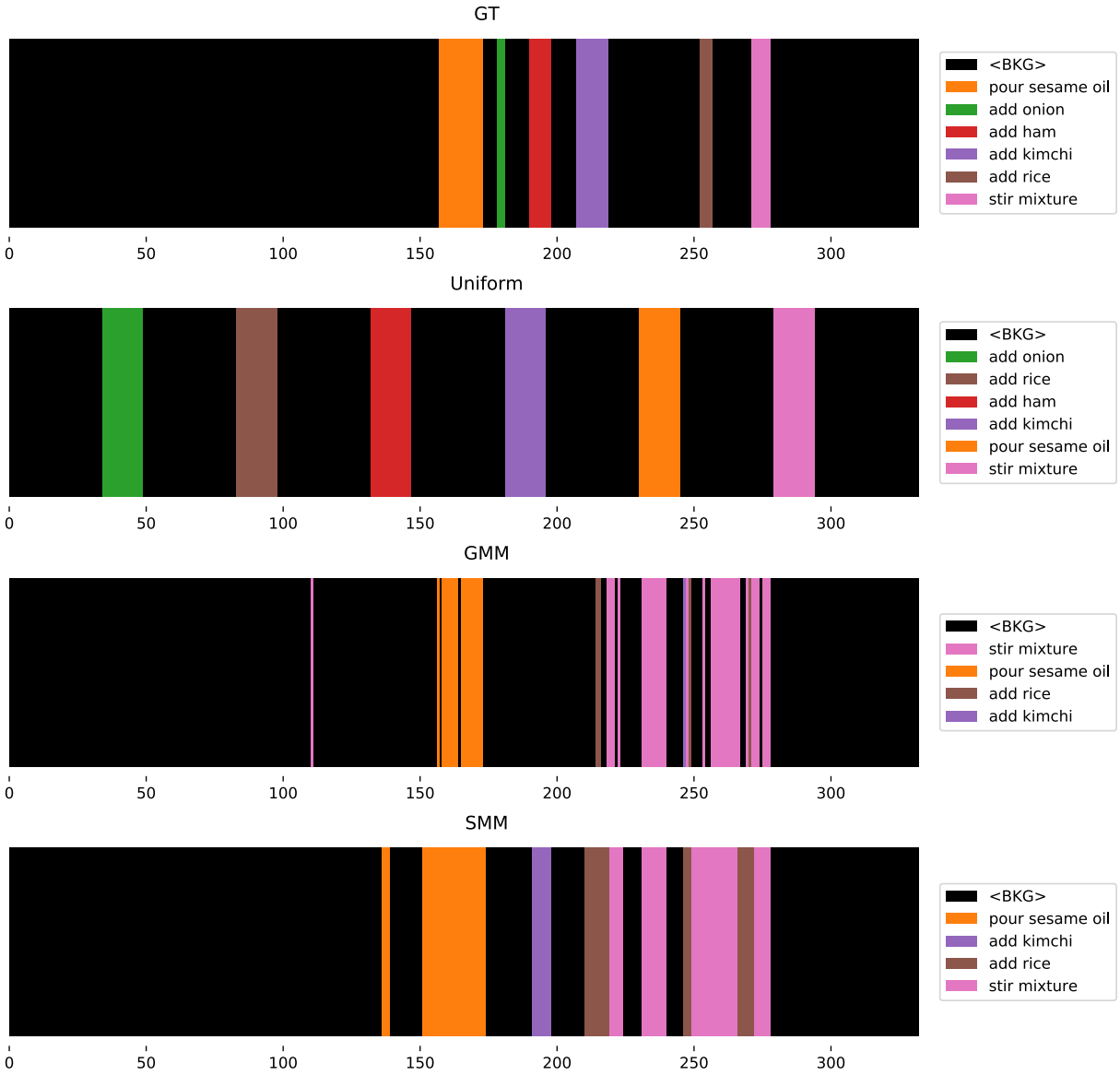


Figure 5: **Supervised segmentations** We visualize segmentations from the validation set for a video from the task *make kimchi fried rice*. We show the ground truth (GT), ordered uniform baseline (Uniform), and predictions from the unstructured Gaussian mixture model (GMM), and structured semi-Markov model (SMM) trained in the supervised setting. Predictions from the unstructured model are more fragmented than predictions from the SMM. The x-axis gives the timestep in the video.

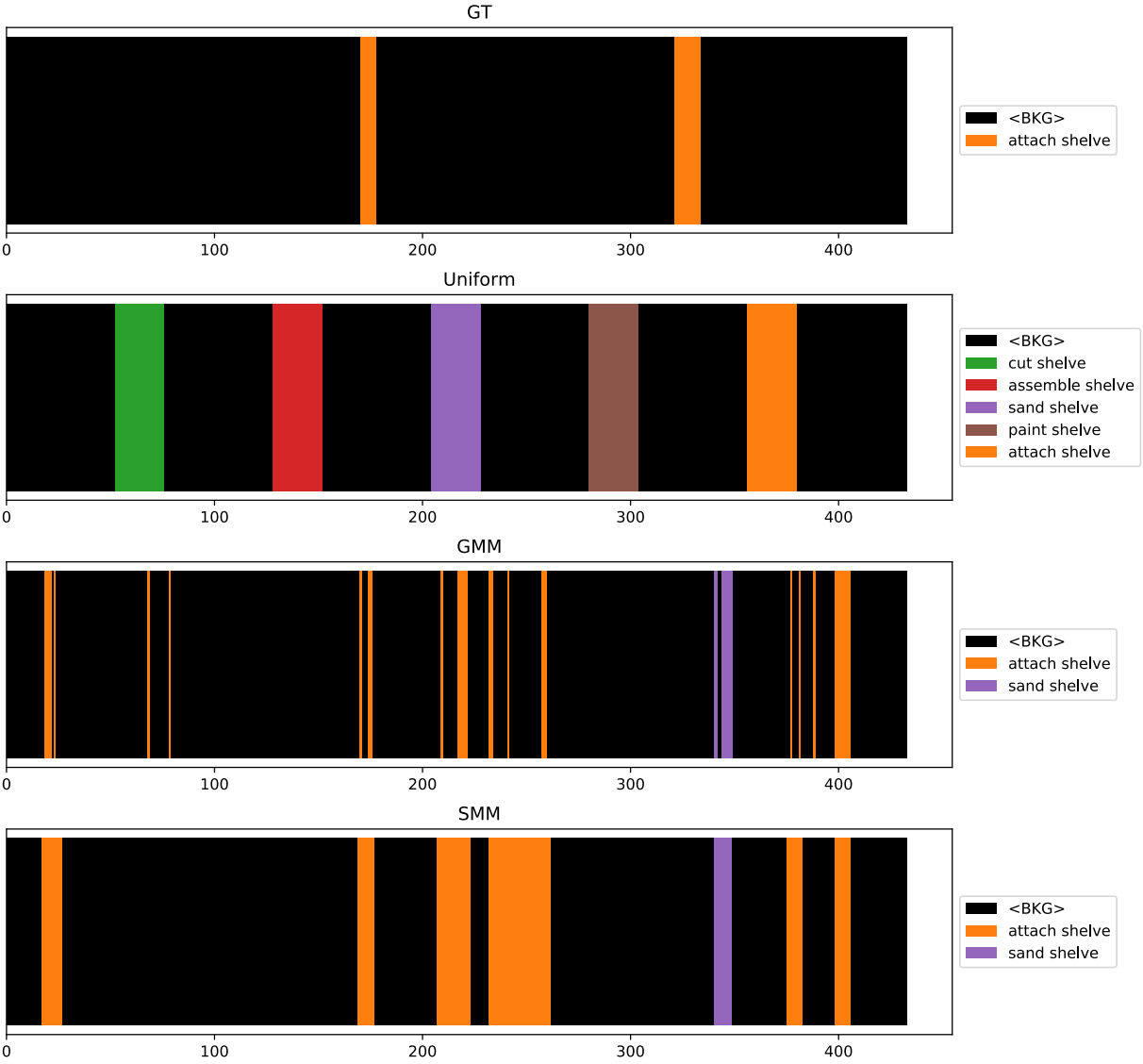


Figure 6: **Supervised segmentations** We visualize segmentations from the validation set for a video from the task *build simple floating shelves*. We show the ground truth (GT), ordered uniform baseline (Uniform), and predictions from the unstructured Gaussian mixture model (GMM), and structured semi-Markov model (SMM) trained in the supervised setting. Predictions from the unstructured model are more fragmented than predictions from the SMM. The x-axis gives the timestep in the video.

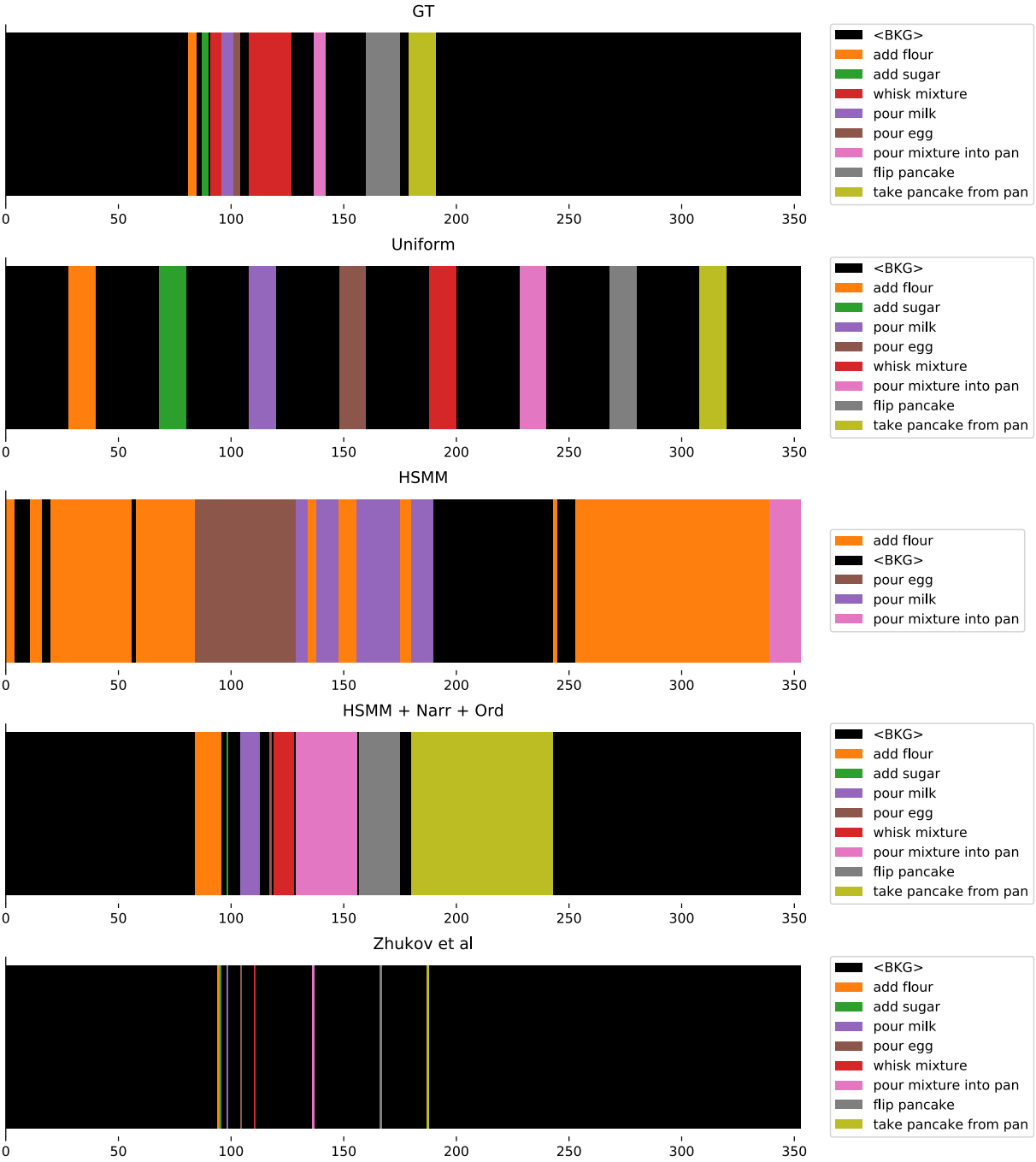


Figure 7: **Unsupervised and weakly-supervised segmentations** We visualize segmentations from the validation set for a video from the task *make pancakes*. We show the ground truth (GT), ordered uniform baseline (Uniform), and predictions from the hidden semi-markov trained without constraints (HSMM) and with constraints from narration and ordering (HSMM+Narr+Ord), and the system of Zhukov et al. The x-axis gives the timestep in the video.

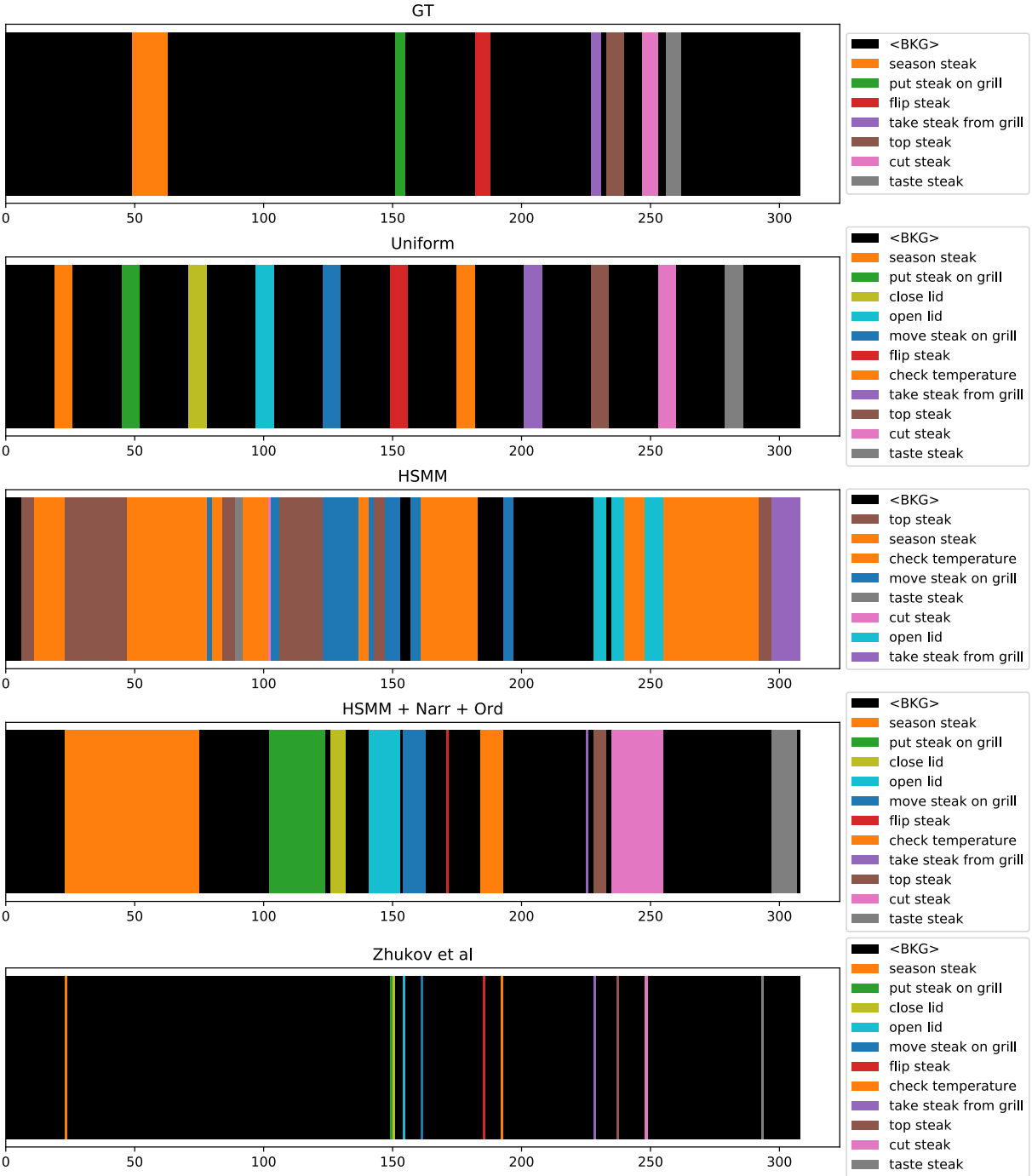


Figure 8: **Unsupervised and weakly-supervised segmentations** We visualize segmentations from the validation set for a video from the task *grill steak*. We show the ground truth (GT), ordered uniform baseline (Uniform), and predictions from the hidden semi-markov trained without constraints (HSMM) and with constraints from narration and ordering (HSMM+Narr+Ord), and the system of Zhukov et al. The x-axis gives the timestep in the video.