GEPSAN: Generative Procedure Step Anticipation in Cooking Videos

Mohamed A. Abdelslam[1], Samrudhdhi B. Rangrej[1*], Isma Hadji[1*], Nikita Dvornik[2*],
Konstantinos G. Derpanis[1,3], Afsaneh Fazly[1]
[1]Samsung AI Centre , [2]Waabi , [3]York University
{m.abdelsalam, s.rangrej, isma.hadji, a.fazly}@samsung.com,
dvornik.nikita@gmail.com, kosta@yorku.ca

Abstract

We study the problem of future step anticipation in procedural videos. Given a video of an ongoing procedural activity, we predict a plausible next procedure step described in rich natural language. While most previous work focuses on the problem of data scarcity in procedural video datasets, another core challenge of future anticipation is how to account for multiple plausible future realizations in natural settings. This problem has been largely overlooked in previous work. To address this challenge, we frame future step prediction as modelling the distribution of all possible candidates for the next step. Specifically, we design a generative model that takes a series of video clips as input, and generates multiple plausible and diverse candidates (in natural language) for the next step. Following previous work, we side-step the video annotation scarcity by pretraining our model on a large text-based corpus of procedural activities, and then transfer the model to the video domain. Our experiments, both in textual and video domains, show that our model captures diversity in the next step prediction and generates multiple plausible future predictions. Moreover, our model establishes new state-of-the-art results on YouCookII, where it outperforms existing baselines on the next step anticipation. Finally, we also show that our model can successfully transfer from text to the video domain zero-shot, i.e., without fine-tuning or adaptation, and produces good-quality future step predictions from video.

1. Introduction

Anticipating future steps while performing a task is a natural human behaviour necessary to successfully accomplish a task and cooperate with other humans. Thus, it is important for a smart AI agent to exhibit this behaviour too, in order to assist humans in performing procedural tasks (*e.g.*, cooking, assembling furniture or setting up an electronic device). For example, consider a cooking AI assistant that observes a user as they cook a dish. To be useful, this assistant needs to anticipate possible next steps in order to provide timely support with ingredients, cooking tools and actions. Anticipating future steps from a video stream is a challenging task, where simply recognizing the current action or objects is not sufficient. To anticipate future actions, one needs to parse and recognize the human-object interactions from an unstructured and cluttered environment in the current frame, predict the possible task being performed (possibly leveraging the past observations) and finally anticipate plausible next steps. Given the importance and challenges associated with this task, several research efforts targeted this application in the recent years [24, 25, 9, 37].

We follow recent work [24] and tackle future anticipation in the realm of cooking activities. Given visual observations of the first t steps, our task is to predict the next step to be executed. This task entails recognizing the current step and the recipe being made, which is particularly challenging given the modest size of cooking video datasets

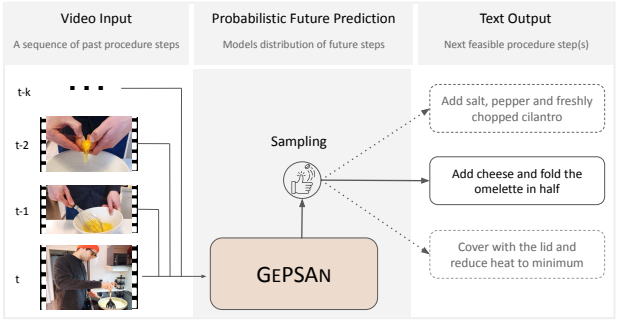


Figure 1. **Summary of the proposed GEPSAN model.** Our model, given an initial video stream representing a sequence of past procedural steps, predicts multiple feasible alternatives for the next step in natural language. We first train our model on text-only data, followed by zero-shot transfer to the video domain.

*Equal Contribution

with annotations. Fortunately, such instructional knowledge is available in abundance in the text domain (think of all the dish recipes online) and can be leveraged to help video prediction. Prior work [24, 32] builds on this observation and proposes to first pretrain the anticipation model on a large corpus of text-based recipes, *i.e.*, Recipe1M+ [17] to acquire knowledge about the recipe domain, and then fine-tune the model on visual data. This line of work effectively alleviates the video annotation problem, however, these works only predict a single future realization, and thus it does not take into account all the variability present in the recipes. For example, given the task of making a salad, and assuming the first three observed step are: *Chop Vegetable*, *Add Tomatoes*, *Add Cucumber*, the plausible next step can be: *Add Olive Oil* or *Add Salt and Pepper* (for those who like more seasoning) or simply *Serve*. This simple example highlights that the task's output is, in fact, multi-modal. This observation suggests that a good future step anticipation model must be able to predict diverse and plausible future realizations. Moreover, it is known that in a multi-modal setup using a model that outputs a single prediction (as done by previous work [24, 32]) may harm the performance [35] even further by producing unrealistic samples that "fall between" the true modes.

In this work, we embrace the uncertainty inherent in the task of future anticipation and propose to learn a Generative Procedure Step Anticipation model (GEPSAN), that captures the distribution of possible next steps and allows to sample multiple plausible outputs given an initial observation from video input. A summary of our proposed work is depicted in Figure 1. To achieve this goal, we design a model that consists of three modules: 1) a modality encoder, that ingests a sequence of previous instruction step observations (in either text or video format), 2) a generative recipe encoder, that, given the observation history, proposes the next plausible instruction vector, and 3) an instruction decoder, that transforms the next step prediction (given by the recipe encoder) into rich natural language. The core component of the model is the generative recipe encoder; it combines the benefits of the transformer model [28] (to process long input sequences) and Conditional Variational AutoEncoder (CVAE) (to capture the uncertainty inherent to the task) and can produce multiple plausible alternatives for the next step in a procedure. Another key element of the pipeline is the input encoder; in contrast to the previous works that learn it from scratch, we adapt a pretrained video-language feature extractor [16] to serve as our encoder. Since the encoder has been trained to map video and text into a common embedding space, our model, trained only on the recipe text corpus, can generalize to future step anticipation from video zero-shot, without any finetuning.

Contributions. Our contributions are twofold:

- We propose GEPSAN, a new generative model for future step anticipation that captures the uncertainty inherent in the task of next step prediction.
- We show that GEPSAN, only trained using text recipes, can generalize to video step anticipation zero-shot, without finetuning or adaptation.

Thanks to that, we achieve state-of-the-art results in next step anticipation from video on YouCookII [40] and show that our model can generate diverse plausible next steps, outperforming the baselines in modelling the distribution of next steps.[1]

2. Related work

Procedure planning and future anticipation. Previous work on future anticipation [5, 6, 20] and procedure planning in video [1, 3, 27] is mainly based on the visual modality and relies on strong visual supervision. Moreover, action anticipation is often considered as a classification problem, where the task is to predict a future action label from a predefined closed action set, *e.g.*, [9, 31, 10, 19, 39, 34]. Unlike most previous work, we rely on weak supervision from the language modality, and predict future actions in rich natural language. This allows us to transfer knowledge from a large scale text data to the visual domain, while only requiring a small text-aligned video data.

A few recent studies draw on language instructions as a source of weak supervision to perform future anticipation for procedural activities [25, 24, 32, 8]. Given a portion of an instructional video, these models predict plausible future actions, expressed using natural language. Whereas the model of [25, 24] works by re-using the text recipe model parameters and fitting a visual encoder to the rest of the pretrained model, the models of [32, 8] transfer textual knowledge to the task of visual action anticipation via knowledge distillation. Our work is most similar to that of [25, 24], but we offer a number of improvements. First, we design a modern transformer-based architecture and present an improved training objective with complementary loss functions. Second, for efficient transfer learning across modalities, we leverage single-modality (language and video) encoders that are jointly trained for cross-modality alignment. Importantly, thanks to this design choice, we not only show improved predictions compared to relevant previous work of [24, 25], but also present a new benchmark in zero-shot cross-modality transfer with competitive performance. Further, we consider diversity in future prediction task to capture the inherent uncertainty in future anticipation, which is overlooked in most related prior work on future anticipation from video [24, 25]. Notably, recent work on proba-

[1] Our code will be available at https://github.com/SamsungLabs/GePSAN

bilistic procedure planning [36] also explicitly tackles uncertainty. However, unlike this work, which is conditioned on the start and end of the procedure, we model uncertainty in a more challenging setting, where we only observe the start of the procedure, thereby making an approach modeling uncertainty even more relevant.

Visual-textual representation learning. The ability to learn with minimal supervision is becoming increasingly important, and as such recent work focuses on the complementarity across the visual and textual modalities as an inexpensive source of supervision [4, 23, 36, 26]. To further enable the use of cross-modal supervision, a large body of work uses *aligned* multimodal data for learning rich representations that can be adapted for downstream tasks with minimal finetuning on task-specific annotations [16, 18, 30]. Such methods learn multiple single-modality encoders, each producing features aligned with the other modalities. We leverage the multimodal alignment offered by one such model - UniVL [16] - to facilitate cross-modal transfer learning. For our initial text-only model, we adopt a frozen pretrained UniVL text encoder. Later, for transfer learning to videos, we replace the above encoder with a frozen pretrained UniVL video encoder. Importantly, our model design is not tied to this specific encoder but can readily be adapted to leverage stronger modality encoders.

Modeling uncertainty and diversity. Many previous works explicitly model uncertainty inherent to the task and leverage it to produce diverse output. Recent works [22, 33] propose VAE based methods for diverse human motion synthesis. Other work proposes an RNN augmented with a VAE for stochastic sequence modelling [11] . They evaluate their approach on speech and sequential image generation. In the context of action anticipation, some approaches use VAE to predict diverse future actions for the objects in the static image in terms of pixel-wise motion, *e.g.*, [29], while others use conditional GAN for long-term discrete action label anticipation [37]. Unlike the existing works, we tackle diversity for future action prediction in natural language. Thus, we find inspiration from various approaches for diverse dialogue modelling. DialogWAE [12] is a Wasserstein autoencoder (WAE) based solution for dialogue modeling. SPACEFUSION [7] relies on the fusion between seq2seq model and VAE for diverse Neural Response Generation. Knowledge-Guided CVAE [38] provides discourse-level diversity for open-domain conversations. Similar to the above works, we adopt conditional VAE in our model to predict multiple plausible next steps in rich natural language.

3. Technical approach

In this section, we formalize the problem of multi-modal future step anticipation, where the task is to predict *multiple* plausible next steps (Sec. 3.1). We then describe our solution - a generative future step prediction model (Sec. 3.2). Sec. 3.3 describes the proposed training objectives that we use to train a model from a pure text-based corpus. Next, Sec. 3.4 describes how to transfer the learned model to the video domain, with little fine-tuning or completely zero-shot. Finally, we provide implementation details in Sec. 3.5.

3.1. Problem definition

In this work we tackle the task of next step anticipation in procedural activities, *e.g.* cooking, and propose to explicitly capture the multi-modal nature inherent to the task of future prediction. Specifically, given the first t steps $s_{1:t}$ of a recipe, in video (or textual) format, our model outputs one or multiple ($k \geq 1$) plausible options for the next procedure step, each expressed as a natural language sentence, *i.e.*, $\{s_{t+1}^{(1)}, \cdots, s_{t+1}^{(k)}\}$. To better specify the prediction problem, we follow the prior work [24, 25, 32] and use the ingredient list as the 0-th step, s_0.

3.2. Model

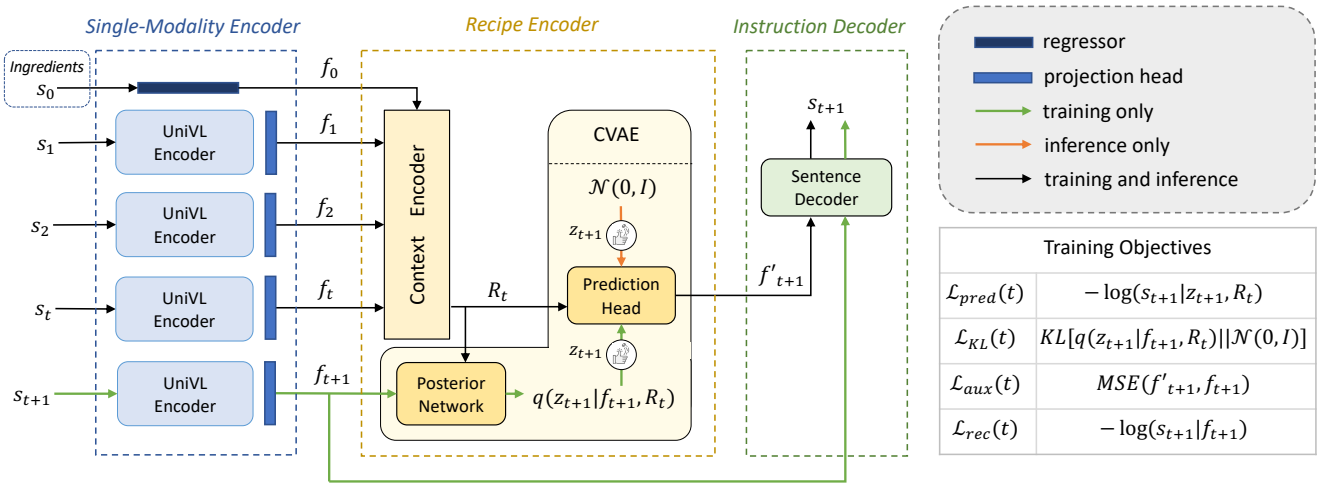
We illustrate our model in Figure 2. Our model consists three modules; namely, a single modality encoder, a recipe encoder and an instruction decoder.

Single-modality (text or video) encoder. Given the observed instruction steps, $s_{0:t}$, in text or video domain, our single-modality encoders produce embeddings $f_{0:t}$. Unlike prior work [24, 25], we use UniVL [16] language and video encoders that were pretrained to embed the sentences or video clips into a common feature space. Additionally, we augment UniVL (text or video) features with a learnable projection head, P, such that $f_t = P(\text{UniVL}(s_t))$.

Recipe encoder. The recipe encoder is the core of our model, it takes in a sequence of embeddings, $f_{0:t}$, corresponding to t observed instruction steps, $s_{0:t}$, and outputs multiple plausible future step embeddings, $\{f_{t+1}'^{(1)}, \cdots, f_{t+1}'^{(k)}\}$. It consists of two components: a context encoder and a conditional Variational Auto Encoder (CVAE).

Context encoder. The context encoder is implemented as a transformer [28] block. It aggregates past sentence embeddings $f_{0:t}$ into a single context vector R_t. To take only the past history into account, we use causal attention in our context encoder, that is, we only use $f_{0:t}$ to produce R_t. During training, our context encoder observes all embeddings up to $t-1$ and produces $R_{0:t-1}$ simultaneously.

Conditional VAE. Our CVAE consists of a posterior network and a prediction head. During training, the poste-



Training Objectives	
$\mathcal{L}_{pred}(t)$	$-\log(s_{t+1}\|z_{t+1}, R_t)$
$\mathcal{L}_{KL}(t)$	$KL[q(z_{t+1}\|f_{t+1}, R_t)\|\|\mathcal{N}(0, I)]$
$\mathcal{L}_{aux}(t)$	$MSE(f'_{t+1}, f_{t+1})$
$\mathcal{L}_{rec}(t)$	$-\log(s_{t+1}\|f_{t+1})$

Figure 2. **Detailed view of our GEPSAN.** Our GEPSAN consists of three modules: a single-modality encoder (*i.e.*, text or video encoder), a recipe encoder and an instruction decoder. **At training time:** A single-modality encoder processes steps $s_{1:t+1}$ (text or video format) independently and produces features $f_{1:t+1}$. Next, given $f_{1:t+1}$, a recipe encoder reconstructs f_{t+1} as follows. First, a context encoder predicts a context vector R_t from $f_{1:t}$ and passes it to a CVAE. Then, in the CVAE, i) a posterior network predicts a posterior distribution $q(z_{t+1}|f_{t+1}, R_t)$ from $f_{1:t}$ and R_t, and ii) a prediction head reconstructs f_{t+1} from R_t and a sample $z_{t+1} \sim q(z_{t+1}|f_{t+1}, R_t)$. We denote the reconstructed f_{t+1} as f'_{t+1}. We pass the above f'_{t+1} to the instruction decoder, which predicts s_{t+1} in natural language. Additionally, the instruction decoder also decodes f_{t+1} to predict s_{t+1} in natural language. We train our model using the training objectives shown in the table on the right. **At inference time:** Given $s_{1:t}$ (text or video), a single-modal encoder predicts features $f_{1:t}$. Given $f_{1:t}$, a context encoder predicts a context R_t and passes it to the CVAE. In the CVAE, i) we draw *multiple* samples for z_{t+1} from a Gaussian prior, and ii) we pass each z_{t+1} and R_t to a prediction head, which predicts multiple independent f'_{t+1}. Given multiple f'_{t+1}, a sentence decoder predicts diverse and independent alternatives for s_{t+1}.

rior network ingests the concatentation of the context vector, R_t, (*i.e.*, conditional input) and the next sentence embedding, f_{t+1}, and predicts a posterior $q(z_{t+1}|f_{t+1}, R_t)$. We then sample a latent, $z_{t+1} \sim q(z_{t+1}|f_{t+1}, R_t)$, concatenate it with the context vector, R_t, and pass them to the predication head to predict the embedding of the next instruction f'_{t+1}. During training, we minimize KL divergence between the predicted posterior $q(z_{t+1}|f_{t+1}, R_t)$ and a standard Gaussian prior with zero mean and unit variance. Thus, at inference, we discard the posterior network, sample z_{t+1} from $\mathcal{N}(0, I)$, and follow the same steps as described above. This way, by using a CVAE on top of the context encoder, we essentially learn a distribution of the next steps, conditioned on the observed step history. Another advantage of the CVAE framework is fast sampling at test time.

Instruction decoder. Given a predicted embedding f'_{t+1}, our instruction decoder decodes the next step in natural language. We implement the instruction decoder as a simple LSTM (not a transformer) as it demonstrated better validation results.

At inference, we sample one or multiple ($k \geq 1$) $z^{(k)}_{t+1}$ from $\mathcal{N}(0, I)$ and combine it with a given context R_t. Subsequently, our CVAE predicts multiple embeddings $\{f'^{(1)}_{t+1}, \cdots, f'^{(k)}_{t+1}\}$ and our sentence decoder decodes each embedding into a separate and independent alternative for the next step.

3.3. Training objectives

Our training objective combines three losses that help GEPSAN capture the probability distribution of next steps, provide good sentence decoding, and stabilize training.

Conditional evidence lower bound. Conditional Evidence Lower Bound (or conditional ELBO) is the loss used to train the CVAE and is mostly responsible for capturing the multi-modal distribution associated with the task of next step prediction. The conditional ELBO used to train our model can be expressed as follows:

$$\mathcal{L}_{ELBO}(t) = \mathcal{L}_{pred}(t) + \beta \mathcal{L}_{KL}(t). \quad (1)$$

$$\mathcal{L}_{pred}(t) = -\sum_{j=1}^{M_{t+1}} \log p(w^j_{t+1}|w^{j'<j}_{t+1}, z_{t+1}, R_t), \quad (2)$$

$$\mathcal{L}_{KL}(t) = \mathbf{KL}[q(z_{t+1}|f_{t+1}, R_t)||\mathcal{N}(0, I)], \quad (3)$$

where $z_{t+1} \sim \mathbb{E}_{q(z_{t+1}|f_{t+1}, R_t)}$ and w^j_{t+1} is the j^{th} word out of total M_{t+1} words in the $(t+1)^{th}$ sentence. To avoid posterior collapse in the early epochs, we introduce β coefficient for the KL divergence in the above objective and anneal β linearly [2].

Auxiliary objective. Previous works [38, 11] suggest that an auxiliary loss is essential to train a CVAE along with a seq2seq model. Thus, we introduce an additional auxiliary loss,

$$\mathcal{L}_{aux}(t) = \text{MSE}(f'_{t+1}, f_{t+1}). \quad (4)$$

Note that we compute gradient of the above objective with respect to f'_{t+1}. The embeddings f_{t+1} act as a target only. Notably, the auxiliary loss also simplifies the sentence decoding process (*i.e.*, Eq. 2) by: i) compelling our CVAE to reconstruct the embeddings f'_{t+1} given $[z_{t+1}; R_t]$, and ii) allowing our sentence decoder to decode the $(t+1)^{th}$ sentence from f'_{t+1}.

Sentence reconstruction objective. While the auxiliary loss is designed to simplify the decoding process, initially, it is still difficult for the sentence decoder to predict the next sentence since f'_{t+1} is not yet learnt. Therefore, we also train our sentence decoder to reconstruct the individual sentences from their projected UniVL embeddings, *i.e.*, f_{t+1}. Note that f_{t+1} is a more stable input for the decoder compared to f'_{t+1}. The reconstruction objective is,

$$\mathcal{L}_{rec}(t) = -\sum_{j=1}^{M_{t+1}} \log p(w^j_{t+1} | w^{j'<j}_{t+1}, f_{t+1}). \quad (5)$$

Our final training objective is:

$$\mathcal{L} = \sum_{t=0}^{T-1} \left\{ \mathcal{L}_{ELBO}(t) + \alpha \mathcal{L}_{aux}(t) + \gamma \mathcal{L}_{rec}(t) \right\}. \quad (6)$$

where α and γ are hyperparameters used to balance the training objectives.

Pretraining domain. Due to its flexible design, GEPSAN can ingest instruction step representations $s_{0:t}$ in the form of text or video. While our final objective is to do next step prediction purely from video, the size of annotated video-based cooking datasets does not allow us to train such a model from scratch. Thus, we follow prior work [24, 25] and pretrain GEPSAN on a large corpus of text-only recipes. That is, given a sequence of step sentences as input, our pretraining objective is to model the next step distribution from textual input only, using the training pipeline and the final loss (in Eq. 6) described above. After this pretraining stage, the model can be adapted to take video as input, using a modest amount of fine-tuning or completely zero-shot.

3.4. Transfer learning

After the model has been pretrained on the text corpus, as described above, we adapt the model to accept video snippets as input. To do so, we replace the frozen UniVL sentence encoder with the frozen UniVL video encoder. Since the UniVL video and text features are aligned by design (*i.e.*, they live in the same embedding space), after the above switch, our model readily offers strong future step anticipation performance, without further fine-tuning or adaptation. We refer to this setting as *zero-shot modality transfer*. Optionally, to further boost the step anticipation performance, we can finetune GEPSAN on a small annotated video dataset, as done in previous work [24, 25] by default. It is important to note that the finetuning stage is essential for the previous methods to work, and is only optional in our case, as we can already perform future step anticipation from video without any finetuning with competitive results as we later demonstarte in the experiments section.

3.5. Implementation details

For the loss hyperparameters, we set $\alpha = 3$ for Recipe1M+, and $\alpha = 1$ for YouCookII. We set $\gamma = 1$ in all cases. For Recipe1M+, we set $\beta = 0.2$ with linear KL annealing to reach that value in $100,000$ steps. β is set to 0.1 for YouCookII. We use the Adam optimizer [15] with a learning rate of 0.0001, weight decay of 0.01, and a one epoch linear warm-up. The batch size is set to 50. Regarding the architecture, we used a 3-layer residual block for the UniVL projection head, and a one-layer MLP for the ingredients regressor that takes as input a one-hot vector of ingredients (the number of ingredients is $3,769$). The context encoder is a 6-layer transformer with an input dimension of 512 and 8 heads. The posterior network and the prediction head are both 3-layer MLPs, with the latent variable z having a dimension of 1024. The instruction decoder is a 3-layer LSTM with a hidden size of 512 and a word embedding size of 256.

4. Experiments

In this section, we evaluate the performance of GEPSAN, our proposed approach, on the task of future step anticipation, make a comparison to relevant baselines, and perform an ablation study of the proposed model components. We begin by detailing the experimental setup and adopted evaluation metrics given our newly proposed view, which takes the multi-modal nature of the task into account (Sec. 4.1). We then evaluate our model on the main task of future step anticipation from video input (Sec. 4.2). We finally show the role of pretraining on a large text-based dataset, highlight the flexibility of our model that can take video or text as input, and demonstrate the role of each component in our training objective (Sec. 4.3).

4.1. Experimental setup

Datasets. For the text-only pretraining stage, we follow previous work and take advantage of a large text-based recipe dataset. Specifically, we use the publicly available Recipe1M+ dataset [17], which contains over one million

	Model	*Unseen Split*					*Seen Split*				
		ING	VERB	B1	B4	MET	ING	VERB	B1	B4	MET
Text → Video Zero-shot Transfer	GEPSAN *(S)*	16.5	24.1	23.0	2.2	8.3	-	-	-	-	-
	GEPSAN *(M)*	30.0	28.7	31.4	3.7	11.6	-	-	-	-	-
Finetuning on Video	BASELINE *(S)*	16.8	26.9	25.1	3.1	9.2	19.6	27.5	25.8	4.0	9.8
	GEPSAN *(S)*	21.5	29.9	27.6	4.8	10.8	25.6	30.8	28.9	5.8	11.8
	BASELINE *(M)* ◇	27.8	31.6	33.1	4.4	12.3	32.2	34.2	35.0	5.9	13.7
	GEPSAN *(M)*	31.6	37.8	35.6	7.9	14.5	36.7	38.4	37.1	9.3	15.7

Table 1. **YouCookII future anticipation from video input.** We report results for two settings: (top) zero-shot text-to-video modality transfer and (bottom) finetuning on video modality. For each setting, we compare our results with the baseline [24] results (when available) for single *(S)* and multiple *(M)* next step prediction. To achieve single and multiple predictions, we evaluate GEPSAN using latent $z_{t+1} = 0$ (*i.e.*, mean of a Gaussian prior) and five random $z_{t+1} \sim \mathcal{N}(0, I)$, respectively. ◇We use *Nucleus sampling* [14] to achieve multiple predictions from the deterministic baseline. Further, we present comparison for recipe-types unseen and seen in the training split.

	Model	*Unseen Split*					*Seen Split*				
		ING	VERB	B1	B4	MET	ING	VERB	B1	B4	MET
Zero-shot Dataset Transfer	BASELINE *(S)*	22.9	29.1	25.8	3.0	10.0	-	-	-	-	-
	GEPSAN *(S)*	20.0	28.3	24.7	3.1	9.4	-	-	-	-	-
	BASELINE *(M)* ◇	27.1	31.3	29.3	2.5	11.4	-	-	-	-	-
	GEPSAN *(M)*	**33.3**	**33.7**	**32.4**	**4.7**	**15.2**	-	-	-	-	-
Finetuning on the Target Dataset	BASELINE *(S)*	26.9	31.8	30.6	6.6	12.2	29.1	32.9	31.0	7.3	12.8
	GEPSAN *(S)*	28.9	33.7	33.0	7.2	13.2	32.7	35.2	35.0	8.5	14.4
	BASELINE *(M)* ◇	38.4	38.8	39.3	8.6	15.8	40.0	39.2	39.6	8.8	16.1
	GEPSAN *(M)*	**41.7**	**42.9**	**41.4**	**11.0**	**17.3**	**44.6**	**43.7**	**43.0**	**12.3**	**18.4**

Table 2. **YouCookII future anticipation from text input.** We report results for two settings: (top) zero-shot dataset transfer and (bottom) finetuning on the target data. For each setting, we compare our results with the baseline [24] results (when available) for single *(S)* and multiple *(M)* next step prediction. To achieve single and multiple predictions, we evaluate GEPSAN using latent $z_{t+1} = 0$ (*i.e.*, mean of a Gaussian prior) and five random $z_{t+1} \sim \mathcal{N}(0, I)$, respectively. ◇We use *Nucleus sampling* [14] to achieve multiple predictions from the deterministic baseline. Further, we present comparison for recipe-types unseen and seen in the training split.

cooking recipes to pretrain our model. Then, given that the main target of the proposed approach is future step anticipation from *video* input, we follow previous work [25] and evaluate on the YouCookII dataset [40], a video-based cooking dataset, (*i.e.*, we use Recipe1M+ for learning procedural knowledge from text, and showcase transfer learning to visual domain on YouCookII). YouCookII consists of 2000 long untrimmed videos (in 3rd person viewpoint) from 89 cooking recipes. Each video is associated with an ordered list of steps describing the recipe being performed in free form natural language, together with start and end times of each step in the video. Notably, while previous work also evaluated on the Tasty video dataset [25], it is not used in this work due to copyright limitations.

Evaluation metrics. Our model predicts next steps in free-form natural language, therefore, we follow standard protocol [25, 24] and evaluate using sentence matching scores, including: BLEU1 (B1), BLEU4 (B4) and METEOR (MET). Notably, we use the standard corpus-level calculation method for the BLEU [21] and METEOR scores, while previous work [25, 24] used the average of sentence-level BLEU and METEOR scores; see supplement for a detailed discussion and the complete set of results using both methodologies.

We also calculate the recall on the set of ingredients (ING) and verbs (VERB) included in the ground truth sentence, *i.e.*, we calculate the ratio of the verbs and ingredients in the ground truth predicted by the model. Note that the recalls on ING and VERB are stronger indicators of model performance as they highlight the diversity of the predicted actions rather than the diversity in the sentence styles.

Importantly, unlike previous work, our approach can predict either *multiple (M)* plausible next steps or a *single (S)* next step (*i.e.*, by setting the latent z_{t+1} to the maximum likelihood sample from the latent prior distribution, $\mathcal{N}(0, I)$, which happens to be a zero vector). To evaluate our approach in the *multiple* setting using the same evaluation metrics described above, we have to select just one out of k predicted next steps, since we only have one ground-truth in the dataset. To do so, we pick the predicted step that is closest to the ground truth sentence using the Jaccard

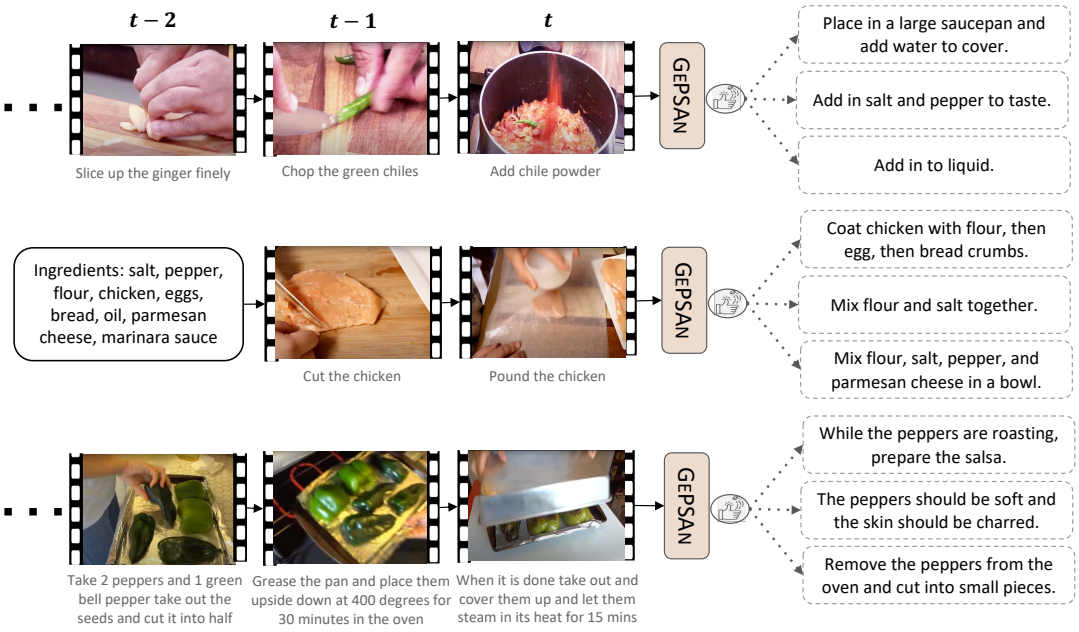


Figure 3. **Qualitative results on the Text → Video Zero-shot Transfer without any finetuning on videos.** Please note that only the video is passed to the model as input, the text here is provided for context.

Similarity (Intersection over Union). Precisely, we treat the words in each sentence as a Bag of Words (BoW) and calculate the Jaccard similarity between each predicted sentence and the ground truth. Notably, we elect to use Jaccard similarity here as it is a well accepted metric for comparing two sentences, but any alternative metric for comparing sentences can be used for this step. Intuitively, this inference procedure is meaningful because the model samples multiple plausible next steps, only one of which is contained in the labels, so the matching step described above associates the ground-truth with the closest predicted sample. k is set to 5 in all the experiemnts. See supplement for the impact of increasing k on the results.

Baselines. We use previous state of the art for future step anticipation from cooking videos [24][2] as our main BASELINE and compare our *single (S)* prediction setting directly to it. For our *multiple (M)* next step prediction setting, there is no directly comparable baseline, to the best of our knowledge. Therefore, we augment our main BASELINE with a simple approach to generate multiple next steps. Specifically, we replace the deterministic greedy approach used in the decoder of the BASELINE, with the Nucleus sampling method [14] to generate k alternatives for the next step.

4.2. Results

YouCookII video-based future anticipation. As previously mentioned, the main target of this work is future step anticipation given video input. Note that we evaluate both the *single* (*i.e.* deterministic) and *multiple* (*i.e.* generative) prediction versions of our approach in two main settings; namely, **(i)** text → video zero-shot transfer and **(ii)** with video finetuning. In the zero-shot modality transfer setting, we use our model pretrained on the text-based Recipe1M+ dataset, and directly replace the textual input with visual input from the YouCookII dataset, and use the UniVL video encoder instead of the text encoder. In the modality finetuning setting, we further finetune the model, except for the pretrained UniVL encoder, on the training split of YouCookII.

In addition to the variations in terms of training settings, we follow previous work [24] and also assess our model on two different splits of the YouCookII dataset; namely, **(i)** unseen split, containing recipes never seen during training, and **(ii)** seen split, containing only seen recipes [24].

Table 1 summarizes our main results on the YouCookII dataset. The results of both variants of our model (*i.e.*, *(S)* vs. *(M)*) compared to the baseline speak decisively in favor of our approach, where we outperform the baseline under all settings in all the considered metrics. Notably, comparing our model to the baseline in the zero-shot modality transfer setting is *not* possible, as the baseline relies on training a new model for the visual modality. In contrast, we judiciously use modality encoders that were pretrained, in unsupervised settings, such that visual and language representations share a common embedding space. This strategy allowed us to focus on pretraining a stronger recipe encoder, which is reflected in results reported in Table 1.

[2]We use the code shared with us by the authors.

Importantly, comparing the *multiple (M)* prediction to *single (S)* prediction settings, highlights the importance of modeling the uncertainty inherent to the task of future anticipation, where a model yielding multiple plausible outputs better captures possible future steps, as evidenced by *(M)* results always outperforming *(S)* results. Figure 3 illustrates some qualitative results from GEPSAN *(M)* , thereby further validating the relevance of the multiple plausible future step predictions. Obtaining corresponding quantitative results to better quantify the plausibility of the generated next steps is unfortunately not possible in the absence of a dataset capturing multiple feasible ground truths, which would allow calculating precision. Curating such a dataset is outside the scope of this paper.

YouCookII text-based future anticipation. While our main goal is video-based future step anticipation, for the sake of completeness we also evaluate our model on text-based future anticipation. The results of predicting *multiple* plausible next steps in Table 2 further confirm the superiority of our approach and the importance of capturing the uncertainty inherent to the task of future prediction. As expected, the results with text-based input are better, compared to the visual-based input, as there is no modality change in these settings. However, although there is no change of modality here, the results before and after finetuning indicate that there is a difference in the distributions of the two datasets. More generally, these results highlight the flexibility of our model that can readily use either textual or visual input in zero-shot settings, unlike previous work [24].

4.3. Ablations

Contribution of the different training objectives. We evaluate the role of each loss component by gradually removing each objective. The results in Table 3 confirm the pivotal role of the auxiliary loss, $\mathcal{L}_{aux}$, to train the CVAE as mentioned in Sec. 3.3. The prediction loss, $\mathcal{L}_{pred}$, also plays an important role in boosting performance. Note that removing the KL divergence, $\mathcal{L}_{KL}$, leads to model divergence. Finally, while the results in Table 3 suggest that the reconstruction loss, $\mathcal{L}_{rec}$, does not contribute much to the model, during our experiment we noted that it plays an important role earlier during training and helps in faster and smoother convergence.

Recipe1M+ pretraining. Additionally, we include a comparison of the pretraining phase performance on the Recipe1M+ text dataset in Table 4. In the *single (S)* prediction setting, our performance is on-par, or slightly sub-par, with the BASELINE, which suggests that when training on a large textual dataset, it might be beneficial to learn the text encoder from scratch instead of using the pre-trained UniVL encoder (though the opposite is true for testing on video). However, GEPSAN outperforms the baseline in the *(M)* prediction setting even with the sub-optimal text encoder, which shows that our model can capture the multi-modal nature of the task under such settings as well. Notably, if the model is trained from scratch on YouCookII, the performance collapses, which shows the importance of the pretraining phase given data scarcity in the video domain, and the difficulty of training a generative model on such a small dataset.

Model	Recipe1M+ (Textual)				
	ING	VERB	B1	B4	MET
GEPSAN *(S)*	27.2	28.5	25.9	7.5	11.2
GEPSAN *(M)*	37.2	36.2	32.2	10.7	14.6
w/o $\mathcal{L}_{aux}$ *(S)*	26.8	27.9	23.9	7.4	10.9
w/o $\mathcal{L}_{aux}$ *(M)*	29.4	29.2	25.4	8.2	11.6
w/o $\mathcal{L}_{pred}$ *(S)*	25.7	29.0	25.9	5.2	11.0
w/o $\mathcal{L}_{pred}$ *(M)*	34.0	35.4	33.2	7.7	13.8
w/o $\mathcal{L}_{rec}$ *(S)*	27.7	28.3	25.7	7.3	11.0
w/o $\mathcal{L}_{rec}$ *(M)*	36.6	36.5	32.2	10.8	14.6

Table 3. **Ablation Study for the text-based future anticipation on Recipe1M+.** We assess contribution of the individual training objectives during the model pretraining phase. We report results for single *(S)* and multiple *(M)* next step prediction. To achieve single and multiple predictions, we evaluate GEPSAN using latent $z_{t+1} = 0$ (*i.e.*, mean of a Gaussian prior) and five random $z_{t+1} \sim \mathcal{N}(0, I)$, respectively.

Model	ING	VERB	B1	B4	MET
BASELINE *(S)*	27.0	29.4	24.1	7.8	11.3
GEPSAN *(S)*	27.2	28.5	25.9	7.5	11.2
BASELINE *(M)* $^\diamond$	34.7	34.6	31.7	9.4	14.2
GEPSAN *(M)*	**37.2**	**36.2**	**32.2**	**10.7**	**14.6**

Table 4. **Text-based future anticipation results on Recipe1M+.** We compare our results with the baseline [24] results for single *(S)* and multiple *(M)* next step prediction. To achieve single and multiple predictions, we evaluate GEPSAN using latent $z_{t+1} = 0$ (*i.e.*, mean of a Gaussian prior) and five random $z_{t+1} \sim \mathcal{N}(0, I)$, respectively. $^\diamond$We use *Nucleus sampling* [14] to achieve multiple predictions from the deterministic baseline.

5. Conclusion

In this work, we have addressed the problem of next step prediction from instructional videos (focusing on cooking activities). In particular, we have proposed GEPSAN, a generative next step prediction model that concentrates on capturing the uncertainty inherent to the task of future step anticipation, which was largely overlooked in previous work tackling this task in realistic open-world setting (*i.e.*,

not relying on a predefined closed set of step labels). In addition, we showed that GEPSAN can effectively capture multiple feasible future realizations, and outperforms existing baselines on video anticipation, with or without domain-specific adaptation, *i.e.*, zero-shot, thanks to the judicious use of aligned modality representation. We hope that this work will open up new avenues for future research that automatically considers multiple possible future realizations in open world next step prediction, with datasets and metrics that better support evaluation under these settings.

References

[1] Jing Bi, Jiebo Luo, and Chenliang Xu. Procedure planning in instructional videos via contextual modeling and model-based policy learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[2] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 10–21, 2016. 4

[3] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure planning in instructional videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[4] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[5] Y. A. Farha, B. Schiele Q. Ke, and J. Gall. Long-term anticipation of activities with cycle consistency. In *arXiv:2009.01142*, 2020. 2

[6] A. Furnari and G. M. Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling LSTMs and modality attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[7] Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. Jointly optimizing diversity and relevance in neural response generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1229–1238, 2019. 3

[8] Sayontan Ghosh, Tanvi Aggarwal, Minh Hoai, and Niranjan Balasubramanian. Text-derived knowledge helps vision: A simple cross-modal distillation for video-based action anticipation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[9] Rohit Girdhar and Kristen Grauman. Anticipative Video Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[10] Dayoung Gong, Joonseok Lee, Manjin Kim, Seong Jong Ha, and Minsu Cho. Future transformer for long-term action anticipation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3052–3061, 2022. 2

[11] Anirudh Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. Z-forcing: Training stochastic recurrent networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 3, 5

[12] Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. In *International Conference on Learning Representations (ICLR)*, 2019. 3

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 11

[14] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020. 6, 7, 8, 11

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[16] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. UniVL: A unified video and language pre-training model for multimodal understanding and generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3

[17] Javier Marín, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. Recipe1M+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019. 2, 5

[18] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[19] Megha Nawhal, Akash Abdu Jyothi, and Greg Mori. Rethinking learning approaches for long-term action anticipation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 558–576. Springer, 2022. 2

[20] Y. B. Ng and B. Fernando. Forecasting future action sequences with attention: a new approach to weakly supervised action forecasting. 29, 2020. 2

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, 2002. 6, 11, 12

[22] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark et al. Learning transferable visual models from natural language

supervision. In *International Conference on Machine Learning (ICML)*, 2021. 3

[24] Fadime Sener, Rishabh Saraf, and Angela Yao. Transferring knowledge from text to video: Zero-shot anticipation for procedural actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022. 1, 2, 3, 5, 6, 7, 8, 11, 12

[25] Fadime Sener and Angela Yao. Zero-shot anticipation for instructional activities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3, 5, 6, 11

[26] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7464–7473, 2019. 3

[27] Jiankai Sun, De-An Huang, Bo Lu, Yun-Hui Liu, Bolei Zhou, and Animesh Garg. Plate: Visually-grounded planning with transformers in procedural tasks, 2021. 2

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 2, 3

[29] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 835–851. Springer, 2016. 3

[30] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6787–6800, 2021. 3

[31] Xinyu Xu, Yong-Lu Li, and Cewu Lu. Learning to anticipate future with dynamic context removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12734–12744, 2022. 2

[32] Z. Yang, J. Liu, J. Huang, X. He, T. Mei, C. Xu, and J. Luo. Cross-modal contrastive distillation for instructional activity anticipation. In *International Conference on Pattern Recognition (ICPR)*, pages 5002–5009, 2022. 2, 3, 11

[33] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[34] Olga Zatsarynna, Yazan Abu Farha, and Juergen Gall. Multi-modal temporal convolutional network for anticipating actions in egocentric videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2249–2258, 2021. 2

[35] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 649–666, 2016. 2

[36] He Zhao, Isma Hadji, Nikita Dvornik, Konstantinos G. Derpanis, Richard P. Wildes, and Allan D. Jepson. P3iv: Probabilistic procedure planning from instructional videos with weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[37] He Zhao and Richard P. Wildes. On diverse asynchronous activity anticipation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 3

[38] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017. 3, 5

[39] Zeyun Zhong, David Schneider, Michael Voit, Rainer Stiefelhagen, and Jürgen Beyerer. Anticipative feature fusion transformer for multi-modal action anticipation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6068–6077, 2023. 2

[40] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598, 2018. 2, 6, 12