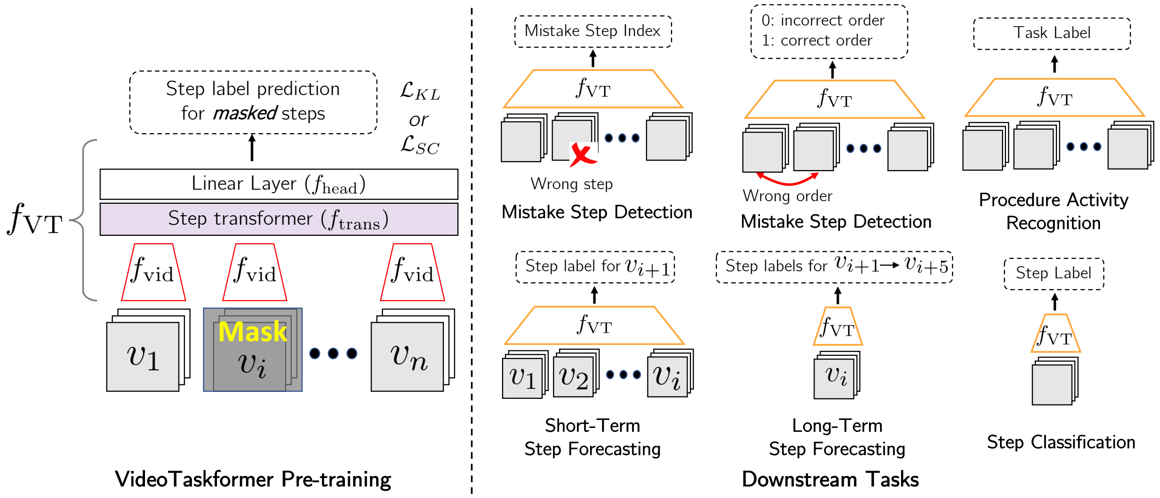

Figure 2: **VideoTaskformer Pre-training (Left).** VideoTaskformer f_{VT} learns step representations for the masked out video clip v_i, while attending to the other clips in the video. It consists of a video encoder f_{vid}, a step transformer f_{trans}, and a linear layer f_{head}, and is trained using weakly supervised step labels. **Downstream Tasks (Right).** We evaluate step representations learned from VideoTaskformer on 6 downstream tasks.

ponent of the task. For example, for the task "*Making a french toast*", examples of steps include "*Whisk the batter*", and "*Dip bread in batter*." We train a video model VideoTaskformer f_{VT} to learn step representations. We mask out a few clips in the input V and feed it to f_{VT} which learns to predict step labels for the masked-out clips. We evaluate the embeddings learned by our pre-training objective on 6 downstream tasks: step classification, procedural activity recognition, step forecasting, mistake step detection, mistake ordering detection, and long term forecasting.

Below, we provide more details on how we pre-train VideoTaskformer using a masked step modeling loss, followed by fine-tuning details on the downstream tasks.

3.1. Pre-training VideoTaskformer with Masked Step Modeling

We extend masked language modeling techniques used in BERT and VideoBERT to learn step representations for instructional videos. While BERT and VideoBERT operate on language and visual tokens respectively, VideoTaskformer operates on clips corresponding to steps in an instructional video. By predicting weakly supervised natural language step labels for masked out clips in the input video, VideoTaskformer learns semantics and long-range temporal interactions between the steps in a task. Unlike prior works wherein step representations are learned from local short video snippets corresponding to the step, our step representations are from the entire video with all the steps as input and capture *global context* of the video.

Masked Step Modeling. Let $V = \{v_1, \ldots, v_K\}$ denote the visual clips corresponding to K steps in video V. The goal of our our Masked Step Modeling pre-training setup is to encourage VideoTaskformer to learn representations of clips v_i that are aware of the semantics of the corresponding step and the context of the surrounding task. To this end, the task for pre-training is to predict categorical natural language step labels for the masked out steps. While we do not have ground truth step labels, we use the weak supervision procedure proposed by [13] to map each clip v_i to a distribution over step labels $p(y_i \mid v_i)$ by leveraging the noisy ASR annotations associated with each clip. The distribution $p(y_i \mid v_i)$ is a categorical distribution over a finite set of step labels Y. More details are provided in Sec. 3.3.

Let $M \subseteq [1, \ldots, K]$ denote some subset of clip indices (where each index is included in M with some masking probability r, a hyperparameter). Let $V_{\setminus M}$ denote a partially masked-out sequence of clips: the same sequence as V except with clips v_i masked out for all $i \in M$.

Let f_{VT} represent our VideoTaskformer model with parameters θ. f_{VT} is composed of a video encoder model f_{vid} which encodes each clip v_i independently, followed by a step transformer f_{trans} operating over the sequence of clip representations, and finally a linear layer f_{head} (which includes a softmax). The input to the model is an entire video (of size $K \times L \times H \times W \times 3$) and the output is of size $K \times S$ (where S is the output dimension of the linear layer).

We pre-train f_{VT} by inputting a masked video $V_{\setminus M}$ and predicting step labels y_i for each masked-out clip v_i, as described below. For the downstream tasks, we extract step-aware representations using f_{VT} by feeding an unmasked video V to the model. We then extract the intermediate outputs of f_{trans} (which are of size $K \times D$, where D is the