Learning to Abstract and Predict Human Actions

Romero Morais
ralmeidabaratad@deakin.edu.au

Vuong Le
vuong.le@deakin.edu.au

Truyen Tran
truyen.tran@deakin.edu.au

Svetha Venkatesh
svetha.venkatesh@deakin.edu.au

Applied Artificial Intelligence Institute
Deakin University
Australia

Abstract

Human activities are naturally structured as hierarchies unrolled over time. For action prediction, temporal relations in event sequences are widely exploited by current methods while their semantic coherence across different levels of abstraction has not been well explored. In this work we model the hierarchical structure of human activities in videos and demonstrate the power of such structure in action prediction. We propose Hierarchical Encoder-Refresher-Anticipator, a multi-level neural machine that can learn the structure of human activities by observing a partial hierarchy of events and roll-out such structure into a future prediction in multiple levels of abstraction. We also introduce a new coarse-to-fine action annotation on the *Breakfast Actions* videos to create a comprehensive, consistent, and cleanly structured video hierarchical activity dataset. Through our experiments, we examine and rethink the settings and metrics of activity prediction tasks toward unbiased evaluation of prediction systems, and demonstrate the role of hierarchical modeling toward reliable and detailed long-term action forecasting.

1 Introduction

An AI agent that shares the world with us needs to efficiently anticipate human activities to be able to react to them. Moreover, the ability to anticipate human activities is a strong indicator of the competency in human behavior understanding by artificial intelligence systems. While video action recognition [3] and short-term prediction [11] have made much progress, reliable long-term anticipation of activities remains challenging [1] as it requires deeper understanding of the action patterns.

The most successful methods for activity prediction rely on modeling the continuity of action sequences to estimate future occurrence by neural networks [7, 11]. However, these networks only consider the sequential properties of the action sequence which tends to fade and entice error accumulation in far-term. This issue suggests exploring the abstract structure of actions that spans over the whole undertaking of the task. One intuitive way to approach this path is to follow the natural human planning process that starts with high level tasks then
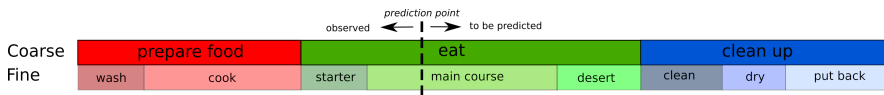
Figure 1: Illustration of a two-level structure of activity "have dinner" and a prediction task.

proceeds to more refined sub-tasks and detailed actions [2]. An example of such structure in an activity is shown in Fig. 1. Our quest is to build a neural machine that can learn to explore such structures by observing a limited section of the video and extrapolate the activity structure into the future for action prediction.

We realize this vision by designing a neural architecture called Hierarchical Encoder-Refresher-Anticipator (HERA) for activity prediction. HERA consists of three sub-networks that consecutively encode the past, refresh the transitional states, and decode the future until the end of the overall task. The specialty of these networks is that their layers represent semantic levels of the activity hierarchy, from abstract to detail. Each of them operates on its own clock while sending its state to parent layer and laying out plans for its children. This model can be trained end-to-end and learn to explore and predict the hierarchical structure of new video sequences. We demonstrate the effectiveness of HERA in improved long-term predictions, increased reliability in predicting unfinished activities, and effective predictions of activities at different levels of granularity.

To promote further research in hierarchical activity structures, we also introduce a new hierarchical action annotation to the popular Breakfast Actions dataset [14]. These annotations contain two-level action labels that are carefully designed to reflect the clean hierarchy of actions following natural human planning. In numbers, it includes 25,537 annotations in two levels on 1,717 videos spanning 77 hours. Once publicly released, this dataset will provide a key data source to support advancing deep understanding into human behaviors with potential applications in detection, segmentation and prediction.

2 Related work

For prediction of actions in videos, the most popular approach is to predict the temporal action segments, by jointly predicting the action labels and their lengths [17]. Recent advances in this front include Farha et al. [2] where random prediction points are used with the RNN/CNN-like model. Moving away from recurrent networks which tend to accumulate errors, Ke et al. [10] used time point as the conditioning factor in one-shot prediction approach with the trade-off in high prediction cost and sparse predictions. While these methods work relatively well in near-term, when the actions are predicted farther into the future, uncertainty prevents them from having reliable results. Variational methods manage uncertainty by using probabilistic modeling to achieve more robust estimation of inter-arrival time [18] and action length [1].

As an action is highly indicative of the next action, Miech et al. [19] proposed a model that is a convex combination of a "predictive" model and a "transitional" model. A memory-based approach network was proposed by Gammulle et al. [8], in which two streams with independent memories analyze visual and label features to predict the next action.

The hierarchy of activities can be considered in atomic scales where small movements constitute an action [15]. Early works investigated the hierarchy of activity through layered HMM [6], layered CRF [23], and linguistic-like grammar [21]. More recent works favor neural networks due to their strong inductive properties [7, 10]. For hierarchy, Recurrent Neural Networks (RNN) can be stacked up, but stacking ignores the multi-clock nature of a

hierarchy unrolled over time. In [25], a hierarchical RNN with asynchronous clocks was used to model the temporal point processes of activity but the information only passes upward and multi-level semantics of events are not explored. The idea of multi-clocks was also explored by Hihi and Bengio [9] and Koutnik *et al.* [13]. The drawback of these methods is that the periods of the clock must be manually defined, which is not adaptive to data structure at hand. Chung *et al.* [5] addressed this problem with a hierarchical multi-scale RNN (HM-RNN), which automatically learns the latent hierarchical structure. This idea has been extended with attention mechanism for action recognition [24]. Our hierarchical modeling shares the structure exploration functionality with these works but is significantly different in the ability to learn the semantic-rich structures where layers of hierarchy are associated with levels of activity abstraction. In particular, in comparison with Clock-work RNN (CW-RNN) [13], HERA shares the fact that units can update at different rates, but HERA is significantly different to CW-RNN in separating the levels of RNN with distinctive associated semantics. HERA also allows RNN units to control their own clocks and their interactions with other units.

3 Learning to abstract and predict human actions

3.1 Problem formulation

We formalize an activity hierarchy H of L levels of a human performing a task observable in a video as $H = \left\{ A^l \right\}_{l=1,2,\ldots,L}$ where each level A^l is a sequence of indexed actions:

$$A^l = \left\{ \left(x_k^l, d_k^l \right) \right\}_{k=1,2,\ldots,n_l}. \tag{1}$$

Here, x_k^l represents the label of k-th action at the l-th level, d_k^l is its relative duration calculated as its portion of the parent activity, and n_l indicates the number of actions at level l. Each action $\left(x_k^l, d_k^l \right)$ is associated with a subsequence of finer actions at level $l+1$, and the latter are called children actions of the former. Any children subsequence is constrained to exclusively belong to only one parent activity[1].

In the special case of a hierarchy with two levels, members of the first level represent *coarse activities*, and those at the second level are called *fine actions*. In this case, we will extend the notation to use the level indices c - for coarse and f - for fine in place of numeric indices $l = 1$ and $l = 2$. An example of a two-level hierarchy is shown in Fig. 1, where for a *task* of <have-dinner>, the first coarse activity <prepare-food> contains three fine actions as children.

Under this structure, the prediction problem is formed when the hierarchy of activities is interrupted at a certain time t^* indicating the point where observation ends. At this time, at every level we have finished events, unfinished events, and the task is to predict events yet to start. The given observation includes the labels and lengths of the finished events, and the labels and *partial lengths* of the unfinished ones. Thus the task boils down to estimating the remaining lengths of the unfinished events, and all details of the remaining events.

3.2 Hierarchical Encoder-Refresher-Anticipator

We design HERA (Fig. 2) to natively handle the hierarchical structure of observation and extend such structure to prediction. HERA has three components: the Encoder, the Refresher,

---

[1]Note that a child action label can be performed by multiple parents at different parent times. See Sec. 3.3.
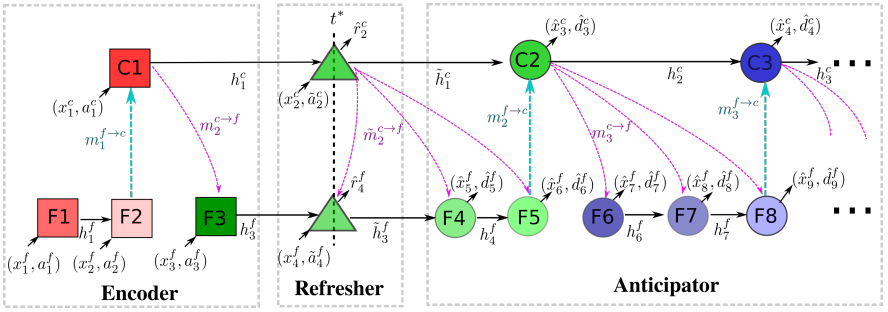
Figure 2: The Hierarchical Encoder-Refresher-Anticipator (HERA) architecture realized in a particular event sequence similar to the one in Fig. 1. Square blocks are Encoder GRU cells, while triangles and circles are those of Refresher and Anticipator, respectively. Color shades indicate cells processing different activity families, e.g., the first coarse cell (red C1) and its two children (fading red F1 and F2) process the first activity family $\{(x_1^c, d_1^c), (x_1^f, d_1^f), (x_2^f, d_2^f)\}$. The prediction point t^* happens at the middle of (x_2^c, d_2^c) and (x_3^f, d_3^f). Black arrows indicate recurrent links while those in pink and cyan are for downward and upward messages, respectively. For visual clarity, optional prediction outputs of Encoder cell and feedback inputs of Anticipator cell are omitted.

and the Anticipator. The Encoder creates a multi-level representation of the observed events which is used by the Refresher and Anticipator to roll-out in a similar manner. The Encoder and Anticipator share the same hierarchical model design for cross-level interaction which we detail next.

Modeling activity hierarchy. The Encoder and Anticipator share an identical architecture of two layers of recurrent neural units (RNN) which are chosen to be based on Gated Recurrent Units (GRU) [4]. The upper layer models the dynamics of coarse activities:

$$h_i^c = \text{GRU}\left(\left[(x_i^c, a_i^c), m_i^{f \to c}\right], h_{i-1}^c\right). \tag{2}$$

The first input to the unit includes a tuple of coarse label x_i^c and accumulated duration $a_i^c = \sum_{k=1}^i d_k^c$. Both x_i^c and a_i^c are encoded using a random embedding matrix. At the Anticipator, these inputs are feedback from the previous prediction step. The second input $m_i^{f \to c}$ is the upward message that will be discussed later.

The lower layer is another RNN that is triggered to start following the parent's operation:

$$h_j^f = \text{GRU}\left(\left[(x_j^f, a_j^f), m_i^{c \to f}\right], h_{j-1}^f\right), \tag{3}$$

where the proportional accumulated duration a_j^f is calculated within the parent activity.

By design, the two layers are asynchronous (i.e. the layers update their hidden state independently and whenever fit) as coarse activities happen sparser than fine actions. A key feature of HERA is the way it connects these two asynchronous concurrent processes in a consistent hierarchy by using the cross-level messages. The downward message $m_i^{c \to f}$ (pink arrows in Fig.2) provides instructions from the previous coarse cell to the current fine cells. This message contains the previous coarse hidden state h_{i-1}^c and can optionally contain the parent's predicted label $\hat{x}_i^c$. The upward message $m_i^{f \to c}$ (cyan arrows) to a coarse node i

from its children contains the information about the detail roll-out in the fine actions. It is implemented as the hidden state of the last child.

Making predictions. At each step of both levels, the hidden state of the current cell is used to infer the label and duration of the next action through multi-layer perceptrons (MLP):

$$\left(\hat{x}_{i+1}^c, \hat{d}_{i+1}^c\right) = \text{MLP}\left(h_i^c\right); \qquad \left(\hat{x}_{j+1}^f, \hat{d}_{j+1}^f\right) = \text{MLP}\left(h_j^f\right) \tag{4}$$

For the Anticipator, these predictions are rolled out until accumulated relative duration reaches 100%. At the fine level, this means the end of the parent coarse activity while at the coarse level it means the overall task is finished. At the Encoder, the predictions are optionally used in training and are discarded in testing.

Connecting the past and the present to the future. The connection between Encoder and Anticipator happens at the interruption point t^*, where the observed hierarchy ends and prediction starts. If t^* is well aligned with the action boundary, we can simply pass the last hidden states and predictions of the Encoder to the Anticipator at the corresponding levels. However, these coincidences are rare; in most cases, the interruption happens at the middle of an action and leaves trailing unfinished activity and actions at different stages.

To connect this gap, we design the Refresher which consists of a pair of connected MLP cells (Triangle blocks in Fig. 2). The coarse Refresher cell gathers all available data and predicts the remaining length $\hat{r}_{i^*}^c$ of interrupted coarse activity i^*:

$$\tilde{h}_{i^*-1}^c = \text{MLP}\left(\left[h_{i^*-1}^c, x_{i^*}^c, \tilde{a}_{i^*}^c, \tilde{d}_{i^*}^c\right]\right); \qquad \hat{r}_{i^*}^c = \text{MLP}\left(\tilde{h}_{i^*-1}^c\right), \tag{5}$$

where $\tilde{d}_{i^*}^c$ and $\tilde{a}_{i^*}^c$ are unfinished duration and accumulated duration, respectively.

The remaining fine action duration $\hat{r}_{j^*}^f$ is estimated similarly, but with the downward message as additional input:

$$\tilde{h}_{j^*-1}^f = \text{MLP}\left(\left[h_{j^*-1}^f, x_{j^*}^f, \tilde{a}_{j^*}^f, \tilde{d}_{j^*}^f, \tilde{m}_i^{c \to f}\right]\right); \qquad \hat{r}_{j^*}^f = \text{MLP}\left(\tilde{h}_{j^*-1}^f\right). \tag{6}$$

Effectively, the overall predicted duration of the interrupted action is amended:

$$\hat{d}_{i^*}^c = \tilde{d}_{i^*}^c + \hat{r}_{i^*}^c; \qquad \hat{d}_{j^*}^f = \tilde{d}_{j^*}^f + \hat{r}_{j^*}^f. \tag{7}$$

After these refreshing steps, the hidden states $\tilde{h}_{i^*}^c$ and $\tilde{h}_{i^*}^f$ are passed to the Anticipator as the initial states, where the hierarchical prediction is rolled out further.

Model training. In HERA's end-to-end training, we calculate the loss at each level l (among coarse and fine) and each stage $\star$ (among the Encoder, Refresher, and Anticipator) as a weighted sum of negative log-likelihood loss (NLL) on predicted labels and mean squared error (MSE) on predicted durations (for the Refresher we only have the MSE loss):

$$\mathcal{L}_l^\star = \frac{1}{n_l} \sum_{k=1}^{n_l} \left[\lambda_{\text{label}}^\star \text{NLL}\left(\hat{x}_k^l, x_k^l\right) + \lambda_{\text{duration}}^\star \text{MSE}\left(\hat{d}_k^l, d_k^l\right)\right], \tag{8}$$

and the total loss for HERA is a sum of the losses in all layers and stages:

$$\mathcal{L} = \sum_{l=1}^{L} \left[\mathcal{L}_l^E + \mathcal{L}_l^R + \mathcal{L}_l^A\right]. \tag{9}$$

The Encoder loss $\mathcal{L}_l^E$ is for regularizing the Encoder and is optional. The weights $\lambda_-^\star$ are estimated together with the network parameters by using a multi-task learning framework similar to that of Kendall *et al*. [11].

For model validation, we use the videos of a single person from each cross-validation split. We train HERA for 20 epochs and select the weights from the epoch with the lowest validation loss. We use the ADAM [12] optimizer with a learning rate of 10^{-3} and a batch size of 512. The selected hidden size for the GRUs and MLPs was 16. We used the PyTorch [20] framework for implementation of HERA.

3.3 Data annotation

To support the problem structure formulated above we reannotated the Breakfast Actions videos [14], which is the largest multi-level video activity dataset publicly available. This dataset contains footage of 52 people preparing 10 distinct breakfast-related dishes, totaling 1,717 videos. It originally contains fine- and coarse-level annotations of the actions but the hierarchy is incoherent (inconsistent semantic abstraction), incomplete (only 804 of the videos have fine-level annotations), and statistically weak (many fine actions are less than a few frames).

We employed two annotators working independently on all 1,717 videos and one verifier who checked the consistency of the annotations. Following the hierarchy definition in Sec. 3.1, we annotated a two-level hierarchy of *coarse activities* and *fine actions*. Each label of activity or action follows the format of `<verb-noun>` where verbs and nouns are selected from a predefined vocabulary. The two vocabulary sets were built by a pilot round of annotation. The coarse activities can share the fine action labels. For instance, `<add-salt>` fine action label can be used for many coarse activities including `<make-salad>`, `<fry-egg>`, and `<make-sandwich>`. In actual annotation, we have 30 `<verb-noun>` pairs for coarse activities and 140 for fine actions that are active. The new annotation resulted in a total of 25,537 label-duration annotations with 6,549 at the coarse level and 18,988 at the fine level. We call the new annotation *Hierarchical Breakfast* dataset and it is available for download at https://github.com/RomeroBarata/hierarchical_action_prediction, alongside the source code for HERA.

3.4 Metrics

Recent action prediction works [7, 10] widely used mean-over-class (MoC) as the key performance metric. However, MoC is susceptible to bias in class imbalance which exists in action prediction datasets. More importantly, as any frame-based metrics, it merits any correctly predicted frames even when the predicted segments are mostly unaligned due to under- or over-segmentation. We verified these conceptual problems by setting up an experiment (detailed in Sec. 4) using an under-segmenting dummy predictor that takes advantage of the flaw of the metric and win over state-of-the-art methods on many settings. We call our dummy predictor "under-segmenting" because it predicts that the future consists simply of one single long action.

In the search for better metrics, we examined options including the segmental edit distance, the mean-over-frame (MoF), and the F1@k. Among them, we learned that the most suitable metric for the purpose of action prediction is the F1@k for its robustness to variation in video duration and minor shifts caused by annotation errors. Furthermore, it penalizes both over- and under-segmentations such as from our dummy predictor. This metric was previously

used for temporal detection and segmentation [16]. Applied to the prediction task, we first calculate the intersection over union (IoU) of the predicted segments with the ground-truth. Any overlapping pair with IoU surpassing the chosen threshold $0 < k < 1$ is counted as correct when contributing to the final $F1 = {2 \times \text{Prec} \times \text{Recall}}/{\text{Prec} + \text{Recall}}$.

4 Experiments

4.1 Experiment settings

We setup experiments following the common settings in which 20% or 30% of the videos are observed and the prediction is done on the remaining portion (70% or 80%). We also follow previous convention to use annotated labels of the observed portion to be the input, with the assumption that in practical applications these labels can be reliably provided by action detection engines [9]. All experiments are done with 4-fold cross–validation as in previous works [7, 10].

We use the new *Hierarchical Breakfast* (see Sec. 3.3) as our principal source of data for its most comprehensive multi-level activities. Besides this one, the 50 Salads dataset [22] also has two-level annotations and was used in several previous works [7, 10]. However per acquisition procedure description [22] and through our independent examination, we concluded that 50 Salads is only suitable for action detection and segmentation but not for action prediction because of the randomness in scripted action sequences. Such scripts were generated from an artificial statistical model that intentionally introduces random sequences instead of following natural human behaviors. This makes most non-trivial methods converge to similar prediction results (reported in the supplementary material), and hence is not suitable for differentiating their performances.

4.2 Metrics and data assessment

We set up experiments to demonstrate the drawback of MoC and verify the robustness of F1@k described in Sec. 3.4. We use a dummy predictor that simply predicts that the interrupted action goes on for the rest of the video, i.e., the extreme under-segmentation. We compare this dummy predictor to the results reported by two top performing predictors by Farha *et al*. [7] and Ke *et al*. [10][2] at the coarse-level of the original Breakfast Actions dataset. As results in Table 1 show, the dummy predictor performs comparable to the best and usually outperforms one or both of the methods in MoC by exploiting its fragility toward over- and under- segmentation.

When we replace MoC with our chosen F1@k metric (Table 2), the dummy predictor only has good scores at the immediate 10% prediction (as designed) and marked down significantly afterward as continuing action no longer matches with the actual events.

4.3 Predicting activity hierarchy

In this section, we validate the performance of our proposed HERA against reference methods. As predicting the roll-out of activity hierarchy is a new task, we implemented several baselines and adapted a state-of-the-art method by Farha *et al*. [7] to work with two-level activities.

---

[2]We could neither obtain nor reproduce the implementation of Ke *et al*. [10], therefore we could only use the reported performance on the original Breakfast annotation and MoC metrics (last row of Table 1).

Table 1: Mean-over-class (MoC) scores on the coarse-level of the original Breakfast Actions dataset. The dummy predictor matches performance with state-of-the-art methods, which demonstrates the weakness of the MoC metric.

Observe	20%				30%			
Predict	10%	20%	30%	50%	10%	20%	30%	50%
Dummy	0.64	0.51	0.44	0.35	0.68	0.54	0.44	0.36
Farha et al. [7]	0.60	0.50	0.45	0.40	0.61	0.50	0.45	0.42
Ke et al. [10]	0.64	0.56	0.50	0.44	0.66	0.56	0.49	0.44

Table 2: F1@0.25 scores on the coarse-level of the original Breakfast Actions dataset. The F1@0.25 metric is robust to the dummy predictor and helps better methods to stand out in long-term predictions.

Observe	20%				30%			
Predict	10%	20%	30%	50%	10%	20%	30%	50%
Dummy	0.77	0.61	0.51	0.34	0.80	0.67	0.56	0.40
Farha et al. [7]	0.76	0.68	0.64	0.58	0.78	0.71	0.68	0.64

All baselines accept observed labels and (accumulated) durations and output those of future events. The first baseline, *Independent-Single-RNN,* uses two separate GRUs for coarse activities and fine actions hence does not consider the correlation between the two levels. To take into account this correlation, *Joint-Single-RNN*, the second baseline, models the joint distribution of the two processes by concatenating input from both levels and predicting them together. The third baseline, *Synced-Pair-RNN*, is more sophisticated and has two parallel GRUs for the two levels operating at the same clock, which communicate regularly by one-way coarse-to-fine messages. Because the last two baselines operate with a single recurrent clock on two signals that are not synchronized, coarse level inputs are repeated as needed to sync-up with the fine level counterparts.

The original Farha et al.'s model [7] (denoted as "Farha") only accepts a single level of observed action as input, hence two separated instances of it are used to predict at coarse and fine level. To make consistent competition, we extend this method to accept hierarchical input by jointly observing and predicting the two levels of actions (named "Farha2").
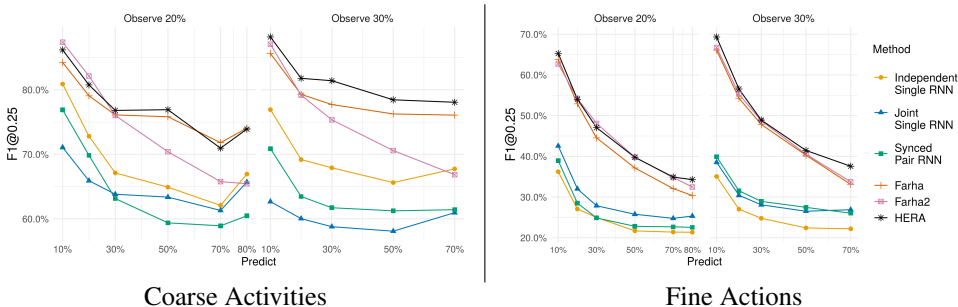

Figure 3: F1@0.25 performance of HERA and related baselines on **coarse** (left fig.) and **fine** (right fig.) levels of Hierarchical Breakfast dataset.
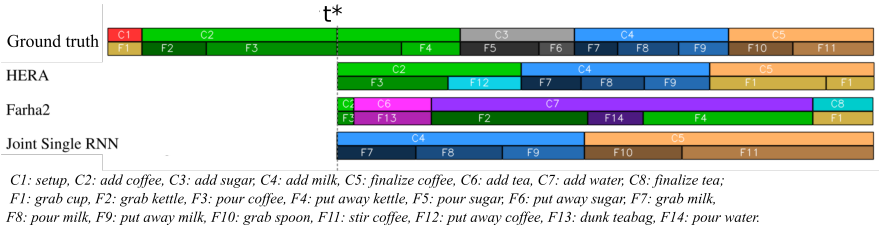
C1: setup, C2: add coffee, C3: add sugar, C4: add milk, C5: finalize coffee, C6: add tea, C7: add water, C8: finalize tea;
F1: grab cup, F2: grab kettle, F3: pour coffee, F4: put away kettle, F5: pour sugar, F6: put away sugar, F7: grab milk,
F8: pour milk, F9: put away milk, F10: grab spoon, F11: stir coffee, F12: put away coffee, F13: dunk teabag, F14: pour water.

Figure 4: Qualitative evaluation of predictions of different methods on task "make coffee". The first timeline shows the observed and ground-truth future. Others show future predictions of corresponding methods.

We compare the performance of HERA and aforementioned reference models on Hierarchical Breakfast. The F1@0.25 scores on two settings are shown in Fig. 3. The performance results suggest notable patterns. First of all, we learned that coarse activities and fine actions have strong but asymmetrical correlation. When modeled in a joint distribution, the fine action prediction (right subfigure) is improved over independent modeling (Joint-Single-RNN to Independent-Single-RNN, and Farha 2 to Farha); meanwhile coarse prediction (left subfigure) is degraded by the noise introduced in the over-detailed data from the fine channel.

Secondly, modeling coarse and fine as parallel interactive concurrent processes (Synced-Pair-RNN) may help especially in encoding long observation. However the naïve synchronicity between the two processes is unreliable and in many cases significantly hurt the performance. Thirdly, when introducing structure to the cross-process interaction (HERA), the performance significantly improved both near- and far- term and across coarse- and fine-channels. This result suggests that appropriate structured modeling is key to deeply observe, understand and generate hierarchical activity patterns. Fourthly, in longer term prediction, asynchronously clocked RNNs (as in HERA) alleviate the error accumulation issues persistent in all other synchronous RNN based models.

Overall, HERA attained higher prediction accuracy in relation to other methods, especially in harder cases such as on far-term fine actions. To further understand this improvement, we visualize the predictions of HERA and compare them with those of other methods. One example is shown in Fig. 4 and more are included in the supplementary material. In this case, HERA predicts most accurately the remaining duration of the unfinished actions thanks to the special design of the Refresher. Furthermore, the strong coarse-to-fine structure helps it recover from mistakes while maintaining cross-hierarchy consistency. Without this structure, other methods tend to roll-out on top of the mistakes and cannot recover. They also sometimes allow ill-formed hierarchy such as the pink `C6-F13` family in Farha2.

We argued that the MoC metric is not appropriate for the problem at hand, but we report it next for transparency and completeness. For observing 20% and predicting the next 10%/20%/30%/50%/70%/80%, HERA attained an MoC of 0.77/0.68/0.57/0.51/0.51/0.57 for the coarse level, and 0.42/0.31/0.26/0.23/0.21/0.21 for the fine level; Farha2 attained an MoC of 0.76/0.67/0.57/0.51/0.50/0.52 for the coarse level, and an MoC of 0.39/0.29/0.26/0.22/0.21/ 0.20 for the fine level. For observing 30%, and predicting the next 10%/20%/30%/50%/70%, HERA attained an MoC of 0.77/0.62/0.59/0.53/0.61 for the coarse level, and an MoC of 0.44/0.33/0.28/0.25/0.23 for the fine level, whereas Farha2 attained an MoC of 0.76/0.64/0.58/ 0.52/0.55 for the coarse level, and an MoC of 0.41/0.32/0.28/0.24/0.22 for the fine level. HERA's MoC is higher than Farha2's MoC in most cases, but as discussed earlier the F1@k

Table 3: F1@**0.25** scores of HERA and its variations on the fine-level of Hierarchical Breakfast dataset with 20% of the videos observed.

Variations	10%	20%	30%	50%	80%
W/o $\downarrow\uparrow$ msg	0.36	0.27	0.25	0.22	0.21
W/o label in $\downarrow$ msg	0.62	0.51	0.45	0.36	0.31
W/ dis. Refresher	0.64	0.53	0.45	0.37	**0.34**
Full HERA	**0.65**	**0.54**	**0.47**	**0.40**	**0.34**

metric should be preferred when comparing models for action prediction.

Ablation study.   To further understand the roles of components and design choices in HERA, we switch off several key aspects of the model and observe the change in performance (Table 3). The first variation without the two-way cross-level messaging suffers significant performance loss as the correlation between the two channels is ignored. The second variation lacks the explicit coarse label $\hat{x}^c_{i+1}$ in downward messages and slightly under performs as missing the direct strong "instruction" of the discrete coarse labels. Lastly, the third row provides evidence for the importance of the Refresher stage in wrapping up unfinished action consistently at all levels.

5   Conclusions

We have introduced HERA (Hierarchical Encoder-Refresher-Anticipator), a new hierarchical neural network for modeling and predicting the long-term multi-level action dynamics in videos. To promote further research we re-annotated from scratch 1,717 videos in the Breakfast Actions dataset, creating a new and complete semantically coherent annotation of activity hierarchy, which we named *Hierarchical Breakfast*. We also reassessed the commonly used MoC metric in action prediction, and found it unreliable for the task. As a result we investigated multiple metrics and found the F1@k metric to reflect human activity best among them. We demonstrated that our HERA naturally handles hierarchically structured activities, including interruptions in the observed activity hierarchy. When compared to related methods that do not exploit the hierarchical structure in human activities, or explore it in a sub-optimal way, HERA attained superior results specially in the long-term regime.

Acknowledgments

References

[1] Yazan Abu Farha and Juergen Gall. Uncertainty-Aware anticipation of activities. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1197–1204, October 2019. doi: 10.1109/ICCVW.2019.00151.

[2] Icek Ajzen. From intentions to actions: A theory of planned behavior. In *Action Control: From Cognition to Behavior*, pages 11–39. Springer Berlin Heidelberg, 1985. ISBN 9783642697463. doi: 10.1007/978-3-642-69746-3_2.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, July 2017. doi: 10.1109/CVPR.2017.502.

[4] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN Encoder–Decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. ACL, 2014.

[5] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net, April 2017.

[6] Thi Duong, Dinh Phung, Hung Bui, and Svetha Venkatesh. Efficient duration and hierarchical modeling for human activity recognition. *Artificial intelligence*, 173(7): 830–856, May 2009. ISSN 0004-3702. doi: 10.1016/j.artint.2008.12.005.

[7] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what? - anticipating temporal occurrences of activities. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5343–5352. IEEE, June 2018. doi: 10.1109/CVPR.2018.00560.

[8] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Forecasting future action sequences with neural memory networks. In *Proceedings of the British Machine Vision Conference 2019*. British Machine Vision Association, September 2019.

[9] Salah El Hihi and Yoshua Bengio. Hierarchical recurrent neural networks for Long-Term dependencies. In *Advances in Neural Information Processing Systems 8*, pages 493–499. MIT Press, 1996.

[10] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-Conditioned action anticipation in one shot. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9917–9926. IEEE, June 2019. ISBN 9781728132938. doi: 10.1109/CVPR.2019.01016.

[11] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, June 2018. doi: 10.1109/CVPR.2018.00781.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

[13] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork RNN. In Eric P Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1863–1871, Bejing, China, 2014. PMLR.

[14] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of Goal-Directed human activities. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 780–787, June 2014. doi: 10.1109/CVPR.2014.105.

[15] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In *Computer Vision – ECCV 2014*, pages 689–704. Springer International Publishing, 2014. doi: 10.1007/978-3-319-10578-9_45.

[16] Colin Lea, Michael D Flynn, René Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1012, July 2017. doi: 10.1109/CVPR.2017.113.

[17] Tahmida Mahmud, Mahmudul Hasan, and Amit K Roy-Chowdhury. Joint prediction of activity labels and starting times in untrimmed videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5784–5793, October 2017. doi: 10.1109/ICCV.2017.616.

[18] Nazanin Mehrasa, Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. A variational Auto-Encoder model for stochastic point processes. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3160–3169. IEEE, June 2019. ISBN 9781728132938. doi: 10.1109/CVPR.2019.00328.

[19] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2915–2922. IEEE, June 2019. ISBN 9781728125060. doi: 10.1109/CVPRW.2019.00351.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, High-Performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8026–8037. 2019.

[21] Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. Predicting human activities using stochastic grammar. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1173–1181, October 2017. doi: 10.1109/ICCV.2017.132.

[22] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738. ACM, September 2013. ISBN 9781450317702. doi: 10.1145/2493432.2493482.

[23] Truyen T Tran, Dinh Q Phung, Svetha Venkatesh, and Hung H Bui. AdaBoost.MRF: Boosted markov random forests and application to multilevel activity recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1686–1693, June 2006. doi: 10.1109/CVPR.2006.49.

[24] Shiyang Yan, Jeremy S Smith, Wenjin Lu, and Bailing Zhang. Hierarchical multi-scale attention networks for action recognition. *Signal Processing: Image Communication*, 61:73–84, February 2018. ISSN 0923-5965. doi: 10.1016/j.image.2017.11.005.

[25] Yatao Zhong, Bicheng Xu, Guang-Tong Zhou, Luke Bornn, and Greg Mori. Time perception machine: Temporal point processes for the when, where and what of activity prediction. *arXiv preprint arXiv:1808.04063*, 2018.