Appendix: Asynchronous Temporal Fields for Action Recognition

Gunnar A. Sigurdsson[1*] Santosh Divvala[2,3] Ali Farhadi[2,3] Abhinav Gupta[1,3]
[1]Carnegie Mellon University [2]University of Washington [3]Allen Institute for Artificial Intelligence
https://github.com/gsig/temporal-fields/

1. Appendix

This appendix contains the following additional content:

1. Description of the CRF.
2. Derivation of the update equations.
3. Details of the learning algorithm.
4. Additional implementation details.
5. Details about intent analysis.
6. Additional visualizations of output predictions.

1.1. Description of the CRF

We create a CRF which predicts activity, object, etc., for every frame in the video. For reasoning about time, we create a *fully-connected temporal CRF*, referred to as Asynchronous Temporal Field in the text. That is, unlike a linear-chain CRF for temporal modelling (the discriminative counterpart to Hidden Markov Models), each node depends on the state of every other node in the graph. We incorporate intention as another latent variable which is connected to all the action nodes.

In this work we encode multiple components of an activity. Each video with T frames is represented as $\{X_1, \dots, X_T, I\}$ where X_t is a set of frame-level random variables for time step t and I is a random variable that represent global intent in the entire video. As discussed in the paper, for clarity of derivation X_t includes all frame level variables $(C_t, O_t, A_t, P_t, S_t)$

Mathematically we consider a random field $\{X, I\}$ over all the random variables in our model ($\{X_1, \dots, X_T, I\}$). We now list the complete description of the CRF.

CRF Variables:

- Random field $\{X, I\} = \{X_1, \dots, X_T, I\}$
- Frame $X_t = \{C_t, O_t, A_t, P_t, S_t\}$, $X_t \in \mathcal{X}, \mathcal{X} = \mathcal{C} \times \mathcal{O} \times \mathcal{A} \times \mathcal{P} \times \mathcal{S}$
  - Category $C_t \in \mathcal{C}, \mathcal{C} = \{1, 2, ..., 157\}$ (For each category in the dataset)
  - Object $O_t \in \mathcal{O}, \mathcal{O} = \{1, 2, ..., 38\}$ (Includes "No object")
  - Action $A_t \in \mathcal{A}, \mathcal{A} = \{1, 2, ..., 33\}$
  - Progress $P_t \in \mathcal{P}, \mathcal{P} = \{1, 2, 3\}$ (Before, Middle, End)
  - Scene $S_t \in \mathcal{S}, \mathcal{S} = \{1, 2, ..., 15\}$
- Intent $I \in \mathcal{I}, \mathcal{I} = \{1, 2, ..., N_I\}$ ($N_I = 30$ in this work)

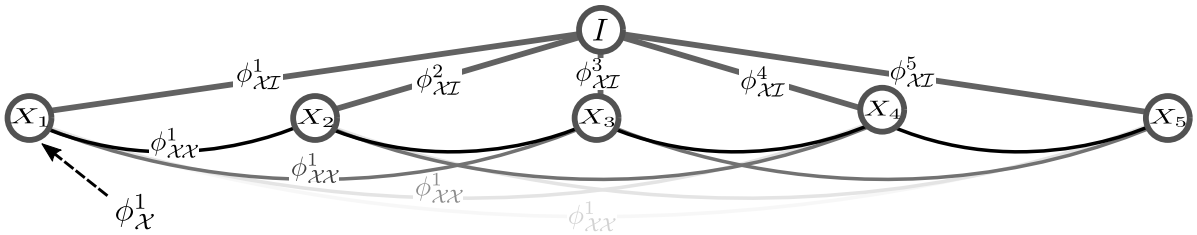

Figure 1. The model captures interactions between all frames X_t and the intent I, that is, a fully-connected model. Here shown for $T = 5$. We visualize some of the potentials of the model, and where they fit into the graph. All $\phi^i_{\mathcal{XI}}$ share the same parameters, but we calculate the gradients with respect for each of them separately below. For efficient inference, we use a mean-field approximation presented below. A mean-field approximation is a simpler distribution that is fit to the original distribution when needed.

CRF Potentials:

- $\phi_{\mathcal{X}} \colon \mathcal{X} \mapsto \mathcal{R}$, equivalently: $\phi_{\mathcal{X}} \colon \mathcal{C}\times\mathcal{O}\times\mathcal{A}\times\mathcal{P}\times\mathcal{S} \mapsto \mathcal{R}$
- $\phi_{\mathcal{X}}$ decomposes as follows: $\phi_{\mathcal{X}}(C_t, O_t, A_t, P_t, S_t)=\phi(O_t, P_t)+\phi(A_t, P_t)+\phi(O_t, S_t)+\phi(C_t, O_t, A_t, P_t)$
  - $\phi(O_t, P_t) \colon \mathcal{O}\times\mathcal{P} \mapsto \mathcal{R}$
  - $\phi(A_t, P_t) \colon \mathcal{A}\times\mathcal{P} \mapsto \mathcal{R}$
  - $\phi(O_t, S_t) \colon \mathcal{O}\times\mathcal{S} \mapsto \mathcal{R}$
  - $\phi(C_t, O_t, A_t, P_t) \colon \mathcal{B} \mapsto \mathcal{R}$, here $\mathcal{B}$ is all configurations of C_t, O_t, A_t, P_t that exist in the training data.
- $\phi_{\mathcal{XI}} \colon \mathcal{X}\times\mathcal{I} \mapsto \mathcal{R}$ (specifically we parametrize this as $\phi_{\mathcal{XI}} \colon \mathcal{O}\times\mathcal{I} \mapsto \mathcal{R}$)
- $\phi_{\mathcal{XX}} \colon \mathcal{X}\times\mathcal{X} \mapsto \mathcal{R}$ (specifically we parametrize this as $\phi_{\mathcal{XI}} \colon \mathcal{O}\times\mathcal{O} \mapsto \mathcal{R}$)

The complete distribution of the model is:

$$P(X, I) = \frac{1}{Z} \exp\left\{ \sum_i \phi^i_{\mathcal{X}}(x_i) + \sum_i \phi^i_{\mathcal{XI}}(x_i, I) + \sum_i \sum_{j\neq i} \phi^i_{\mathcal{XX}}(x_i, x_j) \right\} \tag{1}$$

where $\phi_{\mathcal{XX}}(x_i, x_j)$ is the potential between frame i and frame j, and $\phi_{\mathcal{XI}}(x_i, I)$ is the potential between frame i and the intent. For notational clarity $\phi_{\mathcal{X}}(x_i)$ incorporates all potentials for C_t, O_t, A_t, P_t, S_t. The model is presented in Figure 1.

1.2. Derivation of the Update Equations

Given an input video $V=\{V_1, \ldots, V_T\}$, our goal is to estimate the maximum a posteriori labeling of the random field by marginalizing over the intent I, $\sum_I P(X, I|V)$ as discussed in the paper. In the following derivations we omit the conditioning on V and write $P(X, I)$ and $\phi(X, I)$.

Before we present the update equations and gradients, we define the following messages which will be used in the final version of the following equations for clarity in their presentation. Messages are a term used for cached computations sent between different functions in a dynamic programming fashion. In the following derivations, X^* is used to explicitly denote the ground truth used for training. Plain X is used to refer to the variable.

Outgoing Messages (Messages that are calculated from a single frame)

$$FA_j(x_j) = E_{U\sim Q_j}\left[\mu(x_j, U)\right] \tag{2}$$
$$FB_j(x_j) = E_{U\sim Q_j}\left[\mu(U, x_j)\right] \tag{3}$$
$$H_j(I) = E_{U\sim Q_j}\left[\phi_{\mathcal{XI}}(U, I)\right] \tag{4}$$
$$H^*_j(I) = \phi_{\mathcal{XI}}(x^*_j, I) \tag{5}$$
$$K_j(x_j) = Q_j(x_j) \tag{6}$$
$$K^*_j(x_j) = \mathbf{1}_{x_j = x^*_j} \tag{7}$$

Incoming Messages (Messages that are calculated from messages from multiple frames and used for the computation of a single frame)

$$\mathbb{FA}_i(x_i) = \sum_{j>i} E_{U_j \sim Q_j}[\mu(x_i, U_j)]K(v_i, v_j) = \sum_{j>i} FA_j(x_i)K(v_i, v_j) \tag{8}$$

$$\mathbb{FB}_i(x_i) = \sum_{j<i} E_{U_j \sim Q_j}[\mu(U_j, x_i)]K(v_j, v_i) = \sum_{j<i} FB_j(x_i)K(v_j, v_i) \tag{9}$$

$$\mathbb{H}_i(I) = \sum_{j \neq i} E_{U_j \sim Q_j}\left[\phi_{\mathcal{XI}}(U_j, I)\right] = \sum_{j \neq i} H_j(I) \tag{10}$$

$$\mathbb{H}^*_i(I) = \sum_{j \neq i} \phi_{\mathcal{XI}}(x^*_j, I) = \sum_{j \neq i} H^*_j(I) \tag{11}$$

$$\mathbb{KA}_i(x_i) = \sum_{j>i} Q_j(x_j)K(x_i, x_j) = \sum_{j>i} K_j(x_i) \tag{12}$$

$$\mathbb{KA}^*_i(x_i) = \sum_{j>i} \mathbf{1}_{x_j = x^*_j} K(x_i, x^*_j) = \sum_{j>i} K^*_j(x_i) \tag{13}$$

$$\mathbb{KB}_i(x_i) = \sum_{j<i} Q_j(x_j)K(x_j, x_i) = \sum_{j<i} K_j(x_i) \tag{14}$$

$$\mathbb{KB}^*_i(x_i) = \sum_{j<i} \mathbf{1}_{x_j = x^*_j} K(x^*_j, x_i) = \sum_{j<i} K^*_j(x_i) \tag{15}$$

Instead of computing the exact distribution $P(X, I)$ presented above, the structured variational approximation finds the distribution $Q(X, I)$ among a given family of distributions that best fits the exact distribution in terms of KL-divergence. By choosing a family of tractable distributions, it is possible to make inference involving the ideal distribution tractable. Here we use $Q(X, I) = Q_{\mathcal{I}}(I) \prod_i Q_i(x_i)$, the structured mean-field approximation. More details on mean-field approximation are presented section 11.5 generic update equation for Q (Equation 11.54 in [**?**]) is:

$$Q(x_i) \propto \exp\left\{E_{X_{-i} \sim Q}\left[\log P(x_i | X_{-i})\right]\right\} \tag{16}$$

where X_{-i} refers to all variables except x_i. Using Eq. 1 along with Eq. 16 we get the following update equations:

$$\begin{aligned} Q_i(x_i) &\propto \exp\left\{\phi_{\mathcal{X}}(x_i) + \mathrm{E}_{U \sim Q_{\mathcal{I}}}\left[\phi_{\mathcal{XI}}(x_i, U)\right] + \sum_{j>i} \mathrm{E}_{U_j \sim Q_j}\left[\phi_{\mathcal{XX}}(x_i, U_j)\right] + \sum_{j<i} \mathrm{E}_{U_j \sim Q_j}\left[\phi_{\mathcal{XX}}(U_j, x_i)\right]\right\} \\ &\propto \exp\left\{\phi_{\mathcal{X}}(x_i) + \mathrm{E}_{U \sim Q_{\mathcal{I}}}\left[\phi_{\mathcal{XI}}(x_i, U)\right] + \mathbb{FA}_i(x_i) + \mathbb{FB}_i(x_i)\right\} \end{aligned} \tag{17}$$

$$Q_{\mathcal{I}}(I) \propto \exp\left\{\sum_j \mathrm{E}_{U_j \sim Q_j}\left[\phi_{\mathcal{XI}}(U_j, I)\right]\right\} \tag{18}$$

$$\propto \exp\left\{\mathbb{H}_i(I) + H_i(I)\right\} \qquad \text{(Here } i \text{ refers to the frame of interest, but any choice of } i \text{ holds)} \tag{19}$$

where Q_i is marginal distribution with respect to each of the frames, and $Q_{\mathcal{I}}$ is the marginal with respect to the intent.

1.3. Details of the learning algorithm

Training a deep CRF model requires calculating derivatives of the objective in terms of each of the potentials in the model, which in turn requires inference of $P(X, I|V)$. The network is trained to maximize the log-likelihood of the data:

$$l(X^*) = \log \sum_I P(X^*, I|V) \tag{20}$$

$$= \log \sum_I \frac{\tilde{P}(X^*, I|V)}{Z(V)} \tag{21}$$

$$= \log \sum_I \tilde{P}(X^*, I|V) - \log Z(V) \tag{22}$$

$$Z(V) = \sum_I \sum_X \tilde{P}(X, I|V) \tag{23}$$

where we explicitly write out the partition function Z(V), and $\tilde{P}()$ is the unnormalized version of $P()$. Again, we use X^* to explicitly refer to the ground truth labels. As before, V is omitted from the following derivations. The goal is to update the parameters of the model, for which we need gradients with respect to the parameters. Similar to SGD, we find the gradient with respect to one part of the parameters at a time, specifically with respect to one potential in one frame. That is, $\phi^i_{\mathcal{X}}(x)$ instead of $\phi_{\mathcal{X}}(x)$. The partial derivatives of this loss with respect to each of the potentials are as follows.

1.3.1 Updating the frame potential $\phi_{\mathcal{X}}$

The frame potential $\phi_{\mathcal{X}}(x_i)$ incorporates the interplay between activity category, object, action, progress and scene, and could be written explicitly as $\phi_{\mathcal{X}}(C_t, O_t, A_t, P_t, S_t)$. In practice this potential is composed of unary, pairwise, and tertiary potentials directly predicted by a CNN. We found predicting only the following terms to be sufficient without introducing too many additional parameters: $\phi_{\mathcal{X}}(C_t, O_t, A_t, P_t, S_t)$=$\phi(O_t, P_t)$+$\phi(A_t, P_t)$+$\phi(O_t, S_t)$ + $\phi(C_t, O_t, A_t, P_t)$ where we only model the assignments seen in the training set, and assume others are not possible.

Let us first derive the update equation for $\phi_{\mathcal{X}}$ as a whole, and then demonstrate how to update each of the individual potentials. In the following derivation, we simply take the partial derivative where appropriate and iteratively use the chain rule.

$$\frac{\partial l(X^*)}{\partial \phi^{\hat{i}}_{\mathcal{X}}(\hat{x})} = \frac{1}{\sum_I \tilde{P}(X^*, I)} \left(\sum_I \tilde{P}(X^*, I) \right) \frac{\partial \left(\sum_i \phi^i_{\mathcal{X}}(x^*_i) \right)}{\partial \phi^{\hat{i}}_{\mathcal{X}}(\hat{x})} - \frac{\partial \log Z}{\partial \phi^{\hat{i}}_{\mathcal{X}}(\hat{x})} \tag{24}$$

$$= \mathbf{1}_{\hat{x}=x^*} - \frac{1}{Z} \sum_X \sum_I \frac{\partial \tilde{P}(X, I)}{\partial \phi^{\hat{i}}_{\mathcal{X}}(\hat{x})} \qquad \text{(Denominator and numerator cancel)} \tag{25}$$

$$= \mathbf{1}_{\hat{x}=x^*} - \frac{1}{Z} \sum_X \sum_I \mathbf{1}_{\hat{x}=x} \tilde{P}(X, I) \tag{26}$$

$$= \mathbf{1}_{\hat{x}=x^*} - \sum_X \sum_I \mathbf{1}_{\hat{x}=x} P(X, I) \tag{27}$$

$$\approx \mathbf{1}_{\hat{x}=x^*} - \sum_X \sum_I \mathbf{1}_{\hat{x}=x} Q(X, I) \qquad \text{(Using the mean-field)} \tag{28}$$

$$= \mathbf{1}_{\hat{x}=x^*} - \sum_X \sum_I \mathbf{1}_{\hat{x}=x} Q_{\mathcal{I}}(I) \prod_i Q_i(x_i) \tag{29}$$

$$= \mathbf{1}_{\hat{x}=x^*} - Q_{\hat{i}}(\hat{x}) \qquad \text{(Since } \sum_{x_i} Q_i(x_i) = 1) \tag{30}$$

where we use X^* to refer to the ground truth labels, and $\hat{X}$ to refer to the variables we are taking the partial derivative with respect to. We note that $\frac{\partial \left(\sum_i \phi^i_{\mathcal{X}}(x^*_i) \right)}{\partial \phi^{\hat{i}}_{\mathcal{X}}(\hat{x})} = \mathbf{1}_{\hat{x}=x^*}$. Intuitively this implies the partial gradient is the difference between the

ground truth and the model prediction. This equation is easily extended to update each of the individual potentials as follows:

$$\frac{\partial l(X^*)}{\partial \phi^{\hat{i}}(\hat{O}_t, \hat{P}_t)} = \mathbf{1}_{(\hat{O}_t, \hat{P}_t)=(O_t^*, P_t^*)} - \sum_{C_t} \sum_{A_t} \sum_{S_t} Q_{\hat{i}}(X_t^*) \tag{31}$$

$$\frac{\partial l(X^*)}{\partial \phi^{\hat{i}}(\hat{A}_t, \hat{P}_t)} = \mathbf{1}_{(\hat{A}_t, \hat{P}_t)=(A_t^*, P_t^*)} - \sum_{C_t} \sum_{O_t} \sum_{S_t} Q_{\hat{i}}(X_t^*) \tag{32}$$

$$\frac{\partial l(X^*)}{\partial \phi^{\hat{i}}(\hat{O}_t, \hat{S}_t)} = \mathbf{1}_{(\hat{O}_t, \hat{S}_t)=(O_t^*, S_t^*)} - \sum_{C_t} \sum_{A_t} \sum_{P_t} Q_{\hat{i}}(X_t^*) \tag{33}$$

$$\frac{\partial l(X^*)}{\partial \phi^{\hat{i}}(\hat{C}_t, \hat{O}_t, \hat{A}_t, \hat{P}_t)} = \mathbf{1}_{(\hat{C}_t, \hat{O}_t, \hat{A}_t, \hat{P}_t)=(C_t^*, O_t^*, A_t^*, P_t^*)} - \sum_{S_t} Q_{\hat{i}}(X_t^*) \tag{34}$$

where we marginalize out the variables that are not a part of each potential. Again, X_t incorporates all the frame variables $\{C_t, O_t, A_t, P_t, S_t\}$. These partial derivatives are passed down the CNN (backprop) to update the parameters of the network.

1.3.2 Updating the frame-intent potential $\phi_{\mathcal{XI}}$

Similarly to $\phi_{\mathcal{X}}$ we proceed as follows:

$$\frac{\partial l(X^*)}{\partial \phi^{\hat{i}}_{\mathcal{XI}}(\hat{x}, \hat{I})} = \frac{1}{\sum_I \tilde{P}(X^*, I)} \left(\sum_I \tilde{P}(X^*, I) \mathbf{1}_{\hat{x}=x^*} \mathbf{1}_{\hat{I}=I} \right) - \frac{\partial \log Z}{\partial \phi^{\hat{i}}_{\mathcal{XI}}(\hat{x}, \hat{I})} \tag{35}$$

$$= \frac{\tilde{P}(X^*, \hat{I})}{\sum_I \tilde{P}(X^*, I)} \mathbf{1}_{\hat{x}=x^*} - \frac{\partial \log Z}{\partial \phi^{\hat{i}}_{\mathcal{XI}}(\hat{x}, \hat{I})} \tag{36}$$

$$= \frac{\exp\left\{\sum_i \phi^i_{\mathcal{XI}}(x_i^*, \hat{I})\right\}}{\sum_I \exp\left\{\sum_i \phi^i_{\mathcal{XI}}(x_i^*, I)\right\}} \mathbf{1}_{\hat{x}=x^*} - \frac{\partial \log Z}{\partial \phi^{\hat{i}}_{\mathcal{XI}}(\hat{x}, \hat{I})} \qquad \text{(Terms without } I \text{ cancel)} \tag{37}$$

$$= \frac{\exp\left\{\sum_i \phi^i_{\mathcal{XI}}(x_i^*, \hat{I})\right\}}{\sum_I \exp\left\{\sum_i \phi^i_{\mathcal{XI}}(x_i^*, I)\right\}} \mathbf{1}_{\hat{x}=x^*} - \frac{1}{Z} \sum_X \sum_I \frac{\partial \tilde{P}(X, I)}{\partial \phi^{\hat{i}}_{\mathcal{XI}}(\hat{x}, \hat{I})} \tag{38}$$

$$= \frac{\exp\left\{\sum_i \phi^i_{\mathcal{XI}}(x_i^*, \hat{I})\right\}}{\sum_I \exp\left\{\sum_i \phi^i_{\mathcal{XI}}(x_i^*, I)\right\}} \mathbf{1}_{\hat{x}=x^*} - \frac{1}{Z} \sum_X \sum_I \tilde{P}(X, I) \mathbf{1}_{\hat{x}=x} \mathbf{1}_{\hat{I}=I} \tag{39}$$

$$= \frac{\exp\left\{\sum_i \phi^i_{\mathcal{XI}}(x_i^*, \hat{I})\right\}}{\sum_I \exp\left\{\sum_i \phi^i_{\mathcal{XI}}(x_i^*, I)\right\}} \mathbf{1}_{\hat{x}=x^*} - \sum_X \sum_I P(X, I) \mathbf{1}_{\hat{x}=x} \mathbf{1}_{\hat{I}=I} \tag{40}$$

$$\approx \frac{\exp\left\{\sum_i \phi^i_{\mathcal{XI}}(x_i^*, \hat{I})\right\}}{\sum_I \exp\left\{\sum_i \phi^i_{\mathcal{XI}}(x_i^*, I)\right\}} \mathbf{1}_{\hat{x}=x^*} - \sum_X \sum_I Q(X, I) \mathbf{1}_{\hat{x}=x} \mathbf{1}_{\hat{I}=I} \qquad \text{(Mean-field approximation)} \tag{41}$$

$$= \frac{\exp \sum_i \phi_{\mathcal{XI}}(x_i^*, \hat{I})}{\sum_I \exp \sum_i \phi_{\mathcal{XI}}(x_i^*, I)} \mathbf{1}_{\hat{x}=x^*} - Q_{\hat{i}}(\hat{x}) Q_{\mathcal{I}}(\hat{I}) \tag{42}$$

$$= \frac{\exp\left\{\mathbb{H}^*_{\hat{i}}(\hat{I}) + H^*_{\hat{i}}(\hat{I})\right\}}{\sum_I \exp\left\{\mathbb{H}^*_i(I) + H^*_i(I)\right\}} \mathbf{1}_{\hat{x}=x^*} - Q_{\hat{i}}(\hat{x}) Q_{\mathcal{I}}(\hat{I}) \tag{43}$$

This equation can be interpreted in that it captures the difference between the distribution of the intent given the ground truth, and the predicted distribution of the intent.

1.3.3 Updating the frame-frame potential $\phi_{\mathcal{XX}}$

The pairwise potentials $\phi_{\mathcal{XX}}(x_i, x_j)$ for two time points i and j in our model have the form:

$$\phi_{\mathcal{XX}}(x_i, x_j) = \mu(x_i, x_j) \sum_m w^{(m)} k^{(m)}(v_i, v_j) \tag{44}$$

$$= \mu(x_i, x_j) k(v_i, v_j) \tag{45}$$

where μ models the asymmetric affinity between frames, w are kernel weights, and each $k^{(m)}$ is a Gaussian kernel that depends on the videoframes v_i and v_j which are omitted from this notation for convenience, but the probability and the potentials are conditioned on V. In this work we use a single kernel that prioritises short-term interactions:

$$k(v_i, v_j) = \exp\left(-\frac{(j-i)^2}{2\sigma^2}\right) \tag{46}$$

The parameters of the general asymmetric compatibility function $\mu(x_i, x_j)$ are learned from the data, and σ is a hyper-parameter chosen by cross-validation. The parameters of μ are learned as follows, and this could be extended to a more general form of $\phi_{\mathcal{XX}}$:

$$\frac{\partial l(X^*)}{\partial \mu^{\hat{i}}(\hat{x}, \hat{b})} = \frac{1}{\sum_I \tilde{P}(X^*, I)} \left(\sum_I \tilde{P}(X^*, I)\right) \frac{\partial}{\partial \mu^{\hat{i}}(\hat{x}, \hat{b})} \left(\sum_{j>\hat{i}} \phi^i_{\mathcal{XX}}(x^*_i, x^*_j) + \sum_{j<\hat{i}} \phi^i_{\mathcal{XX}}(x^*_j, x^*_i)\right) - \frac{\partial \log Z}{\partial \mu^{\hat{i}}(\hat{x}, \hat{b})} \tag{47}$$

$$= \sum_{j>\hat{i}} \mathbf{1}_{\hat{x}=x^*} \mathbf{1}_{\hat{b}=x^*_j} k(v_{\hat{i}}, v_j) + \sum_{j<\hat{i}} \mathbf{1}_{\hat{x}=x^*} \mathbf{1}_{\hat{b}=x^*_j} k(v_j, v_{\hat{i}}) - \frac{1}{Z} \sum_X \sum_I \frac{\partial \tilde{P}(X, I)}{\partial \mu^{\hat{i}}(\hat{x}, \hat{b})} \tag{48}$$

$$= \sum_{j>\hat{i}} \mathbf{1}_{\hat{x}=x^*} \mathbf{1}_{\hat{b}=x^*_j} k(v_{\hat{i}}, v_j) + \sum_{j<\hat{i}} \mathbf{1}_{\hat{x}=x^*} \mathbf{1}_{\hat{b}=x^*_j} k(v_j, v_{\hat{i}})$$

$$- \frac{1}{Z} \sum_X \sum_I \tilde{P}(X, I) \sum_i \left(\sum_{j>i} \mathbf{1}_{\hat{x}=x} \mathbf{1}_{\hat{b}=x_j} k(v_i, v_j) + \sum_{j<i} \mathbf{1}_{\hat{x}=x} \mathbf{1}_{\hat{b}=x_j} k(v_j, v_i)\right) \tag{49}$$

$$= \sum_{j>\hat{i}} \mathbf{1}_{\hat{x}=x^*} \mathbf{1}_{\hat{b}=x^*_j} k(v_{\hat{i}}, v_j) + \sum_{j<\hat{i}} \mathbf{1}_{\hat{x}=x^*} \mathbf{1}_{\hat{b}=x^*_j} k(v_j, v_{\hat{i}})$$

$$- \sum_X \sum_I Q_{\mathcal{I}}(I) \prod_i Q_i(x_i) \sum_i \left(\sum_{j>i} \mathbf{1}_{\hat{x}=x} \mathbf{1}_{\hat{b}=x_j} k(v_i, v_j) + \sum_{j<i} \mathbf{1}_{\hat{x}=x} \mathbf{1}_{\hat{b}=x_j} k(v_j, v_i)\right) \qquad \text{(Mean-field)} \tag{50}$$

$$\frac{\partial l(X^*)}{\partial \mu^{\hat{i}}(a, b)} = \sum_{j>\hat{i}} \mathbf{1}_{a=x^*_{\hat{i}}} \mathbf{1}_{b=x^*_j} k(v_{\hat{i}}, v_j) - Q_{\hat{i}}(a) \sum_{j>\hat{i}} Q_j(b) k(v_{\hat{i}}, v_j) + \sum_{j<\hat{i}} \mathbf{1}_{b=x^*_{\hat{i}}} \mathbf{1}_{a=x^*_j} k(v_j, v_{\hat{i}}) - Q_{\hat{i}}(b) \sum_{j<\hat{i}} Q_j(a) k(v_j, v_{\hat{i}}) \tag{51}$$

$$= \mathbf{1}_{a=x^*_{\hat{i}}} \mathbb{KA}^*_{\hat{i}}(b) - Q_{\hat{i}}(a) \mathbb{KA}_{\hat{i}}(b) + \mathbf{1}_{b=x^*_{\hat{i}}} \mathbb{KB}^*_{\hat{i}}(a) - Q_{\hat{i}}(b) \mathbb{KB}_{\hat{i}}(a) \tag{52}$$

This update equation consists of two symmetric parts, one for influence from frames before, and one for influence from frames after. Intuitively, this captures the difference in the true affinity between frame i and all frames j on the one hand, and on the other hand the predicted affinity, where the affinity is weighted by the kernel.

1.4. Additional implementation details

A more detailed algorithmic description of the model is presented in Algorithm 1. More details can be found on the project page `https://github.com/gsig/temporal-fields/`.

Training time Training the models in this paper took a while: The RGB stream of the Two-Stream model converged after only 0.2 epochs (20% of the total data, randomly selected) of the training data, but training the Flow stream needed 4.0 epochs to reach the best performance. Our model needed 0.7 epochs for the RGB stream and 8.3 epochs for the Flow stream. Each 0.1 epoch is approximately 1450 batches of size 256 (all labelled frames at 8 FPS), and takes between 3-8 hours depending on

Algorithm 1 Learning for Asynchronous Temporal Fields (Detailed)

Given videos $\mathcal{V}$
while not converged **do**
  for each example in mini-batch **do**
    Sample frame $v \in \mathbf{V} \subseteq \mathcal{V}$ that has index i
    Calculate messages with Eq. 8-15, approximated by Eq. 9 (from paper)
    Alternate updating Q_i and $Q_{\mathcal{I}}$ until convergence
    Find gradients with Eqs. 30,43,52
    Backprop gradients through CNN
    Store computations of Eq. 2-7 for later use
  Update CNN using accumulated gradients

hardware and model. Our learning rate schedule was chosen by finding the largest learning rate that did not cause divergence, and then making sure the learning rate was decayed by a factor of 100 over the course of training. Investigations into training these kinds of models faster are likely to yield substantial benefits.

Training Deep Models with Latent Variables One of the pursuits of this work was introducing latent variables into a deep framework, the intent. The gradient for the frame-intent potential, contains predictions of the model on both sides, which is a common problem in deep reinforcement learning, where a variety of tricks such as target fixing, double Q-learning, and gradient clipping, are used to combat the instability caused by this. In this work we found that simply severing the dependency of the frame-intent variable on the input data got rid of the instability, and still gave acceptable performance on the RGB stream, however we found that this did not give good performance on the Flow stream.

In order to train the network with the frame-intent potential depending on the input data, we experimented with a variety of techniques from the reinforcement learning literature. Only two methods were found to help: Alternating target and prediction networks, and regularization. For alternating target and prediction networks, the network predicts two frame-intent potentials, and then the network randomly chooses which to use as the target, and which to use as the source, and backprop only through one of them. For regularization, we enforce the frame-intent potential to be close to zero, similar to weight decay (set to $4 \cdot 10^{-4}$). Regularization was found to be give slightly better performance, and easy to implement/tune, and was used in this work.

1.5. Details about intent analysis

To analyze the learned intent variable, we defined 10 types of intent: *getting something to eat*, *clean the living space*, *getting dressed*, *getting something from storage*, *get informed*, *get out of bed*, *leave the house*, *photograph something*, *relaxing*, *working*. To identify videos corresponding to the intent, we used keyword related to the intent (such as `closet` and `clothes` for *getting dressed*) and manually verified that the content of the video matched the intent. The analysis demonstrates that the latent intent variables captures non-trivial structure of the label space, but precisely understanding goal-oriented behavior compared to simple activity analysis remains important future work.

1.6. Additional Visualizations of Output Predictions

Due to space constraints in the full paper, we present here additional visualizations from the model. In Figure 2 we present in the same way as Figure 9 (from the paper). That is, we present the 3 most confident categories, 2 most confident actions, and 1 most confident object. For example, in the first row we can see that once the light turns on in the room and the couch becomes visible the category *Sitting on a sofa/couch* fires, which in turn increases the likelihood of *sitting* in the next few frames. Furthermore, in Figure 3 we present similar visualizations, but only the 6 most confident categories, to further understand the interplay between the activity categories. In the first row, we can see a video of a person walking towards the camera, and we can see how one after the other the model recognizes cup, phone, and sandwich, and reasons about these connected activities. Finally, in Figure 4 we present a breakdown of the mean average precision (mAP) by our model for each class of the dataset, sorted by the mAP of our model.

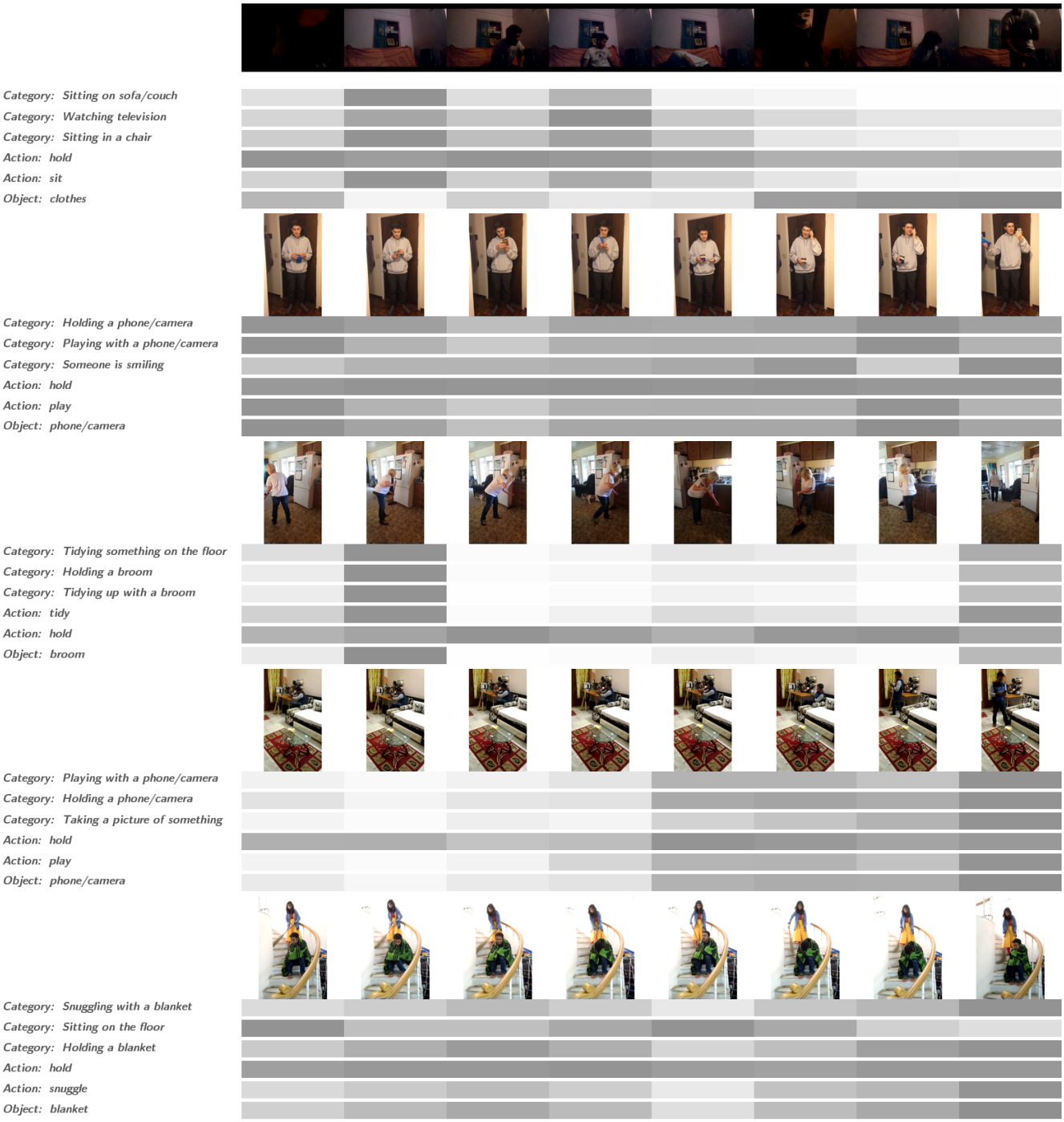

Figure 2. Visualizations of the model predictions for the 3 most confident categories, 2 most confident actions, and 1 most confident object. Darker colors indicate higher likelihood.

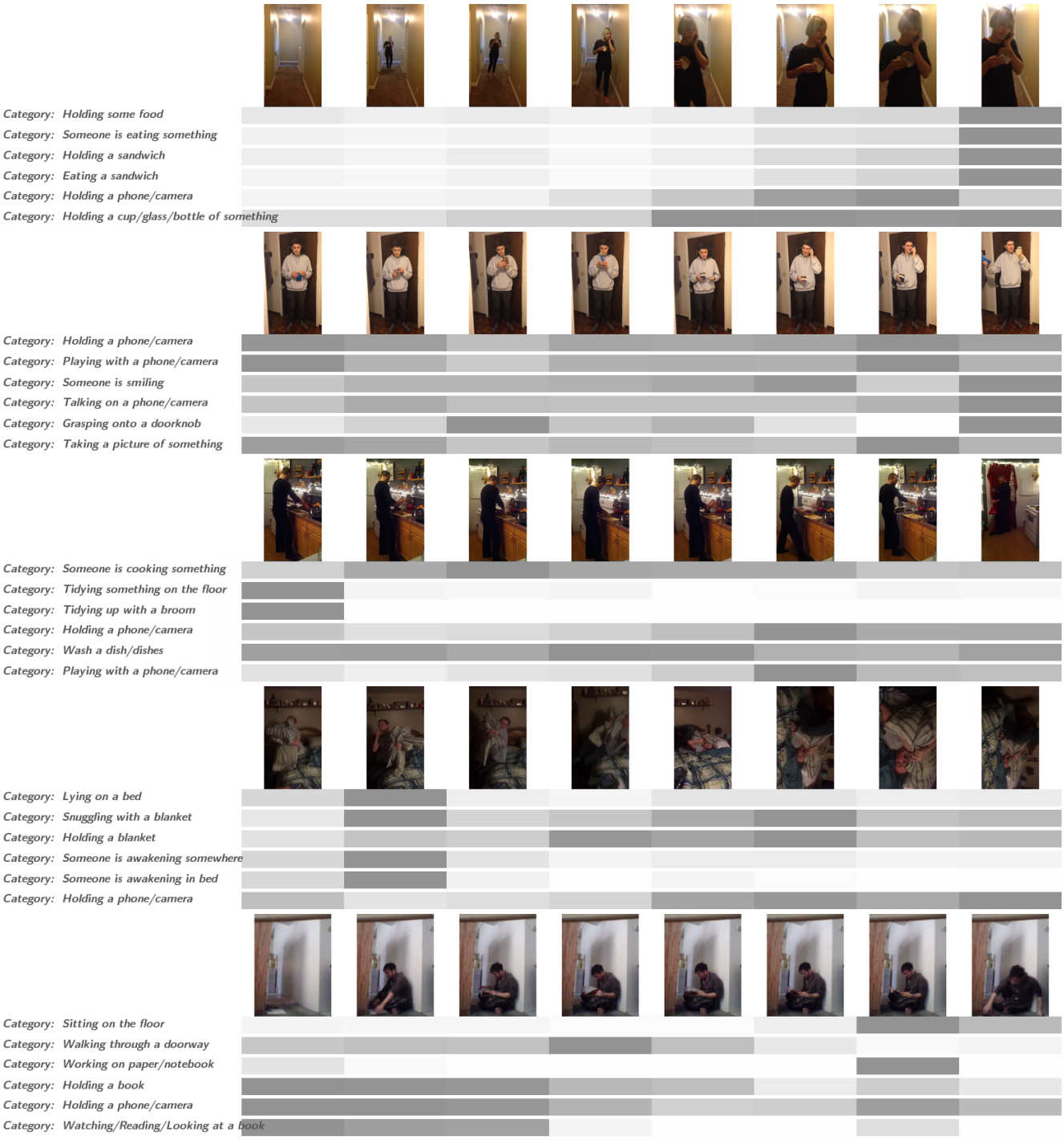

Figure 3. Visualizations of the model predictions for the 6 most confident categories. Darker colors indicate higher likelihood.

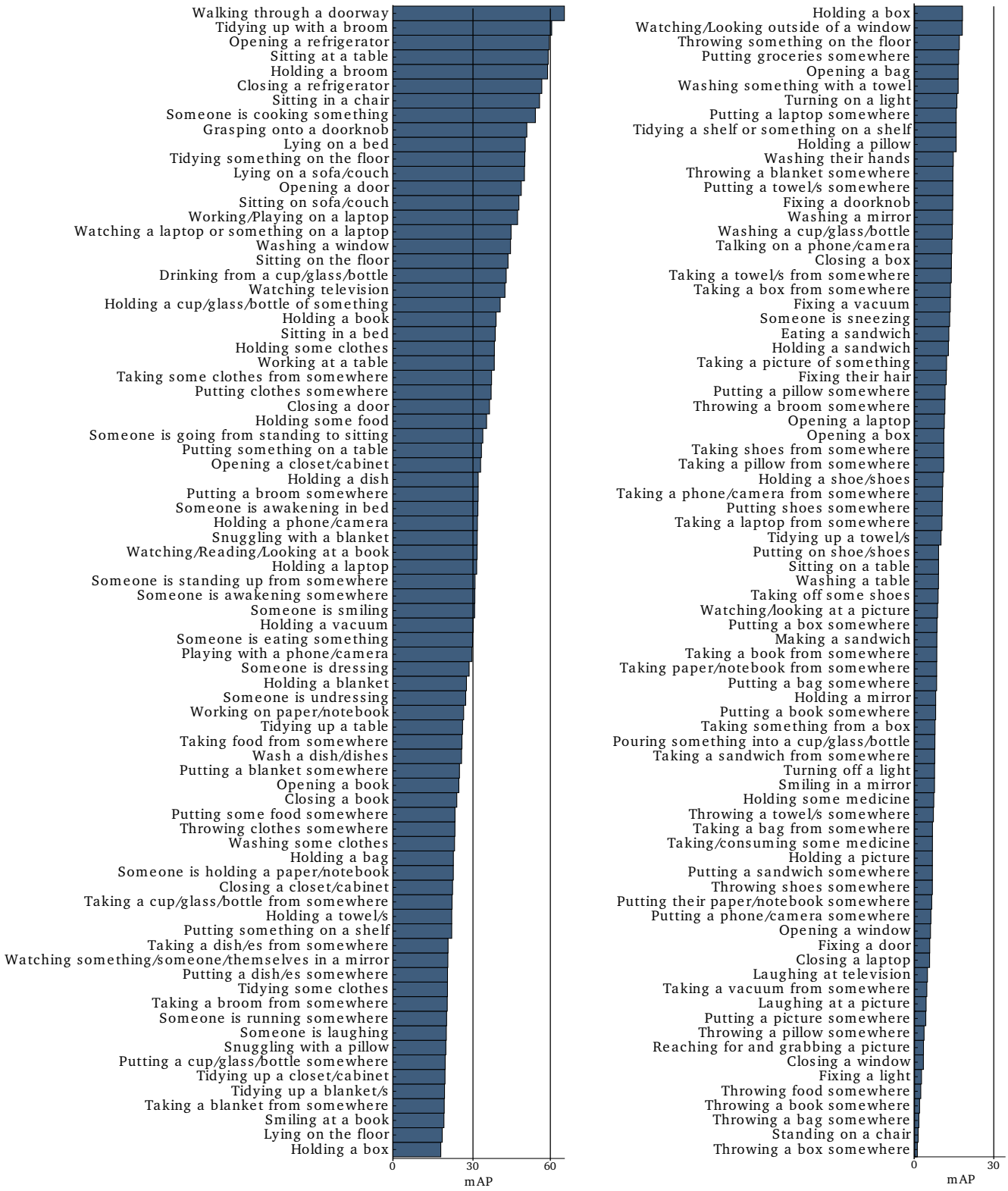

Figure 4. mAP for our model for all classes, sorted by mAP. The column on the right is the continuation of the left column.