Egocentric Video Task Translation

Zihui Xue\textsuperscript{1,2}\* Yale Song\textsuperscript{2} Kristen Grauman\textsuperscript{1,2} Lorenzo Torresani\textsuperscript{2}
\textsuperscript{1}The University of Texas at Austin \textsuperscript{2}FAIR, Meta AI

Abstract

Different video understanding tasks are typically treated in isolation, and even with distinct types of curated data (e.g., classifying sports in one dataset, tracking animals in another). However, in wearable cameras, the immersive egocentric perspective of a person engaging with the world around them presents an interconnected web of video understanding tasks—hand-object manipulations, navigation in the space, or human-human interactions—that unfold continuously, driven by the person’s goals. We argue that this calls for a much more unified approach. We propose EgoTask Translation (EgoT2), which takes a collection of models optimized on separate tasks and learns to translate their outputs for improved performance on any or all of them at once. Unlike traditional transfer or multi-task learning, EgoT2’s “flipped design” entails separate task-specific backbones and a task translator shared across all tasks, which captures synergies between even heterogeneous tasks and mitigates task competition. Demonstrating our model on a wide array of video tasks from Ego4D, we show its advantages over existing transfer paradigms and achieve top-ranked results on four of the Ego4D 2022 benchmark challenges.\footnote{Work done during an internship at FAIR, Meta AI.}

1. Introduction

In recent years, the introduction of large-scale video datasets (e.g., Kinetics \textsuperscript{[6, 33]} and Something-Something \textsuperscript{[22]}) have enabled the application of powerful deep learning models to video understanding and have led to dramatic advances. These third-person datasets, however, have overwhelmingly focused on the single task of action recognition in trimmed clips \textsuperscript{[12, 36, 47, 64]}. Unlike curated third-person videos, our daily life involves frequent and heterogeneous interactions with other humans, objects, and environments in the wild. First-person videos from wearable cameras capture the observer’s perspective and attention as a continuous stream. As such, they are better equipped to reveal these multi-faceted, spontaneous interactions. Indeed egocentric datasets, such as EPIC-Kitchens \textsuperscript{[9]} and Ego4D \textsuperscript{[23]}, provide suites of tasks associated with varied interactions. However, while these benchmarks have promoted a broader and more heterogeneous view of video understanding, they risk perpetuating the fragmented development of models specialized for each individual task.

In this work, we argue that the egocentric perspective offers an opportunity for holistic perception that can beneficially leverage synergies among video tasks to solve all problems in a unified manner. See Figure 1.

Imagine a cooking scenario where the camera wearer actively interacts with objects and other people in an environment while preparing dinner. These interactions relate to each other: a hand grasping a knife suggests the upcoming action of cutting; the view of a tomato on a cutting board suggests that the object is likely to undergo a state transition from whole to chopped; the conversation may further reveal the camera wearer’s ongoing and planned actions.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Given a set of diverse egocentric video tasks, the proposed EgoT2 leverages synergies among the tasks to improve each individual task performance. The attention maps produced by EgoT2 offer good interpretability on inherent task relations.}
\end{figure}
Apart from the natural relation among these tasks, egocentric video’s partial observability (i.e., the camera wearer is largely outside of the field of view) further motivates us to seek synergistic, comprehensive video understanding to leverage complementary cues among multiple tasks.

Our goal presents several technical challenges for conventional transfer learning (TL) [65] and multi-task learning (MTL) [63]. First, MTL requires training sets where each sample includes annotations for all tasks [15, 24, 48, 53, 55, 62], which is often impractical. Second, egocentric video tasks are heterogeneous in nature, requiring different modalities (audio, visual, motion), and different scales (e.g., temporal, spatial or semantic), and different temporal granularities (e.g., action anticipation requires long-term observations, but object state recognition operates at term granularities). Finally, while existing works advocate the use of a shared encoder across tasks to learn general representations, object state recognition poses a hazard to parameter sharing which can lead to negative transfer [21, 24, 38, 53].

To address the above limitations, we propose EgoTask Translation (EgoT2), a unified learning framework to address a diverse set of egocentric video tasks together. EgoT2 is flexible and general in that it can handle separate datasets for the different tasks; it takes video heterogeneity into account; and it mitigates negative transfer when tasks are not strongly related. To be specific, EgoT2 consists of specialized models developed for individual tasks and a task translator that explicitly models inter-task and inter-frame relations. We propose two distinct designs: (1) task-specific EgoT2 (EgoT2-s) optimizes a given primary task with the assistance of auxiliary tasks (Figure 2(c)) while (2) task-general EgoT2 (EgoT2-g) supports task translation for multiple tasks at the same time (Figure 2(d)).

Compared with a unified backbone across tasks [62], adopting task-specific backbones preserves peculiarities of each task (e.g., different temporal granularities) and mitigates negative transfer since each backbone is optimized on one task. Furthermore, unlike traditional parameter sharing [51], the proposed task translator learns to “translate” all task features into predictions for the target task by selectively activating useful features and discarding irrelevant ones. The task translator also facilitates interpretability by explicitly revealing which temporal segments and which subsets of tasks contribute to improving a given task.

We evaluate EgoT2 on a diverse set of 7 egocentric perception tasks from the world’s largest egocentric video benchmark, Ego4D [23]. Its heterogeneous tasks extend beyond mere action recognition to speaker/listener identification, keyframe localization, object state change classification, long-term action anticipation, and others, and provide a perfect fit for our study. Our results reveal inherent task synergies, demonstrate consistent performance improvement across tasks, and offer good interpretability in task translation. Among all four Ego4D challenges involved in our task setup, EgoT2 outperforms all submissions to three Ego4D-CVPR’22 challenges and achieves state-of-the-art performance in one Ego4D-ECCV’22 challenge.

2. Related Work

Transfer Learning. TL [65] aims at transferring knowledge from a source domain to improve the performance in a target domain. The most widely adopted approach is to pretrain a model on a source task then finetune on the target task, as shown in Figure 2(a). Following this paradigm, many video classification models [1, 5, 42, 59] are initialized from models pretrained on ImageNet [11]. In addition, many works propose to transfer knowledge from a large-scale video dataset (e.g., Kinetics [6, 33]) to benefit action recognition in smaller-scale datasets [54] such as UCF-101 [52] and HMDB-51 [37] or to improve other video tasks, such as spatiotemporal action localization [2, 17, 19, 27, 49] and video anomaly detection [25, 41]. While this technique is ubiquitous in video understanding, prior approaches only consider the transfer from one single source task (dataset) and are thus unable to model the relations among multiple video tasks simultaneously.

Taskonomy [62] presents task transfer with a thorough analysis on the structure of multiple visual tasks. Many works [15, 48, 53, 61] continue along this direction and explore visual task relations, yet they limit the discussion to
static images and generally require a unified design across all tasks. In contrast, we consider a diverse set of egocentric video tasks, which are addressed with a heterogeneous set of task-specific video architectures (e.g., accommodating different time, space, or multimodality). Clearly, forcing the same network architecture across all tasks can be suboptimal for each individual task. This motivates our proposed EgoT2-s (Figure 2(c)), where we preserve the heterogeneous backbones developed for each task and build a task translator on top of the task-specific models.

**Multi-task Learning.** In MTL [63], a single model is trained to address multiple tasks simultaneously in order to capture synergistic supervision across tasks. As depicted in Figure 2(b), hard parameter sharing [51] (i.e., sharing a backbone among tasks and keeping one separate head for each task) is the most commonly used technique within this genre. Although MTL has shown to be beneficial of video analysis [3, 18, 26, 32, 39, 44, 45], there is ongoing debate about the best strategies to determine what parameters to share across which tasks [7, 24, 31, 53, 55]. As pointed out in [34], when MTL is achieved by means of a single common backbone, the performance tends to decrease when the number of tasks grows beyond a certain point. Furthermore, many works [21, 24, 38, 53] observe that over-sharing a network across unrelated tasks causes negative transfer and hinders individual task performance. While soft parameter sharing [14, 60] mitigates this by retaining distinct copies of parameters, it still requires adopting the same identical architecture and “similar” weight values across all tasks.

In the video domain, several works utilize synergies between related tasks (e.g., action recognition with gaze prediction [18, 26, 39] or body pose estimation [44]). However, when selected tasks are not strongly related, prior approaches that split the learning capacity of a shared backbone over multiple tasks can suffer from task competition and inferior performance. In the image domain, with the great advancement of transformers [58], training with multiple datasets together for a generalist model is gaining popularity. Recent work [8, 20, 29, 30, 35, 43] investigates a unified transformer architecture across a diverse set of tasks. Our variant EgoT2-g (Figure 2(d)) is motivated by the desiderata of shared knowledge encapsulated by MTL and of a generalist model. Unlike previous learning paradigms, we adopt a “flipped design” involving separate task-specific backbones and a task translator shared across all tasks. This effectively mitigates task competition and achieves task translation for all tasks simultaneously.

### 3. Approach

We are given $K$ video tasks, $T_k$ for $k = 1, \ldots, K$. We note that our approach does not require a common training set with annotations for all tasks. Let the dataset for task $T_k$ be $\mathcal{D}^T_k = \{(x_{i, k}^T, y_{i, k}^T), \forall i \in N_k\}$, where $(x_{i, k}^T, y_{i, k}^T)$ denotes the $i$-th pair of (input video, output label) and $N_k$ represents the number of given examples. Note that “labels” $y_{i, k}^T$ can be a variety of output types, and are not limited to category labels. For simplicity we omit the subscript $i$ hereafter.

We consider two formulations with distinct advantages: (1) task-specific translation, where we partition the tasks into one primary task $T_p$ and $K - 1$ auxiliary tasks, and optimize the objective to improve performance on $T_p$ with the assistance of the auxiliary tasks (EgoT2-s, Sec. 3.1); (2) task-general translation, where we treat all $K$ tasks equally, and the goal is to maximize the collective performance of all the tasks (EgoT2-g, Sec. 3.2). As demonstrated in our experiments, objective (1) leads to the strongest performance on the primary task, while objective (2) offers the benefit of a single unified model addressing all tasks at once.

#### 3.1. Task-Specific Translation: EgoT2-s

The training of EgoT2-s is split over two stages.

**Stage I: Individual-Task Training.** We train a separate model $f_k$ on each individual task dataset $\mathcal{D}^T_k$, obtaining $K$ task-specific models $\{f_k\}_{k=1}^K$. We do not place any restrictions on the task-specific model designs, nor do we require a unified design (i.e., identical encoder-decoder architecture) across tasks. Therefore, any available model checkpoint developed for task $T_k$ can be adopted as $f_k$ within our framework, offering maximum flexibility.

**Stage II: Task-Specific Translation.** We train a task translator that takes features produced by task-specific models as input and outputs predictions for the primary task. Formally, let $h_k \in \mathbb{R}^{T_k \times D_k}$ be features produced by the $k$-th task-specific model $f_k$, where $T_k$ is the temporal dimension and $D_k$ is the per-frame feature dimension for model $f_k$. Following the feature extraction step, we design a projection layer $P_k \in \mathbb{R}^{D_k \times D}$ for each $f_k$ to map task-specific features to a shared latent feature space. This yields a temporal sequence of task-specific tokens $\tilde{h}_k \in \mathbb{R}^{T_k \times D}$.

We process this collection of task-specific temporal sequences using a transformer encoder [58] of $L$ layers to capture both inter-frame and inter-task dependencies. We denote the propagation rule of layer $l$ by $z^{l+1} = Encoder^l(z^l)$. Finally, we adopt a decoder head $Decoder^{T_p}$ to obtain predictions for the primary task $T_p$.

In all, this stage has four major steps: (1) feature extraction; (2) feature projection; (3) transformer fusion; and (4) feature decoding. The procedure is summarized below:

\begin{align*}
    h_k &= f_k(x^{T_p}_r), \quad \forall k \in \{1, 2, \ldots, K\} \\
    \tilde{h}_k &= P_k h_k, \quad \forall k \in \{1, 2, \ldots, K\} \\
    z^0 &= [\tilde{h}_1, \tilde{h}_2, \ldots, \tilde{h}_K] \\
    z^{l+1} &= Encoder^l(z^l), \forall l \in \{0, 1, \ldots, L - 1\} \\
    y_{pred}^{T_p} &= Decoder^{T_p}(z^L)
\end{align*}
where \( y_{pred}^T \) denotes the prediction given by EgoT2-s. During the second stage of training, we freeze the task-specific models and optimize the task translator with respect to the primary task dataset \( D_{T1}^P \).

Figure 3 (left) illustrates the design of EgoT2-s using three social interaction tasks from Ego4D [23] as an example. EgoT2-s allows heterogeneity in the task-specific models (i.e., \( f_1 \) is unimodal while \( f_2 \) and \( f_3 \) are multimodal; also the three task-specific models can be associated with different frame rates and temporal durations) and utilizes a transformer encoder to model inter-frame and inter-task relations. The resulting EgoT2-s learns to adaptively utilize auxiliary task features for the primary task prediction.

### 3.2. Task-General Translation: EgoT2-g

EgoT2-s optimizes performance for a single primary task. Therefore, in the event all \( K \) tasks must be addressed, it requires \( K \) separate training runs and \( K \) distinct translators. This motivates us to extend EgoT2-s to perform task translation for all \( K \) tasks at once. In EgoT2-g, the task translator processes features from all \( K \) tasks and learns to “translate” features conditioned on the task of interest.

The first stage of EgoT2-g is identical to EgoT2-s. For the second stage, we propose two main modifications. First, we replace the task-specific decoder in EgoT2-s with a “generalist” decoder that outputs predictions conditioned on the task of interest. Natural language provides us with a flexible scheme to specify all tasks as a sequence of symbols. Inspired by [8], we tokenize all task outputs and replace the original task-specific decoder with a sequence decoder [30] for a unified interface. Specifically, we first transform the original label \( y^T \) to a target output sequence \( y_{seq}^T \in \mathbb{R}^M \), where \( M \) is the target sequence length. For the task translator to produce task-dependent outputs, we prepend a task prompt token \( y_{prompt} \) to the target output, i.e., \( y_{seq}^T = y_{prompt} \). We then let the sequence decoder generate a sentence answering the requested task. Figure 3 (right) illustrates how we express task outputs as sequences of discrete tokens and attach task prompts.

With the transformed output, we treat the problem as a language modeling task and train the task translator to predict subsequent tokens (one token at a time) conditioned on the input video and its preceding tokens. The training objective is \( L_T = \sum_{j=1}^M w_j \log P(y_{seq}^T_t | x^T, y_{seq}^T_{t+1} ... ) \). Note that the maximum likelihood loss is weighted to mask the loss corresponding to the task prompt token: \( w_j = 0 \) for \( j = 1 \), and to 1 for any other \( j \). During inference, the task prompt is prepended, and the task translator predicts the remaining output tokens. We use argmax sampling (i.e., take the token with the largest likelihood) to sample tokens from the model likelihood and transform the output tokens back to the original label space. Detokenization is easy as we simply reverse the tokenization process.

The second modification lies in the training strategy. While EgoT2-s adopts the primary task dataset for training, EgoT2-g requires joint training on all \( K \) task datasets. Sim-
ilar to the training strategy in \cite{liu2018unsupervised, liu2020self}, we sample one batch from each task, compute the task loss, aggregate the $K$ gradients, and perform model updates in one training iteration. The final training objective is $L = \sum_{k=1}^{K} L^k$. Figure 3 contrasts the design of EgoT2-s and EgoT2-g. They both provide a flexible framework that can incorporate multiple heterogeneous task-specific models (e.g., the three example tasks we give here focus on different aspects of human-object interactions). With a design and an optimization that are specialized to a single primary task, EgoT2-s is expected to lead to superior individual task performance while EgoT2-g brings the efficiency and compactness benefits of a single translator addressing all tasks.

4. Experiments

4.1. Experimental Setup

Dataset and Tasks. We evaluate on Ego4D \cite{ragan2019ego4d}, the world’s largest egocentric video dataset with 3,670 hours of videos spanning hundreds of scenarios (e.g., household, outdoor, leisure). It offers five benchmarks: episodic memory (EM), hands and objects (HO), audio-visual diarization (AV), social interactions (Social) and forecasting. For our study, we select 7 tasks spanning 4 benchmarks, representing a variety of tasks in egocentric perception, as illustrated in Figure 4. The 7 tasks fall into two broad clusters: (a) human-object interactions and (b) human-human interactions. Table 1 summarizes our task setup. For each cluster, we use tasks from the same benchmark as well as tasks across benchmarks, in an attempt to reveal connections among seemingly unrelated tasks. The 7 candidate tasks are heterogeneous in nature as they are defined on videos of varying duration, adopt different video models as backbones, and process unimodal (i.e., video) or multimodal (i.e., video and audio) input, offering a diverse task setup for our study. See Appendix A.2.1 for more details.

Models and Baselines. For each task, we adopt for consistency the baseline models introduced with the Ego4D dataset\footnote{We use model checkpoints provided on the Ego4D website: \url{https://github.com/EGO4D}.} as the task-specific (TS) models in EgoT2. For task-specific translation (Sec. 4.2), we train one task translator for each primary task and use all the other tasks in the same cluster (either human-object interactions or human-human interactions) as auxiliary tasks. We compare EgoT2-s with two representative transfer learning approaches: (1) Transfer \cite{zhou2022pre} denotes finetuning a transfer function on top of features produced by the auxiliary task models (Figure 2(a)). (2) Late Fusion \cite{sun2021late} (LF) concatenates auxiliary task features along with primary task features, and finetunes a few layers that receive the concatenated features as input for the final prediction. Furthermore, to gauge possible improvements over TS by increasing capacity, we consider a

<table>
<thead>
<tr>
<th>Task</th>
<th>Benchmark</th>
<th>Mod.</th>
<th>Duration (seconds)</th>
<th>Model backbone</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNR</td>
<td>HO</td>
<td>V</td>
<td>8.0</td>
<td>E3D RN-50 [6]</td>
</tr>
<tr>
<td>OSCC</td>
<td>HO</td>
<td>V</td>
<td>8.0</td>
<td>E3D RN-50 [6]</td>
</tr>
<tr>
<td>AR</td>
<td>Forecasting</td>
<td>V</td>
<td>8.0</td>
<td>SlowFast [19]</td>
</tr>
<tr>
<td>LTA</td>
<td>Forecasting</td>
<td>V</td>
<td>16.0</td>
<td>SlowFast [19]</td>
</tr>
<tr>
<td>LAM</td>
<td>Social</td>
<td>V</td>
<td>0.2</td>
<td>3D RN-18 [57]</td>
</tr>
<tr>
<td>TTM</td>
<td>Social</td>
<td>V&amp;A</td>
<td>2.7</td>
<td>3D RN-18 [57]</td>
</tr>
<tr>
<td>ASD</td>
<td>AV</td>
<td>A&amp;V</td>
<td>3.7</td>
<td>TalkNet [56]</td>
</tr>
</tbody>
</table>

Table 1. Task Descriptions. ‘Mod.’ is short for modality; ‘A’ and ‘V’ denote audio and video, respectively.

Finetuning \cite{sabour2017dynamic} baseline, which finetunes a few layers on top of the features produced by the primary task model. In order to make a fair comparison, the first-stage training of these baselines is identical to that of EgoT2, and the number of parameters in the second stage of training is set to match that of EgoT2-s as closely as possible.

For task-general translation (Sec. 4.3), the task translator is jointly optimized for all tasks within a cluster\footnote{There is a significant domain gap between human-human and human-object interaction videos. See Appendix A.3 for cross-cluster EgoT2-g.}, thus we have one task translator for human-object interactions that attends to all tasks simultaneously and one translator that performs three human-human interaction tasks at the same time. For comparison with EgoT2-g, we implement the most widely adopted multi-task learning approach, hard parameter sharing \cite{lee2019multi} (Figure 2(b)).

Implementation Details. There is one video preprocessing step before the feature extraction step in Equation (1), where we transform the original video input from $x_k^{T_p}$ to match the input format of the $k$-th task-specific model $f_k$. In particular, $x_k^{T_p}$ is first upsampled or downsampled to match the frame rates required by $f_k$. Next, if the temporal span of the auxiliary task is smaller than that of the primary task, we slide $f_k$ in a moving window to extract a sequence of features, where the window length is the temporal span required by $f_k$, and stride size is a hyperparameter. Conversely, if $f_k$ requires video inputs of a longer temporal span...
than $x^{T_p}$, we exclude task $k$ from auxiliary task candidates to avoid providing potential advantages of a longer observation window to our framework as otherwise we need to expand video length of $x^{T_p}$ to match the requirement of $f_k$. Moreover, if the auxiliary task dataset is multimodal (i.e., video and audio) and the primary task involves only video, we apply the unimodal video pathway of $f_k$ to obtain features; if the primary task is multimodal, we provide all task-specific features that are computable from these modalities. See Appendix A.2.2 for more implementation details.

4.2. Evaluation of Task-Specific Translation

Results. We conduct experiments with EgoT2-s for each task being the primary task and summarize the results for human-object interactions and human-human interactions in Table 2 and 3, respectively.

From the two tables, we observe uneven performance by the baseline methods. Transfer and Late Fusion sometimes outperform the dedicated TS model and sometimes underperform it. When tasks do not exhibit a strong transfer relation, reusing the backbone of the auxiliary task for the primary task leads to negative transfer and performance degradation. For instance, in Table 2, when $T_p$ is AR, Transfer (OSCC) and Late Fusion both down grade accuracy, suggesting object state change is more dependent on verbs and unrelated to noun prediction tasks in AR.

On the contrary, our proposed EgoT2-s learns to adaptively utilize task-specific features and effectively mitigates negative transfer, demonstrating consistent improvement over the TS model for all 6 cases. For instance, in Table 3, when $T_p$ is ASD, Late Fusion indicates there is a deleterious relation from LAM and TTM to ASD, as it suffers from an accuracy degradation of 1.51% over TS, yet EgoT2-s still obtains slightly better performance compared to TS (i.e., 79.38% vs. 79.05%). Moreover, when auxiliary tasks are beneficial for the primary task, EgoT2-s outperforms all baselines with fewer trainable parameters. For example, when $T_p$ is TTM, it achieves a +7.63% mAP improvement over the original TS model by training a lightweight task translator with only 0.7M parameters on top of it (TS is kept frozen). These results across different primary and auxiliary task combinations help demonstrate the generalizability of EgoT2-s. See Appendix A.3 for experiments using a subset of auxiliary tasks rather than all tasks.

Ablation Study. In Table 4, we ablate three different design choices of EgoT2-s using TTM as the primary task: (a) We replace the LAM and ASD TS models in EgoT2-s with two TTM models with different parameters. This yields a task fusion transformer that is architecturally identical to EgoT2-s but takes only TTM tokens as input; (b) We pass features produced by TS models after temporal pooling as the input of our task fusion transformer; (c) We do not freeze TS models in our second-stage training. By comparing (a) with our default configuration (d), we see that EgoT2-s indeed benefits from the introduction of auxiliary tasks. Although equipped with three different TTM models

### Table 2. Results of EgoT2-s as we vary the primary human-object interaction task $T_p$. First row records performance of the task-specific (TS) model we obtain in the first-stage training; we compare EgoT2-s with other baseline methods in the second-stage training. We list the number of trainable parameters for each separate stage as well as the total (i.e., trainable parameters plus parameters of frozen TS models) in parentheses. Following [23], the evaluation metric is temporal localization error (unit: seconds) for PNR, accuracy for OSCC and AR, and edit distance at future 20 time stamps (i.e., ED@20) for LTA. For localization error and ED@20, lower is better. EgoT2-s reliably adapts the auxiliary tasks to suit the target task.

<table>
<thead>
<tr>
<th>$T_p$ is PNR</th>
<th># Params -10⁶</th>
<th>Error (%)</th>
<th>$T_p$ is OSCC</th>
<th># Params -10⁶</th>
<th>Acc. (%)</th>
<th>$T_p$ is AR</th>
<th># Params -10⁶</th>
<th>Acc. (%)</th>
<th>$T_p$ is LTA</th>
<th># Params -10⁶</th>
<th>ED@20 (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS model [23]</td>
<td>32.2 (32.2)</td>
<td>0.615</td>
<td>32.2 (32.2)</td>
<td>68.22</td>
<td>63.3 (63.3)</td>
<td>22.18</td>
<td>21.55</td>
<td>180 (242)</td>
<td>0.746</td>
<td>0.789</td>
<td></td>
</tr>
<tr>
<td>Finetuning [13]</td>
<td>8.4 (40.6)</td>
<td>0.611</td>
<td>8.4 (40.6)</td>
<td>67.93</td>
<td>4.9 (66.8)</td>
<td>21.64</td>
<td>22.84</td>
<td>48.6 (206)</td>
<td>0.744</td>
<td>0.787</td>
<td></td>
</tr>
<tr>
<td>Transfer [62]</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>66.80</td>
<td>4.9 (37.1)</td>
<td>19.98</td>
<td>5.44</td>
<td>65.4 (97.6)</td>
<td>0.778</td>
<td>0.902</td>
<td></td>
</tr>
<tr>
<td>Transfer [62]</td>
<td>8.4 (40.6)</td>
<td>0.611</td>
<td>N/A</td>
<td>66.80</td>
<td>4.9 (37.1)</td>
<td>20.00</td>
<td>9.61</td>
<td>65.4 (97.6)</td>
<td>0.774</td>
<td>0.899</td>
<td></td>
</tr>
<tr>
<td>Transfer [62]</td>
<td>9.5 (71.4)</td>
<td>0.613</td>
<td>9.4 (71.4)</td>
<td>70.98</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>53.3 (115)</td>
<td>0.745</td>
<td>0.806</td>
<td></td>
</tr>
<tr>
<td>LF [45] (All Tasks)</td>
<td>9.6 (135)</td>
<td>0.610</td>
<td>9.6 (135)</td>
<td>72.10</td>
<td>5.2 (131)</td>
<td>21.11</td>
<td>19.24</td>
<td>83.6 (427)</td>
<td>0.744</td>
<td>0.788</td>
<td></td>
</tr>
<tr>
<td>EgoT2-s (All Tasks)</td>
<td>6.4 (132)</td>
<td>0.610</td>
<td>7.4 (133)</td>
<td>72.69</td>
<td>4.3 (130)</td>
<td>23.04</td>
<td>23.28</td>
<td>41.8 (348)</td>
<td>0.731</td>
<td>0.769</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3. Results of EgoT2-s as we vary the primary human-human interaction task $T_p$. EgoT2-s consistently improves the TS model.

<table>
<thead>
<tr>
<th>$T_p$ is TTM</th>
<th># Params -10⁶</th>
<th>mAP</th>
<th>$T_p$ is ASD</th>
<th># Params -10⁶</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS model [23]</td>
<td>20.2 (20.2)</td>
<td>58.91</td>
<td>15.7 (15.7)</td>
<td>79.05</td>
<td></td>
</tr>
<tr>
<td>Finetuning [13]</td>
<td>0.8 (20.8)</td>
<td>59.67</td>
<td>1.1 (16.8)</td>
<td>78.62</td>
<td></td>
</tr>
<tr>
<td>Transfer [62] (LAM)</td>
<td>0.8 (15.4)</td>
<td>63.59</td>
<td>1.6 (16.2)</td>
<td>66.40</td>
<td></td>
</tr>
<tr>
<td>Transfer [62] (TTM)</td>
<td>N/A</td>
<td>N/A</td>
<td>1.6 (21.6)</td>
<td>71.06</td>
<td></td>
</tr>
<tr>
<td>Transfer [62] (ASD)</td>
<td>0.8 (16.5)</td>
<td>62.31</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>LF [45] (All Tasks)</td>
<td>1.2 (51.1)</td>
<td>64.29</td>
<td>1.6 (51.9)</td>
<td>77.54</td>
<td></td>
</tr>
<tr>
<td>EgoT2-s (All Tasks)</td>
<td>0.7 (51.1)</td>
<td>66.54</td>
<td>1.5 (51.9)</td>
<td>79.38</td>
<td></td>
</tr>
</tbody>
</table>

---

4. Following the time-span guidelines in Sec. 4.1, LAM is not considered as the primary task and LTA is not adopted as an auxiliary task. Nevertheless, Appendix A.3 shows some special cases for completeness.
and a larger model size (the total number of parameters of three TTM models is larger than the sum of three TS models), variant (a) does not bring as much performance gain as EgoT2-s (d). Also, preserving the temporal information of task-specific tokens further boosts performance, as can be seen in the comparison of EgoT2-s (b) with EgoT2-s (d). Finally, not freezing TS (c) greatly increases the training cost yet brings no performance gain. These results validate the design of our proposed EgoT2-s.

### 4.3. Evaluation of Task-General Translation

**Results.** Table 5 provides results of EgoT2-g. Since the TTM and LAM baseline models use identical video backbones \( i.e., 3D \text{ResNet-18} \), the hard parameter sharing multi-task baseline \([51]\) can jointly learn TTM and LAM. Yet this model design is unable to solve the ASD task without further modifications to the ASD backbone model. In contrast, our EgoT2-g provides a flexible solution that can incorporate a heterogeneous mix of pretrained models. Similarly, we apply the multi-task baseline to PNR and OSCC, as they use the same video backbone \( i.e., 13D \text{ResNet-50} \). Compared with dedicated TS models, our proposed EgoT2-g performs task translation for all tasks at the same time and achieves on parallel or better performance for all tasks. For instance, it achieves +5.58\% mAP improvement for TTM and 3.5\% accuracy gain for OSCC. Notably, on ASD, it retains the top-performance of the original TS models when the other two auxiliary tasks do not help. In contrast, we observe task competition for the multi-task baseline: the improvement for TTM \( i.e., +3.0\% \text{mAP} \) is at the cost of significantly downgraded LAM performance \( i.e., -17.26\% \text{mAP} \). Similarly, sharing an encoder for PNR and OSCC also leads to task competition and suboptimal performance for the multi-task baseline. For a side-by-side comparison, we also implement EgoT2-g that performs task translation for PNR and OSCC only and observe its adverse effect.

Comparison with SOTA Approaches. To further demonstrate the efficacy of both EgoT2-s and EgoT2-g, we submit our model to the EvalAI server to compare it with winning submissions to Ego4D-CVPR’22 and Ego4D-ECCV’22 challenges on the withheld test set. Table 6 shows the results.\(^5\) EgoT2-s achieves top performance for all 4 challenges.

\(^5\)ASD & AR are not applicable since they are not Ego4D challenges.
challenges. By only incorporating basic video backbones (e.g., 3D ResNet-18 and SlowFast) as the task-specific model, EgoT2-s achieves similar or better performance than works that adopt more powerful, novel architectures such as Video Swin Transformer. Moreover, the benefits of our approach are orthogonal to such architecture improvements; e.g., for the OSCC challenge, replacing the I3D ResNet-50 backbone with the one used in EgoVLP [40] can further elevate the accuracy of EgoVLP by 1%. This indicates the success of EgoT2 stems from its effective use of task synergies.

While EgoT2-g is a strong performer that surpasses or matches TS across all tasks, if we compare its results with those of EgoT2-s, we observe that EgoT2-s demonstrates superior performance. This is understandable given that EgoT2-s is individually optimized for each primary task and employs a specialized translator. On the other hand, EgoT2-g provides a favorable unified framework that performs task translation for all tasks simultaneously via the design of a task-general translator. Thus, EgoT2-s serves as the framework of choice for top performance while EgoT2-g provides added flexibility. See Appendix A.3 for a detailed comparison of the performance and efficiency of these two variants.

4.4. Visualization of Uncovered Task Relations

Our proposed EgoT2 explicitly models task relations via a task translator and offers good interpretability on task relations. For EgoT2-s, Figure 5 shows the attention weights of task tokens when the primary task is LTA and the auxiliary task is AR. Given two adjacent input video clips, the goal of LTA is to predict the next action (e.g., put container and turn off nozzle for the two examples here). In the upper example, there is a scene change from the first clip (the temporal segment corresponding to put wheel) to the second clip (the clip corresponding to take container). The attention weights of AR tokens are small for the first clip and large for the second clip. Clearly, the future action to predict is more closely related to the second temporal segment due to similarities in the scene and objects. In the lower example, the AR tokens have large attention weights, as the video is temporally similar and the previous two actions are indicative of the next action. These results show how EgoT2-s accurately characterizes temporal and auxiliary task information to improve the primary task. More visualizations are in Appendix A.4.

Similarly, for EgoT2-g, we visualize its encoder-decoder attention weights from the last layer transformer in Figure 6. Given the same video clip as input, feature tokens are activated differently when EgoT2-g is given different task prompts, demonstrating that EgoT2-g learns to perform task translation conditioned on the task of interest. As it assigns small weights to task features that are not beneficial for the task of interest (e.g., PNR features to noun prediction tasks), EgoT2-g discards non-relevant task features to mitigate task competition. We also observe temporal differences of attention weights from same task features, indicating that EgoT2-g captures both inter-frame and inter-task dependencies to improve the task of interest. Finally, recall that in Figure 1, we derive task relations for 4 human-object interaction tasks via attention weights provided by EgoT2-g. The attention weights are temporally pooled and averaged over all validation data, revealing task relations from a global perspective. Results for human-human interaction tasks are presented in Appendix A.4. In all, EgoT2 provides good interpretability patterns on (1) which subset of tasks (2) which time segments lead to the final prediction.

5. Conclusion

As a step towards unified egocentric perception, we propose EgoT2, a general and flexible design for task translation. EgoT2 consists of heterogeneous video models optimized for each individual task and a transformer-based task translator that captures inter-frame and inter-task relations. We propose EgoT2-s to improve one primary task and EgoT2-g to conduct task translation for all tasks simultaneously. Results on 7 diverse egocentric video tasks reveal valuable task relations and validate the proposed design.
References


