MLink: Linking Black-box Models for Collaborative Multi-model Inference

Mu Yuan, Lan Zhang*, Xiang-Yang Li

University of Science and Technology of China
ym0813@mail.ustc.edu.cn, zhanglan@ustc.edu.cn, xiangyangli@ustc.edu.cn

Abstract

The cost efficiency of model inference is critical to real-world machine learning (ML) applications, especially for delay-sensitive tasks and resource-limited devices. A typical dilemma is: in order to provide complex intelligent services (e.g. smart city), we need inference results of multiple ML models, but the cost budget (e.g. GPU memory) is not enough to run all of them. In this work, we study underlying relationships among black-box ML models and propose a novel learning task: model linking. Model linking aims to bridge the knowledge of different black-box models by learning mappings (dubbed model links) between their output spaces. Based on model links, we developed a scheduling algorithm, named *MLink*. Through collaborative multi-model inference enabled by model links, *MLink* can improve the accuracy of obtained inference results under the cost budget. We evaluated *MLink* on a multi-modal dataset with seven different ML models and two real-world video analytics systems with six ML models and 3,264 hours of video. Experimental results show that our proposed model links can be effectively built among various black-box models. Under the budget of GPU memory, *MLink* can save 66.7% inference computations while preserving 94% inference accuracy, which outperforms multi-task learning, deep reinforcement learning-based scheduler and frame filtering baselines.

Introduction

Multi-model inference workloads are increasingly prevalent, e.g., smart speaker assistants (Bentley et al. 2018), smart cities (Duan et al. 2018), drone-based video monitoring (Dilshad et al. 2020), multi-modal autonomous driving (Feng et al. 2020), etc. Besides the accuracy of the trained models, costs in the inference phase can become the bottleneck to the quality of services, especially for delay-sensitive tasks and resource-limited devices.

Towards cost-efficient inference, existing work explored various perspectives to achieve the resource-performance trade-offs. Multi-task learning and zipping (He, Zhou, and Thiele 2018; Sanh, Wolf, and Ruder 2019; Crawshaw 2020; Zhang and Yang 2021) can reduce the computing overheads by sharing neurons among different tasks; Model compression (Hinton, Vinyals, and Dean 2015; Liu et al. 2018; Goldblum et al. 2020; Bai et al. 2020) techniques attempt to eliminate parameters and connections not related to the inference accuracy; Inference reusing (Guo et al. 2018; Ning, Guan, and Shen 2019) approaches aim to avoid the same or similar computations; Source filtering (Li et al. 2020) methods try to transmit only necessary input data to backend ML models. Adaptive configuration (Jiang et al. 2018) and multi-model scheduling (Yuan et al. 2020a) were proposed to make inference workloads adaptive to the dynamics of input content. We summarize them as answers to an interesting question:

How to obtain as accurate inference results as possible without the exact execution of ML models?

From this perspective, multi-task learning and model compression generates a lighter model for the same inference task(s) by pruning the original model(s). Inference reusing and source filtering techniques reuse previous inference results as the predicted results through analyzing the correlation between inputs. Based on the observation that, for some input data, the accuracy of expensive and cheap models is similar, adaptive configuration analyzes the input dynamics and predicts the inference results of expensive models by executing cheap ones. Adaptive multi-model scheduling predicts unnecessary inference results as empty using the executed models' outputs as the hint information.

We address this problem from a novel perspective: *linking black-box models*. We were motivated by the insight that even ML models that are different in input modalities, learning tasks, architectures, etc., can share knowledge with each other, since ML models are prone to "overlearning" (Song and Shmatikov 2020) and outputs of different models have semantic correlations (Tan et al. 2018). If we can effectively bridge the knowledge among ML models, we can directly predict inference results of remaining models based on executed models' outputs. If the cost of this prediction is low, it is promising to improve the resulting accuracy of inference results from all models under a limited cost budget, compared to the original workflow where results of unexecuted models cannot be obtained at all. To realize this vision, the following two main challenges need to be solved:

(1) **How to build knowledge-level links among black-box and highly different ML models?** In practice, deployed ML models could have different architectures and input modalities, and they could be developed by different programming languages and ML frameworks. The hetero-

*Lan Zhang is the corresponding author.

geneity makes it challenging to design a general model of knowledge-level connections among ML models. On the other hand, model linking should be non-intrusive to the original inference system and require as little model information and code modification as possible. The black-box access of ML models bringing additional challenges to the design and implementation.

(2) **How to efficiently select models to be executed and models to be predicted?** Given a set of ML models, after constructing model links among them, we need to select a proper subset of models to be executed under a certain cost budget, e.g. the allocatable GPU memory. Highly efficient model selection is critical to the cost-performance trade-off, which is non-trivial due to its theoretical hardness.

In this paper, we first formalize the model linking task and propose the design of model links which supports linking heterogeneous black-box ML models. And we develop a model link-based algorithm, named *MLink*, to schedule multi-model inference under a cost budget. We evaluated our designs on a multi-modality dataset with seven different ML models, covering five classes of learning tasks and three types of input modalities. Results show that our proposed model links can be effectively built among heterogeneous black-box models. We evaluated *MLink* on two real-world video analytics systems, one for the smart building and the other for city traffic monitoring, including six visual models and 3,264 hours of video from 58 cameras. Under the budget of GPU memory, *MLink* outperforms baselines (multi-task learning (Crawshaw 2020), deep reinforcement learning-based scheduler (Yuan et al. 2020a) and frame filtering (Li et al. 2020)) and can save 66.7% inference computation while preserving 94% output accuracy.

Problem Statement

In this section, we define the model linking task and the inference under budgets problem.

Model linking. Given a set of black-box ML models $F = \{f_i\}_{i=1}^k$, where $f_i : X_i \to Y_i$ is a function mapping the input to its inference result. ML models can be highly heterogeneous, i.e., different input modalities, learning tasks, architectures, etc. We only assume that input spaces $\{X_i\}_{i=1}^k$ are the same or aligned. The case that different models share the same input spaces is common, e.g., multi-task learning-based robotics (Crawshaw 2020; Zhang and Yang 2021) and multimedia advertising (Yuan et al. 2020b). The aligned input spaces typically exists in the context of multi-modal scenarios, e.g., multi-modality event detection (Elhoseiny et al. 2016) and visual speech synthesis (Baltrušaitis, Ahuja, and Morency 2018). In practice, synchronization in time can easily align inputs for many applications. Moreover, approaches such as spatial alignment of multi-view videos (Black, Ellis, and Rosin 2002) and audio-visual semantic alignment (Wang, Fang, and Zhao 2020) can be adopted for specific scenarios. We define model linking as a function $g_{ij} : Y_i \to Y_j$, i.e., a mapping from the source model f_i's output space to the target model f_j's. Then the composite function $g_{ij} \circ f_i : X_i \to Y_j$ can perform the inference computation of f_j. Correspondingly, g_{ji} links the knowledge of f_i into $g_{ji} \circ f_j$.

Multi-source model links ensemble. When the number of models $k \geq 3$, for one target model f_j, there could be multiple model links from different sources. Let $A \subseteq F$ denote the set of source models. Then for all $f_i \in A$, $g_{ij} \circ f_i$ performs the prediction task to f_j's inference outputs. The question that follows is, how do we determine the final prediction? From the ensemble learning perspective, $\{g_{ij} \circ f_i\}_{f_i \in A}$ constitute a multi-expert model (Yuksel, Wilson, and Gader 2012), which has the potential to perform better prediction with the multi-task & multi-modal representation (Baltrušaitis, Ahuja, and Morency 2018; Zhang and Yang 2021). We define $h_{A,j}$ as the ensemble model link from A to f_j. So the input of $h_{A,j}$ is the set of predictions by g_{ij} where $f_i \in A$. Note that if A has only one element f_i, then $h_{A,j} = g_{ij} \circ f_i$.

Multi-model inference under budget. The model links can be utilized to achieve resource-performance trade-offs of multi-model inference workloads. Let $c(\cdot)$ denote the cost of running a function, e.g., GPU memory or inference time. For resource-limited devices (e.g., smartwatches and mobile phones) and delay-sensitive tasks (e.g., real-time video analytics and audio assistant), there are certain constraints on the total cost. We define B as the cost budget and aim to maximize the inference accuracy under that budget. Let $p(h_{A,j})$ denote the performance measure of the ensemble model link, which depends on the target model's task. We assume the range of p is normalized into $[0, 1]$. For example, the performance measure can be accuracy for classification task and bounding box IoU for the detection task. Following previous efforts for optimizing the inference efficiency (Li et al. 2020; Yuan et al. 2020a), the performance measure the consistency between obtained results and exact inference outputs, instead of ground-truth labels. The multi-model inference under cost budget problem is formalized as:

$$\max_{A \subseteq F} \overbrace{\left(\frac{1}{|F|}\left(\underbrace{\sum_{f_i \in A} 1}_{\text{activated}} + \underbrace{\sum_{f_j \in F \setminus A} p(h_{A,j})}_{\text{predicted}}\right)\right)}^{\text{average output accuracy}} \quad (1)$$
$$s.t. \quad \underbrace{\sum_{f_i \in A} c(f_i)}_{\text{exact inference}} + \underbrace{\sum_{f_j \in F \setminus A} c(h_{A,j})}_{\text{model links}} \leq B.$$

Under the cost budget, the optimization problem aims to maximize the average performance of all models F by selecting an *activated* subset A to be executed. For ease of description, we define the objective function as the *output accuracy*. Activated models do exact inference, so their performance scores are all 1. Models that are not activated only participate in constructing model links and will not be executed during the inference phase; instead, they are predicted by the activated models via ensemble model links. The cost of activated models is performing exact inference, while the cost of predicted models is from running model links. So the model links should be both accurate and lightweight to reduce the cost while preserving the quality of the multi-model inference workloads.

Black-box Model Linking

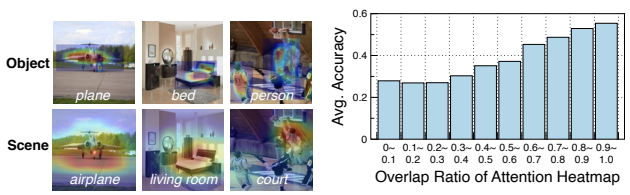
In this section, we discuss the motivation of linking black-box models and present the theoretical analysis, architecture design, ensemble and training methods of model links.

Motivational Study

When training ML models for different tasks, the ideal representation learned by them should be independent and disentangled (Hjelm et al. 2019), i.e. each model only learns the semantics that just covers its objective task. However, due to the mismatched complexity of the data and the model, the machine learning process is prone to "overlearning" (Song and Shmatikov 2020), which means that unintended semantics is encoded in the learned representation. Besides, there exist semantic correlations among outputs of different tasks and different models may pay attention to the same content, e.g., the same regions in images. For example, in Fig. 1a, based on G-CAM (Selvaraju et al. 2020), we plot the attention heatmaps of YOLO-V3 (Redmon and Farhadi 2018) object detector and ResNet50 (He et al. 2016) scene classifier on the same images, and their attention areas have much overlap. We experimented on the correlation between the overlap ratio of attention heatmaps and the performance of model linking on PASCAL VOC (Everingham et al. 2015) dataset. For example, from the scene classification model to the object detection model, as shown in Fig. 1b, the accuracy of model links is obviously relevant with the overlap ratio ($(Map_{source} \wedge Map_{target})/Map_{target}$). To a certain extent, it shows that the correlation learned by model links is similar with the semantic attention. The "overlearning" characteristic and underlying semantic correlations among outputs make mappings from the same or aligned input space to different output spaces transferable (Tan et al. 2018).

Black-box Output vs. Intermediate Representation

A key design principle is that we only use the black-box output of the source model to for model linking. Existing work has shown that by fine-tuning the last few layers (Guo et al. 2019), the intermediate representation can be used to predict other different tasks. However, in real applications, we often have to deal with the deployed models, which only provide a black-box inference API. Compared with intermediate representation, the downstream black-box outputs do have weaker representation capability for general learning tasks. But recent work (Yuan et al. 2020a) shows that, given the same (or aligned) inputs, the executed models' outputs are very effective hints for scheduling unexecuted models. The insight is that the correlation of black-box outputs between multiple tasks with the same input is more explicit and even stronger than the intermediate features. And our experimental results also show that, using the same amount of training data, black-box model linking achieves higher accuracy than a knowledge distillation approach (see Fig. 2) and a multi-task learning approach (see Tab. 4). Considering the better practicality and satisfactory accuracy, we select black-box outputs rather than intermediate representations for linking ML models.



(a) Attention heatmaps of Object and Scene models.

(b) Scene-to-Object MLink accuracy vs. attention overlaps.

Figure 1: Inter-model Semantic Correlation

Sample Complexity Analysis

Let $f \in \mathcal{F}$ denote task-specific parameters and h denote shared parameters across tasks. It has been proved that when the training data for h is abundant, to achieve bounded prediction error on a new task only requires $C(\mathcal{F})$ sample complexity (Tripuraneni, Jordan, and Jin 2020), where $C(\cdot)$ is the complexity of a hypothesis family. Learning a model link $g_{ij} \in \mathcal{G}$ from source model $f_i \in \mathcal{F}_i$ to the target $f_j \in \mathcal{F}_j$ constitutes a compound learning model $g_{ij} \circ f_i$. A lightweight design of model links can makes $C(\mathcal{G}) < C(\mathcal{F}_j)$ hold. Therefore, applying the above result, model linking can significantly reduce the sample complexity to $C(\mathcal{G})$, compared with the $C(\mathcal{F}_j)$ complexity of learning the target model from scratch. This result is also confirmed by our experiments: effective model links can be learnt by a very small amount (e.g. 1%) of training samples (see Fig.2).

Model Link Architecture

Model links map between black-box models' output spaces, so the output format determines the architecture. We classify output formats as the fixed-length vector and the variable-length sequence. These two types of outputs could cover most ML models. We propose four types of model link architectures based on best practices of similar learning tasks.

Vec-to-Vec: The model link maps from a vector-output source to a vector-output target. We use a ReLU-activated multilayer perception (MLP) for the vec-to-vec model link.

Seq-to-Vec: The model link maps from a sequence-output source to a vector-output target. We first use an embedding layer, which performs a matrix multiplication to transform the sequence into a fixed-size embedding. Then we use an LSTM (Hochreiter and Schmidhuber 1997) layer followed by an MLP to generate the vector output.

Vec-to-Seq: The model link maps from a vector-output source to a sequence-output target. We adopt the encoder-decoder framework (Xu et al. 2015), where an MLP serves as the encoder and the decoder consists of an embedding layer, an LSTM layer, an attention layer (Bahdanau, Cho, and Bengio 2015), and a fully-connected layer, in the forward order.

Seq-to-Seq: The model link maps from a sequence-output source to a sequence-output target. We adopt the sequence-to-sequence framework (Sutskever, Vinyals, and Le 2014), where an embedding layer followed by an LSTM layer serves as the encoder and the decoder is the same as the one in the vec-to-seq model link.

The output activation functions are determined by the learning task of the target model. Softmax is used for single-label classification, and sigmoid is used for multi-label classification and sequence prediction. Linear activation works with regression and localization tasks. In our implementation, the default number of hidden units is twice the length of the output dimension, which empirically achieved a good trade-off between effectiveness and efficiency.

Ensemble of Multi-source Links

The ensemble of multi-source model links has the potential to improve the prediction performance (Shen, He, and Xue 2019), since cross-task and cross-modal representation capabilities could be beneficial. For the target model f_j, given the set of sources A, we multiply outputs of g_{ij} by trainable weights, where $f_i \in A$. The weighted prediction is then activated according to f_j's learning task. The learned weights of $h_{A,j}$ can be used to ensemble model links from any subset of sources, i.e., $h_{A',j}, A' \subset A$.

Training

Classic knowledge distillation (Hinton, Vinyals, and Dean 2015) suggests that soft-label supervisions are better for training the "student" model, since the "teacher" model's outputs augment the hard-label space with relations among different classes. Our experimental results show that this empirical experience still holds in the proposed model linking setting. To train model links and the ensemble model, we collect n inference results $\{\{y_i^j\}_{j=1}^k\}_{i=1}^n$ from k models on the same or aligned inputs. Given f_i, f_j as the source and the target, respectively, the objective of training the model link g_{ij} is $\min \sum_{l=1}^n \mathcal{L}_j(g_{ij}(y_i^l), y_j^l)$, where the loss function $\mathcal{L}_j$ depends the learning task of the target model f_j. Given A, f_j as the set of sources and the target, respectively, the objective of training the ensemble model $h_{A,j}$ is $\min \sum_{i=l}^n \mathcal{L}_j(h_{A,j}(\{y_i^l\}_{f_i \in A}), y_j^l)$. Both model links and ensemble models are optimized via gradient descent. Note that if A has only one element f_i, then the ensemble simply fits as an identity layer and $h_{A,j} = g_{ij} \circ f_i$.

Collaborative Multi-model Inference

Let $\mathcal{F}(A)$ denote the objective function $\frac{1}{|F|}(\sum_{f_i \in A} 1 + \sum_{f_j \in F \setminus A} p(A, f_j))$ in the optimization problem (see Eq. (1)). And we define the gain of activating one more model f_i as $\Delta(A, f_i) = \mathcal{F}(A \cup \{f_i\}) - \mathcal{F}(A)$. Assuming that adding a source of model link into the ensemble model will not decrease the performance: $p(A \cup \{f_i\}, f_j) \geq p(A, f_j)$, which is empirically true (Zhou 2012). Then $\Delta(A, f_i) \geq 0$, i.e., the objective function is nondecreasing. In our experiments, we observed two typical cases: (1) Dominance. The performance of the ensemble model approximately equals the best-performance source of model links. Let $f_{i^*} = argmax_{f_i \in A} p(g_{ij})$ denote the source with maximal performance. We observe that $p(h_{A,f_j}) \approx p(g_{i^*j})$, i.e., the best source dominates the ensemble performance. (2) Mutual assistance. The multi-source model links ensemble outperforms any single source. $\forall f_i \in A, p(h_{A,f_j}) > p(g_{ij})$, i.e., sources of model links assist mutually. And in this case, f_j's gain for A_2 is possibly greater than its gain for $A_1, A_1 \subset A_2$, if f_j collaborates better with models in $A_2 \setminus A_1$.

Algorithm 1: Collaborative Multi-model Inference

Input: model set F, cost budget B
For every $f_i, f_j \in F, i \neq j$, train model links g_{ij};
For every $f_j \in F$, train ensemble model $h_{A_j,j}$, where $A_j = F \setminus \{f_j\}$;
for *each period* **do**
  Profiling activation probability of $f_i \in F$;
  Greedily select $A \leftarrow A \cup \{argmax_{f_i \in F \setminus A}(\mathcal{P}_i)\}$ until reach the cost budget B;
  Given incoming input x, for $f_i \in F$:
    if $f_i \in A$ **then**: $y_i \leftarrow f_i(x)$;
    else: $y_i \leftarrow h_{A,i}(\{y_j\}_{f_j \in A})$.
end

Activation probability. Solving Eq. (1) is NP-hard and the existing $(1 - e^{-1})$-approximation algorithm (Sviridenko 2004) needs partial-enumeration and requires $O(n^5)$ computations of the objective function. The optimization is not a one-off process and should be executed online to fit the dynamics of the inference system. So we design a metric of activation probability, whose calculation only depends on the model links' performance rather than ensemble models'. Given a model f_i, the activation probability considers three factors: (1) the average performance of model links from f_i to all the others, denoted as $\mathcal{P}_i^1 = \frac{\sum_{j \neq i} p(g_{ij})}{|F|-1}$; (2) the average performance of model links targeted to f_i from all the others, denoted as $\mathcal{P}_i^2 = \frac{\sum_{j \neq i} p(g_{ji})}{|F|-1}$; (3) the cost of f_i, i.e., $c(f_i)$. Intuitively, the better g_{i*} predicts the others, the worse g_{*i} performs, and the cheaper f_i's execution is, the higher its activation probability should be. Therefore, we design the activation probability as $\mathcal{P}_i = \frac{1+\mathcal{P}_i^1 - \mathcal{P}_i^2}{wc(f_i)}$, where the weight w can be determined by the following normalization. The model with the perfect activation probability should be the one that has links predicting all other models 100% accurately and is predicted by all the others with 0% performance. Also, its cost should be the lowest. Then we have $(1 + 1 - 0)/(w \min_i(c(f_i)) = 1$, i.e., $w = 2/\min_i c(f_i)$. This activation probability can be regarded as an coefficient that are positively correlated with the gain of the objective function when selecting a ML model.

Periodic re-selection. Due to the content dynamics, the optimal subset of activated models may change over time. But adapting to such dynamics brings additional overheads of loading and unloading ML models. So we propose to periodically re-select activated models. At the beginning of each period, we use a small proportion (e.g. 1%) of the data for profiling the prediction performance of model links. Then we update ML models' activation probabilities and re-select models to be loaded during the current period. By reasonably setting the period length and the proportion of data used for profiling, we can amortize the overheads of loading/unloading ML models to negligible.

Algorithm. 1 shows the workflow of integrating MLinks with multi-model inference workloads. Initially, we train

Task Class	ML Model	Input Modality	Output Format	Metric
Single-Label Classification	Gender Classification (gen 2021)	Audio	2-D Softmax Labels	Acc.
Multi-Label Classification	Action Classification (Tran et al. 2015)	Video	12-D Sigmoid Labels	mAP
Localization	Face Detection (Serengil and Ozpinar 2020)	Image	4-D Bounding Box	IoU
	Person Detection (Redmon and Farhadi 2018)	Image		
Regression	Age Prediction (Levi and Hassner 2015)	Image	1-D Scalar	MAE
Sequence Generation	Image Captioning (cap 2021)	Image	Variable-Length Text	WER
	Speech Recognition (spe 2021)	Audio		

Table 1: Summary of ML models used on Hollywood2 dataset.

pairwise model links and ensemble models. During each period, we first calculate the activation probability by running all models on data for profiling. Then we select greedily w.r.t. activation probability under the cost budget. In the serving phase, activated models do exact inference while the others' outputs will be predicted by the model link ensemble of activated sources.

Evaluation

Implementation and Experiment Setup

We implemented our designs in Python based on TensorFlow 2.0 (tf 2021) as a pluggable middleware for inference systems. We tested the integration on programs implemented with TensorFlow (tf 2021) and PyTorch (pyt 2021), with only dozens of lines of code modification, which shows the ease of use of *MLink*. We evaluated our designs on a multi-modal dataset and two real-world video analytics systems.

Multi-modal dataset and ML models. We used the Hollywood2 video dataset (Marszalek, Laptev, and Schmid 2009). To obtain aligned inputs for multi-modal models, we selected the 30th frame and extracted audio data from each video. We deployed seven pre-trained ML models that cover five classes of learning tasks: single-label and multi-label classification, object localization, regression, and sequence generation. And they have different model architectures, input modalities and output formats. To evaluate the performance of model links, we used task-specific metrics, including accuracy, mean average precision (mAP), intersection over union (IoU) of the bounding box, mean absolute error (MAE), and word error rate (WER). Tab. 1 summarizes information of these ML models.

Smart building and city traffic monitoring systems. We evaluated *MLink* on two real-world video analytics systems. 1) Smart building. To support applications including automatic air conditioning and lighting, abnormal event monitoring, and property security, three ML models were deployed: OpenPose (Cao et al. 2019)-based person counting, ResNet50 (He et al. 2016)-based action classification (act 2021), and YOLOV3 (Redmon and Farhadi 2018)-based object counting. We collected two days (one weekday and one weekend) of video frames from all 58 cameras (1 frame per minute). We use an edge server with one NVIDIA 2080Ti GPU. 2) Traffic monitoring. On a city-scale video analytics platform with over 20,000 cameras, three AI models were deployed for traffic monitoring: OpenPose (Cao et al. 2019)-based person counting, ResNet50 (He et al. 2016)-based traffic condition classification (tra 2021), and YOLOV3 (Redmon and Farhadi 2018)-based vehicle counting. We selected 10 cameras at the road intersections and collected two days (one weekday and one weekend) of frames (1 FPS). We used five servers, each with four NVIDIA T4 GPUs.

Baselines. We introduce the naive standalone inference and three strong alternative resource-performance trade-off approaches as baselines. (1) **Standalone**: running models independently. (2) **MTL**: We adopt a multi-task learning architecture (Crawshaw 2020) that consists of a global feature extractor shared by all tasks and task-specific output branches. We use ResNet50 (He et al. 2016) to implement the feature extractor and fully-connected layers for task-specific outputs. We initialize the ResNet50 feature extractor with weights pretrained on ImageNet (Deng et al. 2009) and connect three output branches for person counting, action/traffic classification, and object/vehicle counting tasks, on smart building/traffic monitoring testbeds. The MTL models are trained under the supervision of exact inference results of corresponding models. (3) **Reducto** (Li et al. 2020): a frame filtering approach. For each model, Reducto first computes the feature difference of successive frames. If the feature difference is lower than a threshold, it filters out the current frame and reuses the latest inference output. We tested four types of low-level features as proposed in Reducto and selected the one that has the best performance. (4) **DRLS** (Deep Reinforcement Learning-based Scheduler) (Yuan et al. 2020a): a multi-model scheduling approach. DRLS trains a deep reinforcement learning agent to predict the next model to execute on the given data, based on the observation of executed models' outputs.

Black-box Model Linking

Sensitivity to the size of training data. The original training and test splits in Hollywood2 dataset contain 823 video clips (around 48%) and 884 video clips, respectively. To test the performance of the model linking with different sizes of training data, we further randomly sampled four subsets of training data with 1%, 5%, 10%, 20% ratios, with respect to the total dataset. We trained pairwise model links with the RMSprop (Tieleman and Hinton 2012) optimizer and the same hyper-parameters (0.01 learning rate, 100 epochs, 32 batch size). As a fair comparison, we adopt a knowledge distillation (Hinton, Vinyals, and Dean 2015) method for some target models (Action, Age, Gender) where the student model has two convolutional layers. We repeated the experiments three times and reported the mean and standard deviation of performance. As shown in Fig. 2, using all training

(a) Target: Action (b) Target: Age

(c) Target: Person (d) Target: Gender

Figure 2: Performance of model links from different source models on four targets. KD-Student: student model trained via knowledge distillation on the target model.

Source	Action	Age	Face	Gender
IoU (%)	39.4 (±0.1)	38.9 (±0.1)	**58.5** (±1.3)	39.0 (±0.1)
Corr.	0.123	0.042	**0.244**	-0.053

Table 2: IoU scores of model links targeted to the *Person* model and the Pearson correlations.

data, the *Caption*-to-*Action* model link can achieve 31.7% mAP. And the model links between the two detection models, *Face*-to-*Person* and *Person*-to-*Face* model links, achieve 59% and 32% IoU, respectively. Even with very limited training samples, 1%, some model links achieve high performance. The model link from *Face* to *Gender* achieves 92.1% accuracy. And for model links *Gender*-to-*Age* model achieves 3.0 MAE. Compared with student models trained via knowledge distillation on the target models, model links achieves higher prediction performance, especially when the amount of training data is small. But for speech recognition and video caption models, model links targeted to them cannot be effectively built and have around one WER score.

Discussions. We calculated the Pearson correlation coefficients between inference outputs of different models on the training split. For single-label and multi-label classification models, we used the index with the highest confidence as the label. For localization models, we checked whether the bounding box is empty, and assign 0 or 1 as the label. We used the regression scalar as the label and skipped the two sequence generation models. Tab. 2 shows the results of model links targeted to the *Person* model, and we can see a positive correlation between the model link performance and Pearson correlation coefficient.

Model link ensemble. For one target model, we have built multiple model links from different source models. Then we trained the ensemble models with all sources, using the same optimizer as model links and same hyper-

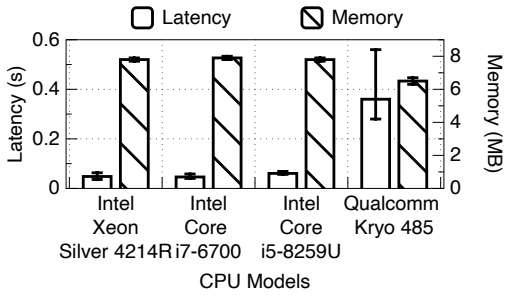


Figure 3: *MLink*'s overheads on servers and mobile phones.

parameters. Tab. 3 shows the results on five target models, both model links and ensemble models were trained using all training samples (48% ratio). The model link ensemble outperforms every single source model. We can see there are two typical cases: dominance and mutual assistance. For *Action*, *Face*, *Person* targets, the *Caption*, *Person*, *Face* sources dominate the ensemble performance, respectively. But for *Age* and *Gender* targets, source models mutually assist and achieve performance improvement by ensemble.

Video Analytics with Model Links

We test *MLink* on 48-hour videos of 58 cameras in a smart building system and 48-hour videos of 10 cameras on a city traffic monitoring platform. We leveraged the first 10% in time of data for training model links and ensembles. We set the period length as one hour and use initial 1% data for profiling activation probabilities. For the counting models, the output accuracy is calculated by checking whether the absolute error of the predicted number is less than 0.5. The time costs of each ML model were the average inference time offline profiled by the training data. In the smart building system, the action/person/object models cost 30/44/60 ms per frame. In the traffic monitoring system, the traffic/person/vehicle models cost 55/66/70 ms per frame. The GPU memory costs of each ML model were the peak usage: 4.6 GB for person counting, 1.5 GB for action/traffic classification, and 3.7 GB for object/vehicle counting. We set the budget B as the maximal GPU memory allocated for ML models to evaluate how *MLink* improves the resource efficiency of multi-model inference. We treat every ML model's output accuracy equally and report their average output accuracy. Under GPU memory budget, the baseline "Standalone" simply selects the model with minimal average time cost. We repeated the scheduling experiments three times and reported the results in Tab. 4. Since the standard deviations are small (< 0.1), we did not present them for simplicity. In both systems, *MLink* outperforms alternatives in output accuracy. Compared with "Standalone", in the smart building system, *MLink* saves 66.7% inference executions, while preserving 94.1% output accuracy.

Overheads of *MLink*. We deployed *MLink* on four different devices (a cloud server, an edge server, a laptop, and a mobile phone) and tested its latency and memory footprint. As shown in Fig. 3, *MLink* only introduces negligible additional overheads.

Target \ Source	Action	Age	Caption	Face	Gender	Person	Speech	Ensemble
Action mAP(%)	-	12.8(±1.3)	**29.7(±1.4)**	10.1(±1.3)	9.3(±0.3)	9.9(±1.2)	8.5(±3.1)	30.8(±1.1)
Face IoU(%)	11(±1.3)	11.2(±1.0)	0 (±0)	-	10.3(±0.9)	**31.9(±0.3)**	0 (±0)	32.2(±0.2)
Person IoU(%)	39.4(±0.1)	38.9(±0.1)	0(±0)	**58.5(±1.3)**	39.0(±0.1)	-	0(±0)	59.2(±1.2)
Age MAE	3.04(±0.01)	-	3.02(±0.01)	3.07(±0.02)	3.0(±0.01)	3.03(±0.01)	3.0(±0.01)	2.98(±0)
Gender Acc.(%)	92(±0.1)	92.1(±0.2)	92(±0.1)	92.1(±0.1)	-	92(±0.1)	92(±0.1)	92.3(±0)

Table 3: Dominance and mutual assistance cases in model link ensemble. Column titles are source models and row titles are target models. The dominant source's performance is in bold.

Method	Building (5/9GB Mem.)		City (5/9GB Mem.)	
	Acc. (%)	Time (ms)	Acc. (%)	Time (ms)
Standalone	33.3/66.7	30/74	33.3/66.7	55/121
MTL	53.3	32.8	61.3	32.5
DRLS	45.7/81.3	58.7/107	39.5/77.6	102/188
Reducto	91.8/96.9	45.7/89	84.1/95.3	64/127
MLink	**94.1/97.9**	39.3/84	**94/97.4**	62/125

Table 4: Comparisons of *MLink*, MTL, Reducto, DRLS, and Standalone methods on two video analytics systems. Mem.: budget of GPU memory. Acc.: average output accuracy. Time: accumulated inference time per frame. MTL serves the same three-task model under budgets of 5GB and 9GB GPU memory, so only one value is reported.

Related Work

Multi-task learning and zipping. One straightforward way to optimize multiple standalone ML models is multi-task learning (Sanh, Wolf, and Ruder 2019; Zhang and Yang 2021; Crawshaw 2020) and zipping (He, Zhou, and Thiele 2018). By sharing the same backbone neural networks among different tasks, multi-task models can provide richer inference results than standalone models under the same cost budget. However, multi-task learning approaches lack flexibility and scalability, i.e., we need to tailor multi-task solutions case by case and re-design once the set of tasks change. In contrast, although there exists parameter redundancy between different black-box ML models, *MLink* approach can be flexibly extended. And experiments show that the accuracy of model linking is higher when given a small amount of training data.

Knowledge distillation. Following the taxonomy in the recent survey (Gou et al. 2021), knowledge distillation has three main sources of knowledge: (1) response (Ba and Caruana 2014; Hinton, Vinyals, and Dean 2015): the output of the "teacher" model; (2) feature (Chen et al. 2021): the intermediate feature maps; (3) relation (Passalis, Tzelepi, and Tefas 2020): the relations of feature maps. Mutual distillation (Zhang et al. 2018; Yao and Sun 2020) was proposed to train an ensemble of "student" models and let them learn from each other mutually. Cross-task distillation (Ye, Lu, and Zhan 2020) was proposed to train a "student" model by a "teacher" model pre-trained for another task. Model linking is complementary to knowledge distillation, i.e. model links can be built among distilled "student" models.

Redundancy filtering. Filtering redundant computation or communication is a promising way towards cost-efficient inference. Yuan et al. (Yuan et al. 2020a) proposed a reinforcement learning-based scheduler for multi-model data labelling tasks, which leverages the executed models' outputs as the hint information to schedule remaining models. Reducto (Li et al. 2020) filters out video frames by setting a threshold on feature difference between successive frames. DNNs-aware video streaming (Xie and Kim 2019; Du et al. 2020) was proposed to compress the pixels less related with inference accuracy for communication-efficient inference. FoggyCache (Guo et al. 2018) reuses cached inference results by adaptive hashing input values. Our proposed *MLink* scheduler optimizes the cost-efficiency in a novel and more direct way: predict inference results of unexecuted ML models by executed models' outputs.

Conclusion and Discussion

In this work, we propose to link black-box ML models and present the designs of model links and a collaborative multi-model inference algorithm. The comprehensive evaluations show the effectiveness of black-box model linking and the superiority of the *MLink* compared to other alternative methods. We summarize limitations and future work as follows: (1) When the semantic correlations between source and target models are low, model linking has poor output accuracy. (2) When the number of joined models is very large, pairwise model linking will become unpractical. So we will study how to smartly select models to build model links in the future. (3) Due to the small amount of parameters, the retraining cost is quite small (e.g., a vec-to-vec model link with 100-dimension input and output lengths costs 12.9s for training 100 epochs using 1k samples), so periodic retraining policy for updating *MLink* is feasible. We will study online learning and active learning mechanisms to further improve *MLink*'s adaptation to the input dynamics and reduce the updating overheads.

Acknowledgements

The research is supported by National Key R&D Program of China 2018YFB0803400, China National Natural Science Foundation with No. 61822209, No. No.61625205, No. 62132018, No. 61932016, Key Research Program of Frontier Sciences, CAS. No. QYZDY-SSW-JSC002.

References

2021. Action-Net. https://github.com/OlafenwaMoses/Action-Net.

2021. DeepSpeech. https://github.com/mozilla/DeepSpeech.

2021. Image Captioning. https://github.com/DeepRNN/image_captioning.

2021. PyGender-Voice. https://github.com/abhijeet3922/PyGender-Voice.

2021. PyTorch. https://github.com/pytorch/pytorch.

2021. TensorFlow. https://github.com/tensorflow/tensorflow.

2021. Traffic-Net. https://github.com/OlafenwaMoses/Traffic-Net.

Ba, J.; and Caruana, R. 2014. Do Deep Nets Really Need to be Deep? *Advances in Neural Information Processing Systems*, 27.

Bahdanau, D.; Cho, K. H.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Bai, H.; Wu, J.; King, I.; and Lyu, M. 2020. Few shot network compression via cross distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3203–3210.

Baltrušaitis, T.; Ahuja, C.; and Morency, L.-P. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2): 423–443.

Bentley, F.; Luvogt, C.; Silverman, M.; Wirasinghe, R.; White, B.; and Lottridge, D. 2018. Understanding the long-term use of smart speaker assistants. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3): 1–24.

Black, J.; Ellis, T.; and Rosin, P. 2002. Multi view image surveillance and tracking. In *Workshop on Motion and Video Computing, 2002. Proceedings.*, 169–174. IEEE.

Cao, Z.; Hidalgo Martinez, G.; Simon, T.; Wei, S.; and Sheikh, Y. A. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chen, D.; Mei, J.-P.; Zhang, Y.; Wang, C.; Wang, Z.; Feng, Y.; and Chen, C. 2021. Cross-layer distillation with semantic calibration.

Crawshaw, M. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dilshad, N.; Hwang, J.; Song, J.; and Sung, N. 2020. Applications and Challenges in Video Surveillance via Drone: A Brief Survey. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 728–732. IEEE.

Du, K.; Pervaiz, A.; Yuan, X.; Chowdhery, A.; Zhang, Q.; Hoffmann, H.; and Jiang, J. 2020. Server-Driven Video Streaming for Deep Learning Inference. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, 557–570.

Duan, L.; Lou, Y.; Wang, S.; Gao, W.; and Rui, Y. 2018. AI-oriented large-scale video management for smart city: Technologies, standards, and beyond. *IEEE MultiMedia*, 26(2): 8–20.

Elhoseiny, M.; Liu, J.; Cheng, H.; Sawhney, H.; and Elgammal, A. 2016. Zero-shot event detection by multimodal distributional semantic embedding of videos. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Everingham, M.; Eslami, S. A.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1): 98–136.

Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Glaeser, C.; Timm, F.; Wiesbeck, W.; and Dietmayer, K. 2020. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3): 1341–1360.

Goldblum, M.; Fowl, L.; Feizi, S.; and Goldstein, T. 2020. Adversarially robust distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3996–4003.

Gou, J.; Yu, B.; Maybank, S. J.; and Tao, D. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 1–31.

Guo, P.; Hu, B.; Li, R.; and Hu, W. 2018. FoggyCache: Cross-device approximate computation reuse. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 19–34.

Guo, Y.; Shi, H.; Kumar, A.; Grauman, K.; Rosing, T.; and Feris, R. 2019. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4805–4814.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, X.; Zhou, Z.; and Thiele, L. 2018. Multi-task zipping via layer-wise neuron sharing. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 6019–6029.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2019. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Jiang, J.; Ananthanarayanan, G.; Bodik, P.; Sen, S.; and Stoica, I. 2018. Chameleon: scalable adaptation of video analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 253–266.

Levi, G.; and Hassner, T. 2015. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 34–42.

Li, Y.; Padmanabhan, A.; Zhao, P.; Wang, Y.; Xu, G. H.; and Netravali, R. 2020. Reducto: On-Camera Filtering for Resource-Efficient Real-Time Video Analytics. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, 359–376.

Liu, S.; Lin, Y.; Zhou, Z.; Nan, K.; Liu, H.; and Du, J. 2018. On-demand deep model compression for mobile devices: A usage-driven model selection framework. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 389–400.

Marszalek, M.; Laptev, I.; and Schmid, C. 2009. Actions in context. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2929–2936. IEEE.

Ning, L.; Guan, H.; and Shen, X. 2019. Adaptive Deep Reuse: Accelerating CNN Training on the Fly. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 1538–1549.

Passalis, N.; Tzelepi, M.; and Tefas, A. 2020. Heterogeneous knowledge distillation using information flow modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2339–2348.

Redmon, J.; and Farhadi, A. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Sanh, V.; Wolf, T.; and Ruder, S. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6949–6956.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2020. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, 128(2): 336–359.

Serengil, S. I.; and Ozpinar, A. 2020. LightFace: A Hybrid Deep Face Recognition Framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 23–27. IEEE.

Shen, Z.; He, Z.; and Xue, X. 2019. Meal: Multi-model ensemble via adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4886–4893.

Song, C.; and Shmatikov, V. 2020. Overlearning Reveals Sensitive Attributes. In *International Conference on Learning Representations*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Sviridenko, M. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1): 41–43.

Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; and Liu, C. 2018. A survey on deep transfer learning. In *International conference on artificial neural networks*, 270–279. Springer.

Tieleman, T.; and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2): 26–31.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 4489–4497.

Tripuraneni, N.; Jordan, M.; and Jin, C. 2020. On the Theory of Transfer Learning: The Importance of Task Diversity. *Advances in Neural Information Processing Systems*, 33.

Wang, J.; Fang, Z.; and Zhao, H. 2020. Alignnet: A unifying approach to audio-visual alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3309–3317.

Xie, X.; and Kim, K.-H. 2019. Source compression with bounded dnn perception loss for iot edge computer vision. In *The 25th Annual International Conference on Mobile Computing and Networking*, 1–16.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057. PMLR.

Yao, A.; and Sun, D. 2020. Knowledge Transfer via Dense Cross-Layer Mutual-Distillation. In *European Conference on Computer Vision*, 294–311. Springer.

Ye, H.-J.; Lu, S.; and Zhan, D.-C. 2020. Distilling Cross-Task Knowledge via Relationship Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12396–12405.

Yuan, M.; Zhang, L.; Li, X.-Y.; and Xiong, H. 2020a. Comprehensive and efficient data labeling via adaptive model scheduling. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 1858–1861. IEEE.

Yuan, M.; Zhang, L.; Wu, Z.; and Zheng, D. 2020b. High-quality Activity-Level Video Advertising. In *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, 1–10. IEEE.

Yuksel, S. E.; Wilson, J. N.; and Gader, P. D. 2012. Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8): 1177–1193.

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4320–4328.

Zhang, Y.; and Yang, Q. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, Z.-H. 2012. *Ensemble methods: foundations and algorithms*. CRC press.