MELTR: Meta Loss Transformer for Learning to Fine-tune Video Foundation Models

Dohwan Ko[1*] Joonmyung Choi[1*] Hyeong Kyu Choi[1]
Kyoung-Woon On[2] Byungseok Roh[2] Hyunwoo J. Kim[1†]

[1]Department of Computer Science and Engineering, Korea University [2]Kakao Brain
{ikodoh, pizard, imhgchoi, hyunwoojkim}@korea.ac.kr
{kloud.ohn, peter.roh}@kakaobrain.com

Abstract

*Foundation models have shown outstanding performance and generalization capabilities across domains. Since most studies on foundation models mainly focus on the pretraining phase, a naive strategy to minimize a single task-specific loss is adopted for fine-tuning. However, such fine-tuning methods do not fully leverage other losses that are potentially beneficial for the target task. Therefore, we propose **ME**ta **L**oss **TR**ansformer (**MELTR**), a plug-in module that automatically and non-linearly combines various loss functions to aid learning the target task via auxiliary learning. We formulate the auxiliary learning as a bi-level optimization problem and present an efficient optimization algorithm based on Approximate Implicit Differentiation (AID). For evaluation, we apply our framework to various video foundation models (UniVL, Violet and All-in-one), and show significant performance gain on all four downstream tasks: text-to-video retrieval, video question answering, video captioning, and multi-modal sentiment analysis. Our qualitative analyses demonstrate that MELTR adequately 'transforms' individual loss functions and 'melts' them into an effective unified loss. Code is available at* `https://github.com/mlvlab/MELTR`.

1. Introduction

Large-scale models trained on a huge amount of data have gained attention due to their adaptability to a wide range of downstream tasks. As introduced in [1], deep learning models with the generalizability are referred to as foundation models. In recent years, several foundation models for various domains have been proposed (*e.g.*, [2, 3] for natural language processing, [4, 5] for images and language, and [6–8] for videos) and they mainly focus on *pretrain* the model often with various *multiple* pretext tasks. On the other hand, strategies for fine-tuning on downstream tasks are less explored. For instance, a recently proposed video foundation model UniVL [7] is pretrained with a *linear* combination of several pretext tasks such as text-video alignment, masked language/frame modeling, and caption generation. However, like other domains, fine-tuning is simply performed by minimizing a *single* target loss. Other potentially beneficial pretext tasks have remained largely unexplored for fine-tuning.

Auxiliary learning is a natural way to utilize multiple pretext task losses for learning. Contrary to multi-task learning that aims for generalization across tasks, auxiliary learning focuses only on the primary task by taking advantage of several auxiliary tasks. Most auxiliary learning frameworks [9, 10] manually selected auxiliary tasks, which require domain knowledge and may not always be beneficial for the primary task. To automate task selection, meta learning was integrated into auxiliary learning [11–13]. Here, the model learns to adaptively leverage multiple auxiliary tasks to assist learning of the primary task. Likewise, the pretext task losses can be unified into a single auxiliary loss to be optimized in a way that helps the target downstream task.

To this end, we propose Meta Loss Transformer (MELTR), a plug-in module that *automatically* and *non-linearly* transforms various auxiliary losses into a unified loss. MELTR built on Transformers [14] takes the target task loss as well as pretext task losses as input and learns their relationship via self-attention. In other words, MELTR learns to fine-tune a foundation model by combining the primary task with multiple auxiliary tasks, and this can be viewed as a meta-learning (or 'learning-to-learn') problem. Similar to meta-learning-based auxiliary learning frameworks [13, 15], this can be formulated as a bi-level op-

*Equal contribution.
†Corresponding author.

timization problem, which generally involves a heavy computational cost due to the second-order derivative and its inverse, *e.g.*, the inverse Hessian matrix. To circumvent this, we present an efficient training scheme that approximates the inverse Hessian matrix. We further provide empirical analyses on the time-performance trade-off of various optimization algorithms.

To verify the generality of our proposed method, we apply it to three video foundation models: UniVL [7], Violet [16], and All-in-one [17]. These foundation models are originally pretrained with a linear combination of several pretext tasks such as text-video alignment, masked language/frame modeling, and caption generation. We experiment by fine-tuning on the text-to-video retrieval, video question answering, video captioning, and multi-modal sentiment analysis task with five datasets: YouCook2, MSRVTT, TGIF, MSVD, and CMU-MOSI. For each task and dataset, our MELTR improves both previous foundation models and task-specific models by large margins. Furthermore, our extensive qualitative analyses and ablation studies demonstrate that MELTR effectively learns to non-linearly combine pretext task losses, and adaptively reweights them for the target downstream task.

To sum up, our **contributions** are threefold:

- We propose **ME**ta **Lo**ss **TR**ansformer (**MELTR**), a novel fine-tuning framework for video foundation models. We also present an efficient optimization algorithm to alleviate the heavy computational cost of bi-level optimization.
- We apply our framework to three video foundation models in four downstream tasks on five benchmark video datasets, where MELTR significantly outperforms the baselines fine-tuned with single-task and multi-task learning schemes.
- We provide in-depth qualitative analyses on how MELTR non-linearly transforms individual loss functions and combines them into an effective unified loss for the target downstream task.

2. Related work

Video foundation models. With sufficient computational power and an abundant source of data, there have been attempts to build a single large-scale foundation model that can be adapted to diverse downstream tasks. Along with the success of foundations models in the natural language processing domain [3, 18, 19] and in computer vision [2, 4, 5], video data has become another data type of interest, as it has grown in scale due to numerous internet video-sharing platforms. Accordingly, several methods to train a video foundation model have been proposed. Due to the innate multi-modality of video data, *i.e.*, a combination of visual · vocal · textual context, most works have centered around the variations of the cross-modal attention mechanism [2, 6, 7, 20–24]. In addition, as most video data lack proper labels or descriptions, contrastive learning methods were studied to learn meaningful feature representations or enhance video-text alignment in a self-supervised manner [6, 7, 24, 25].

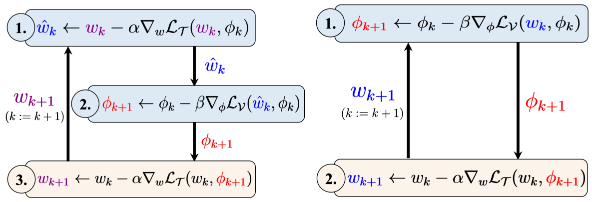
More specifically, MERLOT [8] proposed a multi-modal representation learning method for visual commonsense reasoning, which also performed well in twelve video reasoning tasks. VATT [6] introduced a multi-modal learning method via contrastive learning. The pre-trained model performed well in a variety of vision tasks from image classification to video action recognition and zero-shot video retrieval. Another representative work, UniVL [7] proposed a straightforward pre-training method with auxiliary loss functions. After fine-tuning on a specific task, the pretrained model performed outstandingly in a wide range of tasks of text-to-video retrieval, action segmentation, action step localization, video sentiment analysis, and video captioning. Other foundation models for multiple video tasks include [16, 17, 26–29].

Auxiliary learning. In order to enhance the performance of one or a multitude of primary tasks, auxiliary learning methods can be incorporated. [30] introduced Multi-task learning (MTL) to the deep neural networks by training a single model with multiple task losses to assist learning on the main task. Such a method is generally adapted to pre-train the foundation models in the self-supervised manner [16, 17, 26–29]. However, these various pretext task losses used in the pre-training phase are ignored in the fine-tuning phase, and only the primary task loss is minimized.

Recently, meta-learning methods have been introduced for auxiliary learning. [11–13] proposed a meta-learning method in which the model learns auxiliary tasks to generalize well to unseen data. In these settings, a separate subset of data is held out as the primary task, while the others are used as auxiliary tasks that aid the primary task's performance. Similar methods were adopted for computer vision tasks such as semantic segmentation [31]. Other domain applications include navigation tasks with reinforcement learning [32], or self-supervised learning methods on graph data [15].

3. Preliminaries

We briefly introduce UniVL [7], a video foundation model used as one of baselines for our learning method. We also explain two types of optimization schemes for bi-level optimization problems which commonly occur in meta-learning and auxiliary learning.



(a) Iterative Differentiation (b) Approximate Implicit Diff.

Figure 1. **Comparison of ITD and AID.** The blue boxes indicate upper-level optimization, whereas the yellow boxes refer to lower-level optimization. (a) ITD defines a fixed-point parameter $\hat{w}_k$ for upper-level optimization. To avoid visual clutter, iteration number for $\hat{w}_k$ update is set to 1. (b) AID uses a 2-step optimization scheme which optimizes the upper-level decision vector in a single step via IFT. Both algorithms can be computed efficiently with automatic differentiation [33, 34].

3.1. UniVL

UniVL [7] is a video foundation model pre-trained on the HowTo100M [35] dataset via multi-modal self-supervised learning. It is a unified video and language pre-training model for both video understanding and text generation tasks. It consists of four transformer-based modules (two single-modal encoders, a cross-modal encoder, and a decoder). It is pre-trained with five pretext tasks including the video-text joint ($\mathcal{L}_{\text{Joint}}$), the conditioned masked language model (CMLM; $\mathcal{L}_{\text{CMLM}}$), the conditioned masked frame model (CMFM; $\mathcal{L}_{\text{CMFM}}$), a video-text alignment ($\mathcal{L}_{\text{Align}}$), and the language generation task ($\mathcal{L}_{\text{Decoder}}$). UniVL trains the model simultaneously for five pretext tasks by optimizing the sum of pretext loss functions given as:

$$\mathcal{L}_{\text{UniVL}} = \mathcal{L}_{\text{Joint}} + \mathcal{L}_{\text{CMLM}} + \mathcal{L}_{\text{CMFM}} + \mathcal{L}_{\text{Align}} + \mathcal{L}_{\text{Decoder}}. \tag{1}$$

Although UniVL minimizes *multiple* pretext loss functions during pre-training, it optimizes only *one* target task loss for fine-tuning, *e.g.*, $\mathcal{L}_{\text{Align}}$ for video retrieval and $\mathcal{L}_{\text{Decoder}}$ for video captioning. That is, other loss functions, which are potentially helpful for the target downstream task, are not utilized during fine-tuning. This observation motivates our framework that automatically learns how to combine multiple losses for fine-tuning. This can be viewed as hyperparameter optimization via meta-learning.

3.2. Bi-level optimization and hypergradient approximation

Bi-level optimization commonly arises in meta-learning and hyperparameter optimization. One general class of bi-level problems is given as:

$$\begin{aligned} &\min_{\phi \in \Phi} f(\phi) := \mathcal{L}_{\mathcal{V}}(w^*(\phi), \phi) \\ &\text{s.t.}\ \ w^*(\phi) = \arg\min_{w} \mathcal{L}_{\mathcal{T}}(w, \phi), \end{aligned} \tag{2}$$

where $\phi \in \Phi$ is an upper-level decision vector, and $w \in \mathbb{R}^d$ is a lower-level decision vector. $\mathcal{L}_{\mathcal{V}} : \mathbb{R}^d \times \phi \rightarrow \mathbb{R}$ and $\mathcal{L}_{\mathcal{T}} : \mathbb{R}^d \times \phi \rightarrow \mathbb{R}$ are upper-level and lower-level loss functions, respectively. For instance, in hyperparameter optimization, ϕ is a set of hyperparameters and w is model parameters. $\mathcal{L}_{\mathcal{T}}$ and $\mathcal{L}_{\mathcal{V}}$ can be mapped to training and validation loss functions, respectively.

Grazzi *et al*. [36] have investigated bi-level optimization algorithms from the *hypergradient* approximation perspective. Hypergradient is the gradient of upper-level objective (*i.e.*, $\nabla \mathcal{L}_{\mathcal{V}}$) and it is used for updating upper-level decision vector ϕ. Popular approaches in the literature [36] can be categorized into two groups: Iterative Differentiation (ITD) and Approximate Implicit Differentiation (AID).

Iterative Differentiation [13, 37–40]. This algorithm unrolls the upper-level optimization into two stages by defining a fixed-point parameter, $\hat{w}_k(\phi)$, where k denotes the upper-level optimization step. $\hat{w}_k(\phi)$ is derived by taking iterative learning steps from w_k. Then, assuming this as a contraction mapping with respect to w_k [36], the hypergradient $\nabla \mathcal{L}_{\mathcal{V}}(w_k, \phi_k)$ can be approximated with $\nabla \mathcal{L}_{\mathcal{V}}(\hat{w}_k, \phi_k)$, as in Figure 1(a) step 2. Then, with the updated upper-level decision vector ϕ_{k+1}, model parameter w_{k+1} is computed in the final step 3.

Approximate Implicit Differentiation [41–43]. In this optimization scheme, the hypergradient $\nabla \mathcal{L}_{\mathcal{V}}(w_k, \phi_k)$ is factorized by the implicit function theorem (IFT). This is then solved with a 2-step algorithm which requires inverse Hessian computation (Figure 1(b)). In practice, the inverse Hessian matrix is generally approximated to avoid its computation overhead of $O(n^3)$. Then, similarly to ITD, model parameter w_k is updated to w_{k+1}.

4. Method

The goal of our framework is *learning to fine-tune*. We propose MEta Loss TRansformer (MELTR), a novel auxiliary learning framework that adaptively combines auxiliary losses to assist fine-tuning on the target downstream task. We formulate this as a bi-level optimization problem and present an efficient training procedure with Approximated Implicit Differentiation (AID) built on the Implicit Function Theorem (IFT). Additionally, we introduce a regularization term to alleviate *meta-overfitting* and learn a more effective combination of loss functions.

4.1. Meta Loss Transformer

Our framework generates a unified auxiliary loss function $\mathcal{L}^{\text{aux}}$ by combining auxiliary losses $\mathcal{L}_{\text{Joint}}, \mathcal{L}_{\text{Align}}, \ldots, \mathcal{L}_{\text{Decoder}}$. In other words, our framework takes loss values from multiple auxiliary tasks and converts them to a new combined loss value as shown in Figure 2. In order to leverage the relationship between

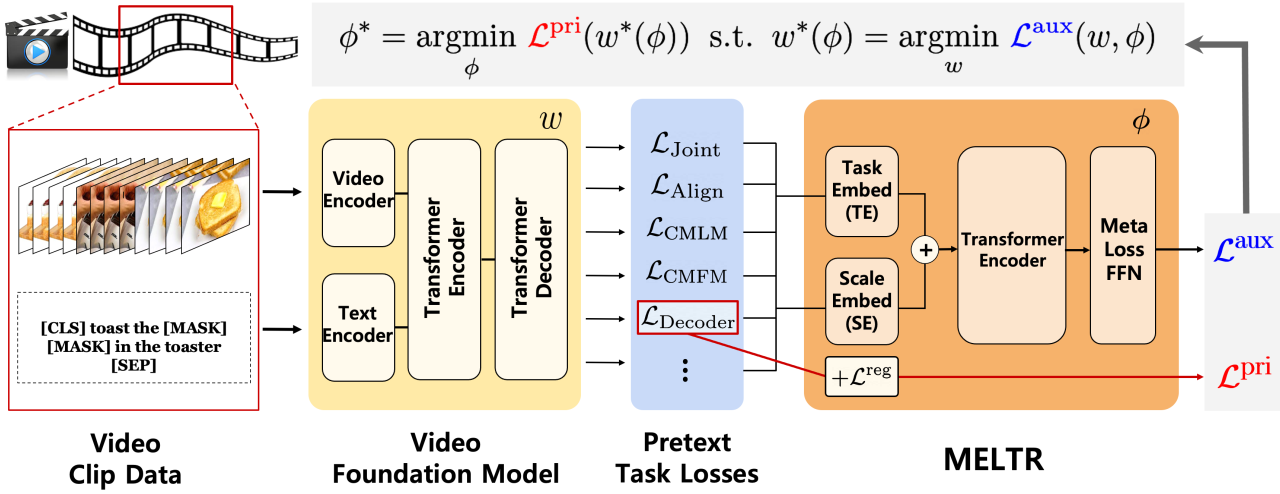


Figure 2. **Overall architecture.** The Meta Loss Transformer (MELTR) is a plug-in module for meta auxiliary learning. The auxiliary pretext task losses derived from the video foundation model (*e.g.*, UniVL [7]) are input to MELTR, which is a transformer-based module that non-linearly aggregates the loss values from different tasks. The module is optimized to help learning of the primary task. This figure illustrates the case when video captioning ($\mathcal{L}_{\text{Decoder}}$) is the primary task.

primary and auxiliary tasks, we adopt the Transformer [14] architecture.

Let $\mathcal{F}(\cdot; w)$ denote a backbone foundation model parameterized by w. For t-th task, given input data x and its label y_t, the loss value ℓ_t is defined as:

$$\ell_t = \mathcal{L}_t(\mathcal{F}(x; w), y_t), \tag{3}$$

where $\mathcal{L}_t$ is a loss function for t-th task. With loss values $\boldsymbol{\ell} = [\ell_0, \dots, \ell_T]$ from the primary task $t = 0$ and auxiliary tasks $\{t = 1, \dots t = T\}$, our framework MELTR learns a unified auxiliary loss function defined as:

$$\mathcal{L}^{\text{aux}} := \text{MELTR}(\boldsymbol{\ell}; \phi), \tag{4}$$

where $\text{MELTR}(\cdot; \phi)$ is a transformer-based neural network parameterized by ϕ, which are meta-parameters in our meta-learning formulation. In order to feed loss values, $\ell_0, \dots, \ell_T$ to a Multi-head Self-attention layer, we transform a scalar loss value into the scale embedding (**SE**) and the task embedding (**TE**). Each auxiliary loss value is first projected to a d-dimensional vector via $\textbf{SE}(\cdot)$, which is an MLP layer with a non-linear activation. Similarly, we adopt a learnable embedding layer for **TE**, which plays the role of positional encodings. Then, $\textbf{SE} : \mathbb{R} \rightarrow \mathbb{R}^d$ and $\textbf{TE} : \{0, \dots, T\} \rightarrow \mathbb{R}^d$ are defined as:

$$\textbf{SE}(\ell) := \text{MLP}(\ell), \text{ and } \textbf{TE}(t) := \text{Embedding}(t). \tag{5}$$

Then, the scale and task embeddings are summed to construct an input token. The input embeddings are self-attended and finally pooled to a scalar loss value, $\text{MELTR}(\boldsymbol{\ell}; \phi) \in \mathbb{R}$, by considering both the *loss scale* and the *task information*. The overall architecture with the UniVL backbone is illustrated in Figure 2.

However, when *meta-data* (or a validation dataset) is small, meta-learning often suffers *meta-overfitting* [44, 45]. In other words, meta-parameter ϕ may overfit to the primary task performance on small validation data. To address this problem, we additionally introduce a regularization term $\mathcal{L}^{\text{reg}}$ given as:

$$\mathcal{L}^{\text{reg}} = \left| \text{MELTR}(\boldsymbol{\ell}; \phi) - \sum_{t=0}^{T} \ell_t \right|. \tag{6}$$

This encourages the learned loss $\text{MELTR}(\boldsymbol{\ell}; \phi)$ to stay within a reasonable range. Then, the primary task loss $\mathcal{L}^{\text{pri}}$, and the unified auxiliary loss $\mathcal{L}^{\text{aux}}$ are defined as follows:

$$\mathcal{L}^{\text{pri}} = \mathcal{L}_0 + \gamma \mathcal{L}^{\text{reg}}, \quad \mathcal{L}^{\text{aux}} = \text{MELTR}(\boldsymbol{\ell}; \phi), \tag{7}$$

where γ is a regularization strength and $\mathcal{L}_0$ is the original supervised loss for the target downstream task. For example, if $\mathcal{L}_{\text{Align}}$ is selected as the primary loss for the text-to-video retrieval task, then $\mathcal{L}_0 = \mathcal{L}_{\text{Align}}$ and all other tasks are considered as pretext tasks, *i.e.*, $\boldsymbol{\ell} = [\mathcal{L}_{\text{Align}}, \mathcal{L}_{\text{Joint}}, \mathcal{L}_{\text{CMLM}}, \mathcal{L}_{\text{CMFM}}, \mathcal{L}_{\text{Decoder}}]$. Note that the primary loss itself is also included in the list of input loss functions.

4.2. Objective function and optimization

MELTR learns how to fine-tune a model by non-linearly combining the auxiliary losses. This can be viewed as hyperparameter optimization, which can be formulated as a bi-level optimization given as:

$$\begin{aligned} \phi^* &= \underset{\phi}{\arg\min}\ \mathcal{L}^{\text{pri}}(w^*(\phi)) \\ \text{s.t. } w^*(\phi) &= \underset{w}{\arg\min}\ \mathcal{L}^{\text{aux}}(w, \phi), \end{aligned} \tag{8}$$

where ϕ denotes the (meta) parameter of MELTR, and w denotes the parameters of our backbone foundation model. Then, we adopt one variant of the Approximate Implicit Differentiation (AID) scheme to optimize (8). Specifically, to optimize (8), we first factorize the *hypergradient*, which is the gradient of $\mathcal{L}^{\text{pri}}$ with respect to ϕ as $\nabla_\phi \mathcal{L}^{\text{pri}} = \nabla_w \mathcal{L}^{\text{pri}} \cdot \nabla_\phi w^*$, where $\nabla_\phi w^* = -(\nabla_w^2 \mathcal{L}^{\text{aux}})^{-1} \cdot \nabla_\phi \nabla_w \mathcal{L}^{\text{aux}}$ by the implicit function theorem (IFT). Then, the hypergradient can be written as:

$$\nabla_\phi \mathcal{L}^{\text{pri}}(w^*(\phi)) = -\nabla_w \mathcal{L}^{\text{pri}} \cdot \left(\nabla_w^2 \mathcal{L}^{\text{aux}}\right)^{-1} \cdot \nabla_\phi \nabla_w \mathcal{L}^{\text{aux}}. \tag{9}$$

The evaluation of hypergradient entails the computation of the inverse of second-order derivatives. In the literature [12, 41], to accelerate the computation, the Neumann series is commonly adopted as:

$$\left(\nabla_w^2 \mathcal{L}^{\text{aux}}\right)^{-1} = \lim_{i \to \infty} \sum_{j=0}^{i} \left(\mathrm{I} - \nabla_w^2 \mathcal{L}^{\text{aux}}\right)^j. \tag{10}$$

In practice, the summation of an infinite series in (10) is approximated by a finite sequence. For instance, the number of iterations i is usually truncated to a small integer (*e.g.*, $i = 3$ in [12]) in exchange for slight performance decay. However, this still requires considerable amount of time in the iterative computation of the Hessian matrix in (10). We further simplify it by approximating the Hessian matrix in (9) as the identity matrix I. Then, our approximated gradient is given as follows:

$$\nabla_\phi \mathcal{L}^{\text{pri}}(w^*(\phi)) \approx -\nabla_w \mathcal{L}^{\text{pri}} \cdot \nabla_\phi \nabla_w \mathcal{L}^{\text{aux}}. \tag{11}$$

This completely removes the need for computation of the inverse Hessian matrix, which otherwise would have required a time complexity of $O(n^3)$. In our experiments, we observe that there is no significant degradation in terms of the performance of a fine-tuned model, see in Section 5.3.

Finally, with the approximated hypergradient (11), we utilize one variant of the AID scheme as an efficient optimization algorithm for MELTR. We first optimize w for K steps by:

$$w^{(k+1)} = w^{(k)} - \alpha \cdot \nabla_w \mathcal{L}^{\text{aux}}. \tag{12}$$

After K steps of (12), we then optimize for ϕ with:

$$\begin{aligned} \phi^* &= \phi - \beta \cdot \nabla_\phi \mathcal{L}^{\text{pri}}(w^{(K)}(\phi)) \\ &= \phi + \beta \cdot \left(\nabla_w \mathcal{L}^{\text{pri}} \cdot \nabla_\phi \nabla_w \mathcal{L}^{\text{aux}}\right), \end{aligned} \tag{13}$$

where α and β are the learning rates of the backbone foundation model and MELTR, respectively. The pseudo-code of our training scheme is provided in Algorithm 1.

Algorithm 1 MELTR optimization algorithm

Inputs: w, ϕ

Parameters: learning rate (α, β), regularization coefficient γ, inner iter K

while not converged **do**
  for $k = 1$ **to** K **do**
    $\ell_t = \mathcal{L}_t(\mathcal{F}(x; w), y_t) \quad \forall t \in [0, T]$
    $\mathcal{L}^{\text{aux}} \leftarrow \text{MELTR}(\boldsymbol{\ell}; \phi)$
    $w \leftarrow w - \alpha \cdot \nabla_w \mathcal{L}^{\text{aux}}\Big|_{w,\phi}$
  end for
  $\ell_t = \mathcal{L}_t(\mathcal{F}(x; w), y_t) \quad \forall t \in [0, T]$
  $\mathcal{L}^{\text{aux}} \leftarrow \text{MELTR}(\boldsymbol{\ell}; \phi)$
  $\mathcal{L}^{\text{pri}} \leftarrow \ell_0 + \gamma \cdot |\text{MELTR}(\boldsymbol{\ell}; \phi) - \sum_t \ell_t|$
  $\phi \leftarrow \phi + \beta \cdot \nabla_w \mathcal{L}^{\text{pri}}\Big|_{w,\phi} \cdot \nabla_\phi \nabla_w \mathcal{L}^{\text{aux}}\Big|_{w,\phi}$
end while
return w

5. Experiments

To verify the effectiveness of our method, we apply it to multiple video foundation models (UniVL [7], Violet [16], All-in-one [17]), and evaluate them on four downstream tasks: text-to-video retrieval, video question answering, video captioning, and multimodal sentiment analysis. For the tasks, we use five benchmark datasets: YouCook2 [46], MSRVTT [47], TGIF-QA [48], MSVD-QA [49], CMU-MOSI [50]. We conduct experiments and analyze the results to answer the following research questions:

Q1. Does the learned combination of auxiliary losses benefit the primary task?

Q2. What does MELTR learn from auxiliary learning?

Q3. Is the proposed optimization method efficient for MELTR?

Datasets. For video retrieval, we use YouCook2 and MSRVTT. For video question answering, TGIF-QA and MSVD-QA datasets are used, and YouCook2 and MSRVTT are used for video captioning. Finally, we use CMU-MOSI for multi-modal sentiment analysis. Further dataset details are provided in the supplement.

Implementation details. MELTR is adapted to UniVL [7], Violet [16], and All-in-one [17] for main experiments and we conduct ablation studies and qualitative analyses on UniVL. As for UniVL, we use five auxiliary loss functions ($\mathcal{L}_{\text{Joint}}$, $\mathcal{L}_{\text{Align}}$, $\mathcal{L}_{\text{CMLM}}$, $\mathcal{L}_{\text{CMFM}}$, and $\mathcal{L}_{\text{Decoder}}$), which were introduced in Section 3.1. We additionally adopt three advanced auxiliary loss functions, $\mathcal{L}_{\text{M-Joint}}$, $\mathcal{L}_{\text{M-Align}}$, and $\mathcal{L}_{\text{M-Decoder}}$, which leverage masked language and masked visual features obtained by converting some of the language or visual tokens into [MASK] or random tokens, along with the five objectives described above. For text-to-video retrieval and video captioning, $\mathcal{L}_{\text{Align}}$ and $\mathcal{L}_{\text{Decoder}}$ are used

Table 1. **Text-to-Video retrieval on YouCook2.** UniVL-Joint and UniVL-Align denote the model fine-tuned with the $\mathcal{L}_{\text{Joint}}$ and $\mathcal{L}_{\text{Align}}$, respectively. MELTR is applied to the UniVL-Align. MELTR$^-$ refers to MELTR without the regularization term $\mathcal{L}^{\text{reg}}$.

Models	R@1↑	R@5↑	R@10↑	MedR↓
HGLMM-FV-CCA [51]	4.6	21.6	14.3	75
HowTo100M [35]	8.2	35.3	24.5	24
ActBERT [29]	9.6	26.7	38.0	19
MIL-NCE [52]	15.1	38.0	51.2	10
COOT [53]	16.7	40.2	25.3	9
TACo [24]	29.6	59.7	72.7	9
VideoCLIP [54]	32.2	62.6	**75.0**	-
UniVL-Joint [7]	22.2	52.2	66.2	5
UniVL-Align [7]	28.9	57.6	70.0	4
UniVL + **MELTR$^-$**	33.4	62.5	73.3	**3**
UniVL + **MELTR**	**33.7**	**63.1**	74.8	**3**

Table 2. **Text-to-Video retrieval on MSRVTT.**

Models	MSRVTT-7k				MSRVTT-9k			
	R@1↑	R@5↑	R@10↑	MedR↓	R@1↑	R@5↑	R@10↑	MedR↓
MIL-NCE [52]	9.9	24.0	32.4	29.5	-	-	-	-
JSFusion [55]	10.2	31.2	43.2	13	-	-	-	-
HowTo100M [35]	14.9	40.2	52.8	9	-	-	-	-
HERO [26]	16.8	43.4	57.7	-	-	-	-	-
ClipBERT [56]	22.2	46.8	59.9	6	-	-	-	-
MMT [20]	-	-	-	-	26.6	57.1	69.6	4
T2VLAD [57]	-	-	-	-	29.5	59.0	70.1	4
TACo [24]	19.2	44.7	57.2	7	28.4	57.8	71.2	4
VideoCLIP [54]	-	-	-	-	30.9	55.4	66.8	-
Frozen [58]	-	-	-	-	32.5	61.5	71.2	3
UniVL-Joint [7]	20.6	49.1	62.9	6	27.2	**55.7**	**68.7**	**4**
UniVL-Align [7]	21.2	49.6	63.1	6	-	-	-	-
UniVL + **MELTR**	**28.5**	**55.5**	**67.6**	**4**	**31.1**	**55.7**	68.3	**4**
Violet [16]	31.7	60.1	74.6	**3**	34.5	63.0	73.4	-
Violet + **MELTR**	**33.6**	**63.7**	**77.8**	**3**	**35.5**	**67.2**	**78.4**	**3**
All-in-one [17]	34.4	65.4	75.8	-	37.9	68.1	77.1	-
All-in-one + **MELTR**	**38.6**	**74.4**	**84.7**	-	**41.3**	**73.5**	**82.5**	-

as the primary loss functions, respectively. Further implementation details including Violet and All-in-one are in the supplement.

5.1. Evaluation on downstream tasks

We here answer **Q1** (the effectiveness of MELTR) by applying our framework to fine-tune the pretrained foundation models on various downstream tasks: text-to-video retrieval, video question answering, video captioning, and multi-modal sentiment analysis.

Text-to-Video retrieval. We evaluate text-to-video retrieval task performance on YouCook2 and MSRVTT. Table 1 shows that our method outperforms all the baseline models by plugging MELTR in the standard UniVL. Specifically, the R@1 is improved by 4.8% compared to UniVL, and 1.5% compared to the previous SOTA, VideoCLIP [24]. Also, METLR with the regularization term $\mathcal{L}^{\text{reg}}$ improves all the performance metrics compared to MELTR$^-$, which

Table 3. **Video question answering on TGIF-QA and MSVD-QA.**

Models	TGIF-QA			MSVD-QA
	Action	Transition	Frame	
HME [59]	73.9	77.8	53.8	33.7
HCRN [60]	75.0	81.4	55.9	36.1
QueST [61]	75.9	81.0	59.7	36.1
ClipBERT [62]	82.9	87.5	59.4	-
Violet [16]	92.5	95.7	62.3	47.9
Violet + **MELTR**	**95.4**	**97.5**	**63.4**	**51.7**

Table 4. **Video captioning on YouCook2.** V refers to the video-only input setting, and V+T the multi-modal setting with video and transcript inputs.

Models	Modality	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
EMT [63]	V	7.53	4.38	11.55	27.44	38
CBT [27]	V	-	5.12	12.97	30.44	64
ActBERT [29]	V	8.66	5.41	13.30	30.56	65
VideoBERT [28]	V	6.33	3.81	10.81	27.14	47
COOT [53]	V	17.97	11.30	19.85	37.94	57
VideoBERT [28]	V+T	7.59	4.33	11.94	28.80	55
DPC [64]	V+T	7.60	2.76	18.08	-	-
AT+Video [65]	V+T	-	9.01	17.77	36.65	112
UniVL [7]	V	16.46	11.17	17.57	40.09	127
UniVL + **MELTR**	V	17.35	11.98	18.19	41.28	138
UniVL [7]	V+T	23.87	17.35	22.35	46.52	181
UniVL + **MELTR**	V+T	**24.12**	**17.92**	**22.56**	**47.04**	**190**

does not use $\mathcal{L}^{\text{reg}}$. This optional regularization term confines the loss value to a reasonable bound, which prevents meta-overfitting.

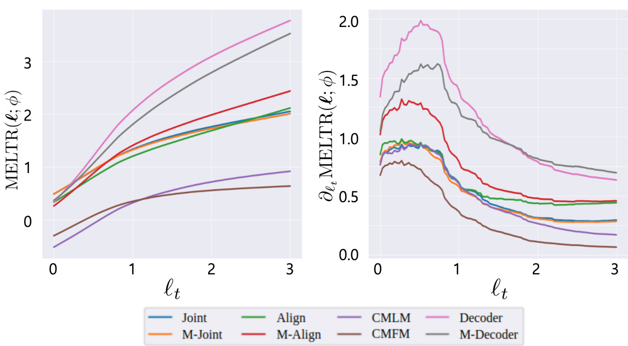
In Table 2, our model outperforms all the baselines including foundation models and task-specific methods in all the retrieval metrics. Specifically, MELTR improved three baseline foundation models: UniVL, Violet, and All-in-one. For each model, R@1 is improved by a margin of 7.3%, 1.9%, and 4.2% on MSRVTT-7k by plugging in MELTR. The R@1 is also improved by a large margin of 4.8%, 7.3%, and 3.9% respectively on YouCook2, MSRVTT-7k, and MSRVTT-9k as well, compared to the standard UniVL variants denoted UniVL-Joint or UniVL-Align.

Video question answering. We experiment video question answering on TGIF-QA and MSVD-QA in Table 3. Plugging MELTR in the foundation model outperforms all the baselines. Especially in MSVD-QA, MELTR obtains a large margin of 3.8% improvement over the standard Violet.

Video captioning. In Table 4 and Table 5, we evaluate video captioning task performance on YouCook2 and MSRVTT. In the case of YouCook2, we conduct experiments on the 'video-input-only' setting and additionally experiment on 'video + text (transcript)' input, following previous works. MELTR outperforms all the baseline models, in terms of all metrics. In the case of MSRVTT, the performance of MELTR significantly improves BLEU scores,

Table 5. **Video captioning on MSRVTT-full.** * refers to the experimental results reported in the official github.

Models	Modality	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
PickNet [66]	V	-	35.6	26.8	58.2	41.0
PickNet [66]	V+T	-	38.9	27.2	59.5	42.1
MARN [67]	V	-	40.4	28.1	60.7	47.1
SibNet [68]	V	-	40.9	27.5	60.2	47.5
OA-BTG [69]	V	-	41.4	28.2	-	46.9
POS-VCT [70]	V	-	42.3	**29.7**	**62.8**	49.1
ORG-TRL [71]	V	-	43.6	28.8	62.1	50.9
UniVL* [7]	V	53.42	41.79	28.94	60.78	50.04
UniVL + **MELTR**	V	**55.88**	**44.17**	29.26	62.35	**52.77**



(a) (b)

Figure 3. **MELTR($\boldsymbol{\ell};\phi$) and ∂_{ℓ_t}MELTR($\boldsymbol{\ell};\phi$).** The combined loss MELTR($\boldsymbol{\ell};\phi$) in (a) and the partial derivative of MELTR($\boldsymbol{\ell};\phi$) with respect to ℓ_t in (b) are visualized by changing each pretext task loss from 0 to 3, while other losses are fixed to their average values. In the video captioning task, the Decoder loss and the Masked-Decoder loss rendered the highest MELTR($\boldsymbol{\ell};\phi$) and ∂_{ℓ_t}MELTR($\boldsymbol{\ell};\phi$) values overall.

which are the major performance metric, BLEU-4.

Multi-modal sentiment analysis. We also experiment the multi-modal sentiment analysis task on CMU-MOSI. Table 6 shows that MELTR surpasses all the baselines. These experimental results indicate that MELTR is successful in adaptively combining the auxiliary losses across backbone model architectures on various tasks.

5.2. Analysis on MELTR

We discuss **Q2** by analyzing how MELTR combines the losses. We first analyze the non-linear relationship between the input and output loss values of MELTR, and examine how MELTR adaptively re-weights the auxiliary tasks. Note, we use MELTR trained for the video captioning task on YouCook2 for these analyses, and abbreviate $\partial_{\ell_t}\text{MELTR}(\boldsymbol{\ell};\phi) := \frac{\partial}{\partial \ell_t}\text{MELTR}(\boldsymbol{\ell};\phi)$ hereafter.

Non-linear loss transformation. Figure 3(a) shows that all auxiliary losses are positively correlated with the output loss. We can also observe that the output and input are non-linearly correlated. On the other hand, Figure 3(b) shows that ∂_{ℓ_t}MELTR($\boldsymbol{\ell};\phi$) have relatively higher values around $\ell_t = 0.5$ and the gradient becomes smaller as ℓ_t increases. This indicates that MELTR guides the learner to focus on reasonably challenging samples and if the loss is too large, it becomes less sensitive (*i.e.*, too large an input loss is interpreted as noise and it tends to be downweighted). Also, given the primary task loss $\mathcal{L}_{\text{Decoder}}$ for video captioning, the MELTR is more sensitive to the change of $\mathcal{L}_{\text{Decoder}}$ and $\mathcal{L}_{\text{M-Decoder}}$ rather than $\mathcal{L}_{\text{CMFM}}$. In other words, MELTR learned that text generation task losses ($\mathcal{L}_{\text{Decoder}}$ and $\mathcal{L}_{\text{M-Decoder}}$) are more relevant to the video captioning than masked frame generation ($\mathcal{L}_{\text{CMFM}}$).

Table 6. **Multimodal sentiment analysis on CMU-MOSI.** BA, F1, MAE, and Corr are binary accuracy, F1 score, mean absolute error, and Pearson correlation coefficient, respectively.

Models	BA↑	F1↑	MAE↓	Corr↑
MulT	83.0	82.8	0.870	0.698
FMT	83.5	83.5	0.837	0.744
UniVL	84.6	84.6	0.781	0.767
UniVL + **MELTR**	**85.3**	**85.4**	**0.759**	**0.789**

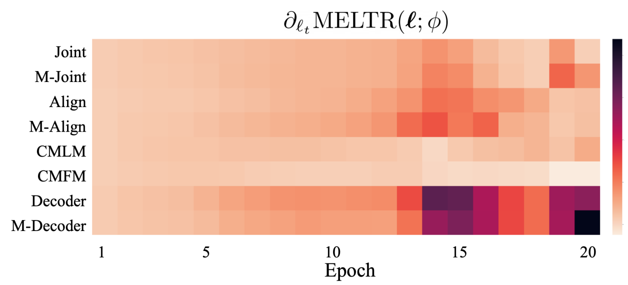


Figure 4. **Illustration of ∂_{ℓ_t}MELTR.** The ∂_{ℓ_t}MELTR($\boldsymbol{\ell};\phi$) values for the pretext task losses are plotted for each epoch, when training for video captioning. The gradients are computed by taking the average for each data sample per epoch. In the first few epochs, all task losses possess a similar scale of gradients. As training continues, the most relevant losses, Decoder and M-Decoder, receives larger gradients, while the least relevant, CMFM, has the smallest gradient value.

Adaptive task re-weighting. As an extension of the above observation, we visualize ∂_{ℓ_t}MELTR($\boldsymbol{\ell};\phi$) for each epoch in Figure 4. At the beginning of training, MELTR equally takes into account all the auxiliary tasks. As training proceeds, MELTR evaluates that $\mathcal{L}_{\text{Decoder}}$ and $\mathcal{L}_{\text{M-Decoder}}$ are effective for the primary loss $\mathcal{L}_{\text{Decoder}}$, while $\mathcal{L}_{\text{CMFM}}$ is relatively less beneficial, if not harmful. This is consistent with our observation in Figure 3 since $\mathcal{L}_{\text{Decoder}}$ and $\mathcal{L}_{\text{M-Decoder}}$ mainly conduct the text generation task while $\mathcal{L}_{\text{CMFM}}$ is for masked frame generation.

In Table 7, we also compare MELTR with five manually designed multi-task learning schemes, each combining the task losses with different linear coefficients. First, by comparing (A) and (B), an auxiliary loss $\mathcal{L}_{\text{M-Decoder}}$ assists

Table 7. **Comparison of various multi-task learning schemes.** We compare MELTR with manually designed five multi-task learning schemes: (A) adopts only $\mathcal{L}_{\text{Decoder}}$, (B) adopts both $\mathcal{L}_{\text{Decoder}}$ and $\mathcal{L}_{\text{M-Decoder}}$ which are useful for video captioning based on our observation, (C) fixes all the coefficients to 1, (D) drops only $\mathcal{L}_{\text{CMFM}}$ which is useless for video captioning from (C) based on our observation, and (E) re-weights the loss coefficients based on task importance, contrary to (D).

Models	Coefficient of each task								Video captioning on YouCook2				
	$\mathcal{L}_{\text{Joint}}$	$\mathcal{L}_{\text{M-Joint}}$	$\mathcal{L}_{\text{Align}}$	$\mathcal{L}_{\text{M-Align}}$	$\mathcal{L}_{\text{CMLM}}$	$\mathcal{L}_{\text{CMFM}}$	$\mathcal{L}_{\text{Decoder}}$	$\mathcal{L}_{\text{M-Decoder}}$	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
(A)	0	0	0	0	0	0	1	0	22.79	16.54	21.73	45.85	1.78
(B)	0	0	0	0	0	0	1	1	23.42	17.14	22.27	46.65	1.85
(C)	1	1	1	1	1	1	1	1	21.72	15.93	20.89	45.16	1.79
(D)	1	1	1	1	1	0	1	1	21.99	16.10	21.09	45.35	1.85
(E)	1	1	1	1	1	0	8	8	23.31	17.23	21.98	46.26	1.85
MELTR	ADAPTIVE								**24.12**	**17.92**	**22.56**	**47.04**	**1.90**

learning of the video captioning task. However, (A) and (C) demonstrate that multi-task learning is not always beneficial, and it sometimes hinders fine-tuning if the auxiliary tasks include harmful task losses. By dropping $\mathcal{L}_{\text{CMFM}}$ from (C), the model performance is slightly improved in (D). Interestingly, this matches our observation that $\mathcal{L}_{\text{CMFM}}$ is disadvantageous for video captioning (Figure 4). Furthermore, (E) outperforms (D), implying that re-weighting among the auxiliary tasks can be beneficial for multi-task learning. Finally, our MELTR surpasses all the multi-task learning schemes above. These experimental results indicate that MELTR effectively learns to fine-tune by adaptively re-weighting the auxiliary tasks, compared to the coarse and heuristically designed multi-task learning schemes.

5.3. Efficient optimization algorithm

We also discuss the optimization algorithms for training MELTR to answer the last question **Q3**. We compare various bi-level optimization algorithms based on ITD or AID schemes. To analyze the efficiency, we measure the latency of an epoch and the performance on the text-to-video retrieval task with MSRVTT-7k. We identically use the MELTR module as the loss combining network with all optimization algorithms for fair comparisons. We also compare with the multi-task learning setting where the model is fine-tuned with a linearly summed loss.

In Table 8, the multi-task learning (MTL) method is faster than meta-learning-based algorithms (denoted by '**MELTR** + α') since MTL is formulated as a uni-level optimization problem. We observe that all MELTR with various bi-level optimization schemes outperform a Multi-task Learning in terms of the target task performance R@1. Among the bi-level optimization schemes, our training scheme denoted as **MELTR + AID-FP-Lite**† introduces only 4.9% overhead in training time than multi-task learning, while improving performance by 2.4%. This is a significant improvement considering that Meta-Weight Net (ITD) takes longer than twice the time required by Multi-task Learning and AID-FP-Lite. Our optimization in Algorithm 1, which approximates $\nabla^2_w \mathcal{L}^{\text{aux}}$ in (9) with the identity matrix I, is the fastest bi-level optimization scheme in Table 8 while achieving a strong R@1 performance.

Table 8. **Efficiency comparison of optimization algorithms.** R@1 scores evaluated on MSRVTT-7k for video retrieval are recorded. Multi-task learning simultaneously trains all tasks with even loss weights. CG and FP are abbreviations of conjugate gradient and fixed-point optimization. In terms of time costs, average training time per epoch is reported. † refers to our optimization algorithm which approximates $\nabla^2_w \mathcal{L}^{\text{aux}}$ as the identity matrix I.

Method	Opt. Scheme	R@1	Time
Multi-task Learning	-	26.1 (+0.0)	547 (+0.0%)
MELTR + Meta-Weight Net [13]	ITD	27.3 (+1.2)	1,296 (+136.9%)
MELTR + StocBIO [72]	N/A	26.8 (+0.7)	686 (+25.4%)
MELTR + CG	AID-CG	28.0 (+1.9)	624 (+14.1%)
MELTR + AuxiLearn [12]	AID-FP	27.9 (+1.8)	638 (+16.6%)
MELTR + AID-FP-Lite†	AID-FP	28.5 (+2.4)	574 (+4.9%)

6. Conclusion

We proposed Meta Loss Transformer (MELTR), an auxiliary learning framework that learns to fine-tune video foundation models. MELTR learns to integrate various pretext task losses into one loss function to boost the performance of the target downstream task. Our qualitative analysis demonstrates that MELTR improves the performance of the primary task by considering the type of task and the scale of the loss value. The proposed training procedure built on AID-FP-Lite with a simple approximation of the inverse Hessian matrix achieved the efficiency without a significant performance loss. By plugging MELTR into various foundation models, our method outperformed state-of-the-art video foundation models as well as task-specific models on a wide range of downstream tasks.

Acknowledgments. This work was partly supported by ICT Creative Consilience program (IITP-2023-2020-0-01819) supervised by the IITP; Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (23ZS1200, Fundamental Technology Research for Human-Centric Autonomous Intelligent Systems); and KaKaoBrain corporation.

References

[1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models, 2021. 1

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 1, 2

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Mateusz Sigler, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 1, 2

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2

[5] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1, 2

[6] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. In *NeurIPS*, 2021. 1, 2

[7] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation, 2020. 1, 2, 3, 4, 5, 6, 7

[8] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. In *NeurIPS*, 2021. 1, 2

[9] Lukas Liebel and Marco Körner. Auxiliary tasks in multi-task learning. *arXiv preprint arXiv:1805.06334*, 2018. 1

[10] Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition. In *Interspeech*, 2017. 1

[11] Shikun Liu, Andrew Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. In *NeurIPS*, 2019. 1, 2

[12] Aviv Navon, Idan Achituve, Haggai Maron, Gal Chechik, and Ethan Fetaya. Auxiliary learning by implicit differentiation. In *ICLR*, 2021. 1, 2, 5, 8

[13] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019. 1, 2, 3, 8

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 4

[15] Dasol Hwang, Jinyoung Park, Sunyoung Kwon, KyungMin Kim, Jung-Woo Ha, and Hyunwoo J Kim. Self-supervised auxiliary learning with meta-paths for heterogeneous graphs. In *NeurIPS*, 2020. 1, 2

[16] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021. 2, 5, 6

[17] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *arXiv preprint arXiv:2203.07303*, 2022. 2, 5, 6

[18] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. 2

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NACCL*, 2019. 2

[20] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020. 2, 6

[21] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *ICCVW*, 2021. 2

[22] Reuben Tan, Bryan A. Plummer, Kate Saenko, Hailin Jin, and Bryan Russell. Look at what i'm doing: Self-supervised spatial grounding of narrations in instructional videos. In *NeurIPS*, 2021. 2

[23] Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. Multi-modality cross attention network for image and sentence matching. In *CVPR*, 2020. 2

[24] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. Taco: Token-aware cascade contrastive learning for video-text alignment. In *ICCV*, 2021. 2, 6

[25] Haofei Kuang, Yi Zhu, Zhi Zhang, Xinyu Li, Joseph Tighe, Sören Schwertfeger, Cyrill Stachniss, and Mu Li. Video contrastive learning with global context. In *ICCVW*, 2021. 2

[26] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+language omni-representation pre-training, 2020. 2, 6

[27] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer, 2019. 2, 6

[28] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *CoRR*, 2019. 2, 6

[29] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020. 2, 6

[30] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2

[31] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, Ferdous Sohel, and Dan Xu. Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In *ICCV*, 2021. 2

[32] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectnav, 2021. 2

[33] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. In *JMLR*, 2018. 3

[34] Evaluating Derivatives. Principles and techniques of algorithmic differentiation. *SIAM*, 2000. 3

[35] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 3, 6

[36] Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *ICML*, 2020. 3

[37] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 3

[38] Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *ICML*, 2017. 3

[39] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *ICML*, 2018. 3

[40] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *ICML*, 2015. 3

[41] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *AISTATS*, 2020. 3, 5

[42] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *ICML*, 2016. 3

[43] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *NeurIPS*, 2019. 3

[44] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *ICLR*, 2019. 4

[45] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *ICML*, 2019. 4

[46] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 5

[47] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 5

[48] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 5

[49] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *MM*, 2017. 5

[50] Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*, 2016. 5

[51] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015. 6

[52] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*, 2020. 6

[53] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. Coot: Cooperative hierarchical transformer for video-text representation learning. In *NeurIPS*, 2020. 6

[54] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *EMNLP*, 2021. 6

[55] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018. 6

[56] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021. 6

[57] Xiaohan Wang, Linchao Zhu, and Yi Yang. T2vlad: global-local sequence alignment for text-video retrieval. In *CVPR*, 2021. 6

[58] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 6

[59] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *CVPR*, 2019. 6

[60] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *CVPR*, 2020. 6

[61] Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *AAAI*, 2020. 6

[62] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021. 6

[63] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018. 6

[64] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. Dense procedure captioning in narrated instructional videos. In *ACL*, 2019. 6

[65] Jack Hessel, Bo Pang, Zhenhai Zhu, and Radu Soricut. A case study on combining asr and visual features for generating instructional video captions. *arXiv preprint arXiv:1910.02930*, 2019. 6

[66] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In *ECCV*, 2018. 7

[67] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. *CVPR*, 2019. 7

[68] Sheng Liu, Zhou Ren, and Junsong Yuan. Sibnet: Sibling convolutional encoder for video captioning. *TPAMI*, 2020. 7

[69] Junchao Zhang and Yuxin Peng. Object-aware aggregation with bidirectional temporal graph for video captioning. In *CVPR*, 2019. 7

[70] Jingyi Hou, Xinxiao Wu, Wentian Zhao, Jiebo Luo, and Yunde Jia. Joint syntax representation learning and visual cue translation for video captioning. In *ICCV*, 2019. 7

[71] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, 2020. 7

[72] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *ICML*, 2021. 8