HetMAML: Task-Heterogeneous Model-Agnostic Meta-Learning for Few-Shot Learning Across Modalities

Jiayi Chen, Aidong Zhang
University of Virginia
{jc4td, aidong}@virginia.edu

ABSTRACT
Most of existing gradient-based meta-learning approaches to few-shot learning assume that all tasks have the same input feature space. However, in the real world scenarios, there are many cases that the input structures of tasks can be different, that is, different tasks may vary in the number of input modalities or data types. Existing meta-learners cannot handle the heterogeneous task distribution (HTD) as there is not only global meta-knowledge shared across tasks but also type-specific knowledge that distinguishes each type of tasks. To deal with task heterogeneity and promote fast within-task adaptions for each type of tasks, in this paper, we propose HetMAML, a task-heterogeneous model-agnostic meta-learning framework, which can capture both the type-specific and globally shared knowledge and can achieve the balance between knowledge customization and generalization. Specifically, we design a multi-channel backbone module that encodes the input of each type of tasks into the same length sequence of modality-specific embeddings. Then, we propose a task-aware iterative feature aggregation network which can automatically take into account the context of task-specific input structures and adaptively project the heterogeneous input spaces to the same lower-dimensional embedding space of concepts. Our experiments on six task-heterogeneous datasets demonstrate that HetMAML successfully leverages type-specific and globally shared meta-parameters for heterogeneous tasks and achieves fast within-task adaptions for each type of tasks.

CCS CONCEPTS
• Computing methodologies → Machine learning approaches; Multi-task learning; Supervised learning by classification.

KEYWORDS
few-shot learning, meta-learning, task heterogeneity, multimodality

1 INTRODUCTION
Humans can quickly learn new concepts from few examples by incorporating prior knowledge and context. Just like humans, few-shot learning (FSL) is the task of learning new concepts with a small number of labelled samples. Meta-learning is one prominent family of approaches to address FSL problems, which focuses on learning how to learn: how to generalize meta-knowledge from a distribution of tasks and utilize it to rapidly adapt to new tasks from the same task distribution. Recently, many gradient-based meta-learning methods have been proposed to handle FSL tasks [10, 18, 27, 38, 42, 46]. Most of these methods are model-agnostic meta-learners built upon MAML [10], aiming to find a single set of model parameters from similar tasks so that a small number of gradient updates will lead to good performance on future new tasks.

However, most existing gradient-based meta-learning methods assume the task distribution is homogeneous, that is, all tasks from either the same concept domain (e.g., a single dataset [10]) or a combination of different domains (e.g., multiple datasets [38]) have the same input feature space—for example, the inputs of all tasks are same-dimensional images. Such an assumption limits the generalization ability of these methods to handle more complex and heterogeneous task distribution (HTD): different tasks may vary in the structure of input data or the number of input modalities. For example, in few-shot emotion classification, some tasks may focus on recognizing people’s emotion from the video modality, while other tasks may predict on language scripts, recorded voices, or any combination of the three modalities. Figure 1(a) shows examples of task-homogeneous FSL, which assume identical data structures in all tasks. Figure 1(b) shows an example of task-heterogeneous FSL. Compared with the homogeneous FSL in Figure 1(a), heterogeneous tasks vary in terms of data type and the structure of input. Figure 2(a) illustrates another example of task-heterogeneous FSL containing four types of tasks, where we show each type of tasks is associated with a specific input space (see the grey areas in the figure). The input space of a type of tasks refers to the feature space shared by all the training/validation samples within these tasks.

In this paper, we propose a gradient-based meta-learning approach to FSL under heterogeneous task distributions. The novel task-heterogeneous few-shot learning problem is inevitable in many real-world low-data scenarios. Considering tasks can be either unimodal or multimodal with respect to their input data sources, there are mainly two cases of FSL across modalities. First, different modalities may convey the same semantics of a concept, and it is feasible to jointly learn multiple types of unimodal tasks if they share knowledge in the same concept domain. For instance, the image of an animal and the caption of it (text modality) both can distinguish this animal from others. Simultaneously learning image classification and text classification tasks that share the same concept domain could help us to build up the knowledge of animal concepts. Second, integrating multiple views of a concept or using auxiliary modalities usually achieves better performance than learning with a single modality [22, 41]. In this multimodal case, a problem neglected by existing literature is that due to the potential data missingness or data corruptness in some few-data scenarios, many tasks may not have a complete set where all modalities are always available [15]. Hence, it is uncertain that all these multimodal tasks have the same input structure, and we have to jointly learn multiple types of multimodal tasks, where different tasks may have different combinations of modalities. For example, in Figure 2(a) where there are four types of tasks, while type-2 tasks only have the modality-2 data.

Preprint. Accepted by CIKM21.
available, type-2 and type-3 tasks have the auxiliary modality-1 and modality-3, respectively. For both cases, it is beneficial to learn the heterogeneous tasks across modalities due to the following reasons: 1) jointly learning multiple views of tasks could help to build up the knowledge of the concept domain; 2) we can generalize reliable semantic feature extractors due to the implicit cross-modal alignment; and 3) the mechanism of training a unified meta-learner over heterogeneous tasks is more efficient than training separate models for each type of tasks as there are global meta-parameters that can be shared across different types of tasks.

With the heterogeneous task distribution (HTD), tasks share the same concept domain but have different input spaces (as shown in Figure 2(a)). As for meta-learning, while heterogeneous tasks could globally share some common knowledge (e.g., knowledge of the concept domain and early-stage feature embedding), there is also type-specific knowledge, locally shared by each type of tasks, which can reflect the different concept learning mechanisms of different types of tasks. Therefore, the key challenge of dealing with task heterogeneity is how to automatically reckon with type-specific information to promote fast within-task adaptions for each type of tasks. That is, we pursue how to customize the globally shared meta-learner for each type of tasks.

We address two issues in this challenge. First, the meta-learner for HTD should be able to balance generalization (i.e., the meta-knowledge globally shared across tasks) and customization (i.e., the type-specific meta-knowledge able to customize the global knowledge) based on each task’s specific input space. The difficulty is that, given a new task sampled from HTD whose input structure is unclear, the meta-learner should automatically leverage type-specific and global knowledge to achieve the task-aware projection: the mechanism that maps samples from the input space to the concept space should be different for different types of tasks, as illustrated in Figure 2(a) (arrow lines). Although one may convert the problem setting from heterogeneity into homogeneity using preprocessing strategies like imputation [2, 29, 36], this may introduce extra noise to the original task which has negative impacts on the performance, especially in low-data scenarios. Directly learning with different input spaces is necessary. Second, due to the inconsistency of input structures, tasks may use different model architectures and thus cannot share a single set of model parameters. Although one can train separate homogeneous meta-learners [10, 38] for each type of tasks, this strategy is not efficient as well as not effective: there are parameters shared by several types of tasks but trained repeatedly; the separate training will reduce the number of training tasks in each type which limits the ability of knowledge generalization; and, the knowledge shared between tasks is not well-explored.

To tackle the above issues, we propose Task-Heterogeneous Model-Agnostic Meta-Learning (HetMAML). The key idea of HetMAML is to leverage globally shared knowledge and type-specific knowledge to promote fast within-task adaptions for each type of tasks. We propose a three-module task-aware learning framework that can effectively capture both the global and type-specific meta-parameters and achieve the knowledge customization while simultaneously preserving knowledge generalization. Our main contributions are summarized as follows:

- We study a novel task-heterogeneous few-shot learning problem involving multiple input spaces, data types, and modality combinations. We show the importance of studying this real-world problem and define our setting of task heterogeneity. As far as we know, this is the first attempt to this problem.
- We introduce HetMAML, a model-agnostic meta-learning approach to task-heterogeneous few-shot learning. We propose a task-aware iterative feature aggregation network to promote the effectiveness of knowledge customization as well as adaptation efficiency.
- Our experimental results on six datasets demonstrate that HetMAML can effectively capture global and type-specific meta-parameters across heterogeneous tasks and fast adapt to different types of new tasks.

2 METHODOLOGY

2.1 Problem Formulation

In this section, we will provide an explicit formulation of the few-shot learning problem under heterogeneous task distribution.
Definition 1 (Heterogeneous Task Distribution). Suppose we have a total of $M$ data sources (modalities) and the data samples in each task are either unimodal or multimodal. A heterogeneous task distribution (HTD) $P(\mathcal{T})$ is defined as a mixture of $M'$ homogeneous task distributions $P(T^1), P(T^2), ..., P(T^{M'})$, where each $P(T^i)$ is associated with a specific input space, i.e., a certain combination of the $M$ modalities. Then there will be a total of $M'$ types of task instances sampled from the HTD, where $2 \leq M' \leq 2^M - 1$.

Task-heterogeneous Meta-learning. We are given a dataset $\mathcal{D}$ consisting of task instances sampled from a heterogeneous task distribution $P(\mathcal{T})$. The dataset $\mathcal{D}$ is divided into the meta-training $\mathcal{D}^{\text{train}}_{\text{meta}}$ and the meta-testing $\mathcal{D}^{\text{test}}_{\text{meta}}$ sets. Each meta-training task contains $N$ classes sampled from a set of classes $\mathcal{C}^{\text{train}}$, while the classes of each meta-testing task are sampled from a disjoint set of new classes $\mathcal{C}^{\text{test}}$. The goal of task-heterogeneous meta-learning is to find a set of meta-parameters from $\mathcal{D}^{\text{train}}_{\text{meta}}$ which can rapidly adapt to future novel tasks $\mathcal{D}^{\text{test}}_{\text{meta}}$.

Each task instance $T_i \sim P(\mathcal{T})$ is composed of a within-task training set $\mathcal{S}_{T_i}$ (support set) and testing set $\mathcal{Q}_{T_i}$ (query set), each of which consists of a few unimodal or multimodal data samples. The support set of a $N$-way $K$-shot classification task contains $K$ samples for each of $N$ classes $\mathcal{S}_{T_i} = \{(x^{(1)}_{i,nk}, y^{(1)}_{i,nk}) | n = 1...N, k = 1...K\}$, where $X_{i,nk} = (x^{(1)}_{i,nk}, x^{(2)}_{i,nk}, ..., x^{(M)}_{i,nk})$ denotes the composite input of each sample and $y^{(1)}_{i,nk}$ is the label. The query set contains $T$ samples from the same $N$ classes $\mathcal{Q}_{T_i} = \{(x^{*}_{i,nk}, y^{*}_{i,nk}) | n = 1...T\}$. Note that although we use the notation $X_i$ to include a full set of $M$ modalities for each sample, it still contains the information about which modalities are not available in this task. For example, if the $m$th source does not exist in task $T_i$, the source website or sensor will return a message indicating its unavailability, or the $m$th modality of each sample $X_i$ in a task is not informative. Then, given a task $T_i$, we are able to recognize the input space of this task by computing a configuration vector $c_{T_i} = [c_{T_i}^{(1)}, ..., c_{T_i}^{(M)}]^{\top}$ from the raw data in the support set $\mathcal{S}_{T_i}$ (described in Section 2.2.1).

2.2 Task-Heterogeneous Model-Agnostic Meta-Learning Framework

A traditional task neural network can be viewed as a two-module architecture, consisting of a backbone module (the earlier layers for feature extraction) and a head module (the later layers for decision making such as classification). Most existing model-agnostic meta-learners for homogeneous tasks are built upon such architecture. Due to the existence of task heterogeneity, we design a three-module framework, HetMAML, as illustrated in Figure 2(b), consisting of three modules: a multi-channel backbone, a single-channel head module, and an additional intermediate module—the task-aware feature aggregation network (TFAN), which can adaptively transform the composite information produced by the backbone into a single hidden representation fed to the head.

2.2.1 Multi-Channel Backbone Module. A challenge in this problem is that the input feature space of samples is type-specific; each type of tasks has a specific combination of modalities and hence the feature extraction schemes between different tasks can be different. For example, while image classification tasks use convolutional neural networks (CNNs) to extract features, other tasks that involve sequential data may use recurrent neural networks (RNNs). Despite the feature embedding being type-specific, each modality’s feature extractor can be shared by several types of tasks. In order to enable simultaneous learning over heterogeneous tasks, our backbone module should be able to effectively model both the shared and type-specific feature extractors. To achieve this goal, we propose a multi-channel backbone module, where each channel is a modality-specific feature extractor. Basically, if two tasks both have...
a modality in their input spaces, they can share the same feature extractor’s meta-parameters for embedding this modality.

Given a task $T$, it is easy to identify the input space of this task based on the data samples in the support set $S_T$. We can obtain a task-specific vector $c_T^\{m\} = [c_T^\{1\}, c_T^\{2\}, \ldots, c_T^\{T\}]^\top$ to represent the input space, where $c_T^\{m\} = 1$ denotes that the $m$th modality is available in this task ($c_T^\{m\} = 0$ denotes missing). Considering the missing modalities are not informative, $c_T^\{m\}$ is set to 0 if and only if the $m$th modalities of all the samples in $S_T$ are similar such that $\text{Var}(m) = \frac{\sum_{n,k} ||x_{nk}(m) - \bar{x}(m)||^2}{NK} < \epsilon$, where $\bar{x}(m) = \frac{\sum_{n,k} x_{nk}(m)}{NK}$ and $\epsilon$ is a threshold.

Then, suppose $\Theta_B = \{\theta_1, \theta_2, \ldots, \theta_M\}$ is the parameter set of the multi-channel backbone module, and $g_{\theta_m}(\cdot; \theta_m)$ is the feature extracting network for modality-$m$ with parameter $\theta_m$. Based on the pre-computed $c_T^\{m\}$ which indicates the input space of the task, for any sample $X = (x^{(1)}, x^{(2)}, \ldots, x^{(M)})$ in the task $T$, each modality can be embedded as $z(m) = g_{\theta_m}(x(m); \theta_m)$ if $c_T^\{m\} = 1$; or, $z(m) = 0$ if $c_T^\{m\} = 0$. Note that there will be no gradient update for those channels with $c_T^\{m\} = 0$.

The multi-channel backbone module will produce $M$ embedded vectors: $z^{(1)}, z^{(2)}, \ldots, z^{(M)}$. Each embedding vector $z^{(m)} \in \mathbb{R}^{F_1}$ represents a piece of modality-specific information. Note that despite the architecture differences between channels, all modalities are encoded onto the same $F_1$-dimensional semantic space, reducing the impact of input statistical divergence of different types of tasks.

### 2.2.2 Task-Aware Feature Aggregation for Heterogeneous Tasks

The set of encoded modality-specific information $\{z^{(m)}\}_{m=1}^M$ produced by the multi-channel backbone is not ready for decision making. One may consider simply concatenating all modality-specific vectors and directly feeding it to the head neural networks. However, this strategy is not effective due to the following two reasons. First, a concatenated vector is large and sparse—its dimension is $M \cdot F_1$ and many modalities that are absent in the task are represented by zero. This will result in a large feature mapping network at the beginning of the head module, which is not feasible in few-shot learning setting. Second, when there are multimodal types of tasks, the multiple input modalities can be highly-interacted [43], hence it is necessary to consider their potential interactions before decision making. These reasons motivate us to consider to effectively combine and condense all the modality-specific features. Thus, we insert an intermediate feature aggregation module $\Theta_A$ between the backbone module and head module, which can transform the set of modality-specific information produced by the backbone into a single lower-dimensional and non-sparse vector that represents the task-specific information.

In large-scale multimodal learning [16, 43], aggregating multiple features is a general topic. However, there are two challenges when multiple features are aggregated in a few-data scenario with an irregular input structure configuration. In other words, the multi-feature aggregation in a task-heterogeneous few-shot setting should satisfy two prerequisites: 1) **Efficiency of feature aggregation**. According to [15], multiple modalities contain complementary information, and effective aggregation should be able to combine them while leveraging their potential interactions. Although techniques such as Tensor Fusion Network (TFN) [43] have achieved effective interaction modeling, they basically require large-scale training data due to extensive parameters. We found that, in few-shot learning, such a large model tends to lead to over-fitting and requires more training episodes during adaption. Also, simple methods such as the concatenation are faster and smaller in model size but less powerful in exploring multimodal interactions. Correspondingly, in a heterogeneous few-data scenario, we should balance the adaptation efficiency and the exploration of multimodal interactions.

2) **Task awareness of feature aggregation**. Homogeneous few-shot learning focuses on learning the transferable knowledge generally shared across tasks (i.e., generalization). In contrast, in our heterogeneous few-shot setting, each task has a specific combination of modalities so that how to perform multimodal interactions can be very different between tasks. For example, while image classification tasks only analyze image data, a bimodal task containing both image and text should consider image-text interactions. The feature aggregation network shared by all types of tasks should be aware of each task’s specific input structure so that it can adaptively perform multimodal interactions. In other words, in addition to generalization, we also should learn how to customize the globally shared meta-learner (i.e., customization). That is, under a heterogeneous task distribution, the feature aggregation network should balance generalization and customization.

Motivated by the two prerequisites, we propose Task-aware Feature Aggregation Network (TFAN), which contains two components: 1) **Iterative Feature Aggregation** based on the bidirectional recurrent neural network and 2) **Attentional Contextual Learning**, to achieve the task-awareness of feature aggregation.

**Iterative Feature Aggregation Network**. As mentioned above, to achieve an effective feature aggregation in low-data scenarios, we pursue a balance between the exploration of multimodal interactions and the adaption speed. We found that the existing large-scale multimodal fusion technique [43, 44] as well as the straightforward feature combinations such as concatenation cannot effectively achieve this goal. Alternatively, we propose to employ the framework of bidirectional recurrent neural network (BRNN) [28] to iteratively combine the information of multiple modalities channel by channel while training parameters for processing only one channel at each step.

Suppose the set of multi-channel encoded modality-specific information $\{z^{(m)}\}_{m=1}^M$ constructs a sequence with $M$ components $\mathbf{Z} = (z^{(1)}, z^{(2)}, \ldots, z^{(M)})$, where we consider each piece of modality-specific information $z^{(m)}$ as a token as in text embedding. Using the bidirectional long-short term memory (BiLSTM) [12] network as an example of BRNN, in text embedding, BiLSTM learns a representation for each token by combining with the other words as context. Similarly, given the sequence $\mathbf{Z}$, each modality-specific embedding $z^{(m)}$ can be combined with the others through the bidirectional modality-by-modality iterative calculation process:

\begin{align}
\mathbf{h}^{(m)} &= \text{LSTM}(z^{(m)}, \mathbf{h}^{(m-1)}) \\
\mathbf{h}^{(m)} &= \text{LSTM}(z^{(m)}, \mathbf{h}^{(m+1)}) \\
\mathbf{h}^{(m)} &= \mathbf{h}^{(m)} \oplus \mathbf{h}^{(m)} \in \mathbb{R}^{F_2},
\end{align}
where $\odot$ denotes concatenation and $F_2$ is the dimension of hidden state. The iterative feature aggregation network will produce $M$ hidden states: $h = \{h^{(m)}\}_{m=1}^{M}$. Each hidden state $h^{(m)}$ is considered as a view of aggregated multimodal information combining all modality-specific information in $Z$.

The idea of iterative aggregation is to encode multimodal features modality by modality while exploring their interactions step by step. According to the forward pass function of an LSTM’s unit

$$
\begin{align*}
    & e_f^{m} = \text{sigmoid}(W_f z^{(m)} + U_f h^{(m-1)} + b_f) \\
    & e_i^{m} = \text{sigmoid}(W_i z^{(m)} + U_i h^{(m-1)} + b_i) \\
    & e_o^{m} = \text{sigmoid}(W_o z^{(m)} + U_o h^{(m-1)} + b_o) \\
    & c_t^{m} = \text{tanh}(W_c z^{(m)} + U_c h^{(m-1)} + b_c) \\
    & o_t^{m} = e_f^{m} \odot c^{m-1} + e_i^{m} \odot c_t^{m} \\
    & h^{(m)} = e_o^{m} \odot \text{tanh}(c_t^{m})
\end{align*}
$$

(2)

where $\{W_f, U_f, b_f\}$, $\{W_i, U_i, b_i\}$, $\{W_o, U_o, b_o\}$, and $\{W_c, U_c, b_c\}$ are the LSTM parameters of the forget gate, the input gate, the output gate, and the memory cell, respectively. At each iterative step, the memory cell’s update mechanism enables the feature interaction between each new coming feature $z^{(m)}$ and previous aggregated information $h^{(m-1)}$. Note that Eq. (2) is for each direction.

As for adaptivity efficiency, the number of trainable parameters of the iterative feature aggregation is $O((F_1 + F_2)F_2)$, where $F_1$ and $F_2$ are the dimensions of the embedding vector of each modality and the hidden state of the recurrent network, respectively. Compared with the highly-interacted feature aggregation methods designed for large-scale data, such as the multimodal outer-product containing $O((F_1)^{M} F_2)$ parameters [43], our model reduces the computation complexity of fusion and thus requires less training episodes and adapts faster. Also, compared with early fusion methods such as [16, 19, 25] containing $O(MF_1 F_2)$ parameters, iterative aggregation is more powerful in exploring multimodal interactions while the computation complexity is unrelated with $M$. In addition, as we also face the task-heterogeneity challenge, iterative aggregation works better in the task-heterogeneous few-shot learning environment.

**Attentional Contextual Learning for Task-aware Feature Aggregation.** Under a heterogeneous task distribution, we pursue the balance between generalization (globally shared knowledge) and customization (task-specific knowledge). In other words, the feature aggregation network should be aware of each task’s specific input structure, and learn how to explore task-specific knowledge to effectively customize the globally shared feature integration process. We propose the attentional contextual learning to achieve such balance, the task-awareness of feature aggregation.

First, since our definition of task heterogeneity is the configuration or availability of a set of input modalities, the sequence $Z = (z^{(1)}, z^{(2)}, \ldots, z^{(M)})$ can reflect the specific knowledge for the given task’s input structure–each modality has its own position in this sequence and the absent modalities can be considered as a special token. In other words, we can view the input structure of tasks as the “context” of the sequence $Z$, which can distinguish different types of tasks. In text embedding, BRNN models the sequential data by leveraging context dependencies in the input sequence (context-aware) [45]. Similarly, when the BRNN-based iterative feature aggregation encodes the sequence of modalities, it can take into account the input structure (context) of the given task, which reflects task-specific knowledge.

Besides, to further customize the iterative feature aggregation network using task-specific knowledge, we propose an external meta-network $g_{\phi}(\cdot; \phi)$, which learns a task embedding vector $r$ for each given task to represent the task-specific knowledge as

$$
\tau = g_{\phi}(c_T; r) \in \mathbb{R}^{F_1},
$$

(3)

where $F_3$ is the dimension of task embedding. We let the task embedding $r$ represent the type-specific knowledge that reflects each task’s specific input space and distinguishes different tasks; hence, $r$ is embedded from the $c_T$ calculated in Section 2.2.1. Then, we utilize $r$ as extra knowledge to customize the globally shared meta-parameters in the feature aggregation network. Specifically, we employ a modified attention mechanism [1] to facilitate the customization. The final task-specific information is represented by

$$
\hat{h} = \sum_{m=1}^{M} A_m h^{(m)} \in \mathbb{R}^{F_2},
$$

(4)

where $\{A_m\}_{m=1}^{M}$ are attention coefficients indicating the role of each modality with respect to the type-specific knowledge $r$:

$$
A_m = \frac{\exp(v^T \text{tanh}(W_k[h^{(m)} \odot r]))}{\sum_{l=1}^{F_2} \exp(v^T \text{tanh}(W_k[h^{(l)} \odot r]))},
$$

(5)

where $\odot$ denotes concatenation operation, and $v \in \mathbb{R}^{F_1}$ and $W_k \in \mathbb{R}^{F_1 \times (F_1 + F_3)}$ are learnable parameters of the attention module. The attention mechanism incorporates the type-specific knowledge $r$ into the feature aggregation, so that it can adaptively perform the within-task adaption for different types of tasks.

Overall, our task-aware feature aggregation network (TFAN) transforms the multi-channel modality-specific information into task-specific information. Network parameters in this module include the iterative feature aggregation $\theta_{iter}$, the task-embedding $\phi$, and the attention module $(v, W_k)$.

**2.2.3 Single-Channel Head Module.** In the previous stage, TFAN adaptively projects samples from different input spaces onto a unified concept space—the generated task-specific representations of each sample in different type of tasks are in the same concept space (see arrow lines in Figure 2(a)). As for final decision making, we use a single-channel head module $f_{\theta_{h}}(\cdot; \theta_{H})$. For few-shot classification, the architecture of the head module is a multi-layer classifier followed by a softmax layer, i.e., $\text{softmax}(f_{\theta_{h}}(h^*; \theta_{H}))$.

**2.3 Training Procedure**

HetMAML is built upon the model-agnostic meta-learning (MAML) framework [10, 26], which solves a bilevel optimization problem to find an initialization $\theta_0$ for a neural network as the meta-parameters. This training procedure assumes that the single meta-initialization $\theta_0$ is an appropriate generalization of prior knowledge for all tasks. It requires tasks to be homogeneous in terms of the size of labelled data, the classes number, the input structure, and so on.

Unlike homogeneous MAMLs, our task-heterogeneous model leverages extra task-specific knowledge to perform slightly different adaption between different types of tasks, that is, finding a balance between generalization and customization. Therefore, we divide
2.3.1 Internal and External Parameters. We define internal parameters (IP) \( \Theta \) as the meta-parameters, which take the gradient updates during the inner-loop optimization, to perform the adaption to each specific task. The internal parameters of HetMAML include the meta-initialization of the iterative feature aggregation network and the head module, i.e., \( \Theta = \{ \theta_{\text{iter}}, \theta_{\text{H}} \} \). The task-aware iterative feature aggregation projects input features onto the concept space, by considering the input context of each task. Hence \( \theta_{\text{iter}} \) may vary among different tasks and need to adapt to each task. The head module (classifier) \( \theta_{\text{H}} \) learns decision boundaries within the concept space. Since different tasks have different concepts, the parameters for dividing the concept space is specific for each task.

External parameters (EP) \( \Phi \) are defined as the meta-parameters, which pass through the neural network but does not need to adapt to each task during inner-loop optimization. The external parameters of HetMAML include the parameters of multi-channel backbone module, the task embedding network as well as the attention mechanism used for customization, i.e., \( \Phi = \{ \theta_{\text{B}}, \phi, v, W_b \} \). The backbone \( \theta_{\text{B}} \) can be shared across tasks as it performs early-stage feature embedding of each data source. Meta-parameters in the task embedding network \( \phi \) aim to capture the knowledge of how to distinguish different types of tasks, based on the structure property of each given task. \( \phi, v, \) and \( W_b \) embeds each task’s input structure and utilize it to customize the internal parameters during adaptation.

2.3.2 Bilevel Optimization. The overall training procedure of HetMAML is described in Algorithm 1. Following [10, 26], the outer loop updates the meta-initialization of internal parameters \( \Theta_0 \) and the external parameters \( \Phi \) to enable fast adaptation over a batch of task instances. The inner loop takes the outer-loop \( \Theta_0 \) and \( \Phi \), and, separately for each task, performs a few gradient updates of internal parameters over the labelled examples in the support set, while freezing external parameters.

Formally, let \( \Theta'_i \) signify \( \Theta \) for task \( T_i \) during the inner-loop optimization, and let the initial \( \Theta'_0 = \Theta_0 \). In the inner-loop adaption, during each gradient update, we compute

\[
\Theta'_i \leftarrow \Theta'_i - \alpha \nabla_{\Theta'_i} L_{T_i}(f(X; \Theta'_i, \Phi), y; S_{T_i}),
\]

where \( f(\cdot) \) is the forward function of HetMAML network, and \( L_{T_i}(\cdot; S_{T_i}) \) is the loss on the support set of task \( T_i \).

Separately for each task, after a fixed number of inner-loop updates, we obtain the adapted parameter \( \Theta'_i(\Theta_0) \), which is dependent on meta-initialization \( \Theta_0 \). Then, the outer-loop optimization updates \( \Theta_0 \) and \( \Phi \) over a batch of task instances:

\[
\Theta_0 \leftarrow \Theta_0 - \beta \nabla_{\Theta_0} \sum_{T_i \sim p(T)} L_{T_i}(f(X'; \Theta'_i(\Theta_0), \Phi), y'; Q_{T_i}) \tag{7}
\]

\[
\Phi \leftarrow \Phi - \beta \nabla_{\Phi} \sum_{T_i \sim p(T)} L_{T_i}(f(X'; \Theta'_i(\Theta_0), \Phi), y'; Q_{T_i}) \tag{8}
\]

where \( L_{T_i}(\cdot; Q_{T_i}) \) is the loss on the query set of task \( T_i \).

3 EXPERIMENTS

In this section, we compare HetMAML with state-of-the-art baselines on six task-heterogeneous few-shot datasets.

Algorithm 1 Training Procedure of HetMAML

1. Requires: Heterogeneous task distribution \( p(T) \)
2. Requires: Learning rates \( \alpha, \beta \)
3. Randomly initialize internal parameters \( \Theta_0 \)
4. Randomly initialize external parameters \( \Phi \)
5. // outer-loop optimization
6. while not done do
7. Sample batches of heterogeneous tasks \( T_i \sim p(T) \)
8. // inner-loop optimization
9. for all \( i \) do
10. Obtain data \( \{ S_{T_i}, Q_{T_i} \} \) for each task \( T_i \)
11. Obtain task structure \( c_{\text{trn}} \) and \( c_{\text{test}} \) from each \( S_{T_i} \) as Section 2.2.1
12. Obtain task embedding \( r_i = g(c_{\text{trn}}; \phi) \) using external \( \phi \)
13. Compute adapted internal parameters with a fixed number of steps \( w.r.t. \) the NK examples from \( S_{T_i} \) as in Eq.(6)
14. Evaluate \( L_i(f(X'; \Theta'_i, \Phi), y'; Q_{T_i}) \) w.r.t. \( T \) samples of \( Q_{T_i} \)
15. end for
16. Update initialization of internal parameters \( \Theta_0 \) as Eq.(7)
17. Update external parameters \( \Phi \) as Eq.(8)
18. end while
19. return: \( \Theta_0 \) and \( \phi \)

3.1 Datasets

Since we define a new task-heterogeneous few-shot learning problem, we constructed our datasets from existing multimodal datasets.

3.1.1 Selection of the Source Multimodal Datasets. One prerequisite for the source multimodal dataset is that the total number of classes \( |C| \) should be large. First, we have to split \( C \) into two disjoint sets of class set, \( C_{\text{trn}} \) and \( C_{\text{test}} \), for constructing \( D_{\text{trn}} \) and \( D_{\text{test}} \), respectively. Then, the class set \( C_i \) of each task instance is sampled from \( C \). To construct a meta-dataset containing a proper number of task instances (for training meta-learner), where the \( N \) classes in different task instances do not overlap too much, a large number of total classes is necessary. Another requirement is that, in the source multimodal dataset, the number of subjects (multimodal samples) for each class should also be large to make sure each data sample will not repeat frequently over tasks. Since some multimodal datasets with three or more modalities such as emotion classification datasets have a small number of classes and subjects, we did not consider using them as the source.

We chose the following two datasets as our source multimodal datasets: 1) Caltech-UCSD-Birds 200-2011 (CUB-200) [39] dataset contains 11,788 images for 200 bird species, including black footed albatross, bobolink, fish crow, and so on. Each image is annotated with a vocabulary of 312 attributes (e.g., crown color, belly color, eye color, etc.), which we use as the text modality. We used the two data sources from the CUB-200 dataset: the image modality \( x^{\text{image}} \in \mathbb{R}^{3 \times 256 \times 256} \), and the text modality \( x^{\text{text}} \in \mathbb{R}^{312} \). 2) ModelNet40 [40] is a 3D object detection dataset containing 12,311 3D CAD shapes covering 40 common categories, including airplane, bathtub, bookshelf, bottle, bowl, cone, cup, and so on. Following [8], we use it as a bimodal dataset where the two modalities are the two views of shape representation extracted from two pre-trained networks: Multi-view Convolutional Neural Network
TABLE 1: Statistics of the six task-heterogeneous few-shot datasets (X1: MVCNN modality, X2: GVCNN modality).
Table 2: Few-shot classification accuracy (%) on the meta-test splits of hetModelNet-1 (2, 3, and 4) datasets.

| Method                  | hetModelNet-1 |       |       | hetModelNet-2 |       |       | hetModelNet-3 |       |       | hetModelNet-4 |       |
|-------------------------|---------------|-------|-------|---------------|-------|-------|---------------|-------|-------|---------------|-------|
|                         | 5-way         | 1-shot|       | 5-way         | 1-shot|       | 5-way         | 1-shot|       | 5-way         | 1-shot|
| MAML                    | 80.6          | 91.2  | 60.5  | 80.4          | 90.6  | 46.5  | 89.6          | 98.7  | 70.1  | 84.2          | 94.7  |
| Multi-MAML(TF)          | 75.7          | 94.2  | 50.5  | 68.2          | 88.6  | 32.3  | 77.4          | 93.3  | 54.3  | 74.3          | 91.4  |
| Multi-MAML(BF)          | 76.1          | 94.3  | 50.7  | 80.8          | 94.3  | 62.5  | 92.7          | 98.9  | 81.3  | 83.9          | 94.5  |
| HetMAML (ours)          | 84.5          | 94.7  | 72.8  | 85.7          | 95.2  | 71.9  | 95.0          | 99.3  | 90.1  | 92.3          | 95.3  |

Table 3: Few-shot classification accuracy (%) on the meta-test splits of hetCUB200-1 and hetCUB200-2 dataset.

| Method                  | hetCUB200-1 |       |       | hetCUB200-2 |       |       |
|-------------------------|-------------|-------|-------|-------------|-------|-------|
|                         | 5-way       | 1-shot|       | 5-way       | 1-shot|       |
| MAML                    | 43.9        | 60.2  | 52.7  | 69.5        |       |       |
| Multi-MAML(TF)          | 41.3        | 63.1  | 39.1  | 51.2        |       |       |
| Multi-MAML(BF)          | 41.4        | 62.8  | 50.3  | 68.4        |       |       |
| HetMAML (ours)          | 48.7        | 64.3  | 54.2  | 70.7        |       |       |

Table 4: Comparison of model size and the average meta-training time (MT-Time) on hetModelNet-4.

| Method                  | MT-Time   | Model Size |
|-------------------------|-----------|------------|
| Multi-MAML (TF)         | 26.2 min  | 2,762,905  |
| Multi-MAML (BF)         | 34.5 min  | 1,736,783  |
| HetMAML (ours)          | 17.5 min  | 908,485    |

3.3.3 Efficiency and Scalability. Table 4 reports the comparison on the model sizes and training time between HetMAML and Multi-MAML. Compared with separate training baselines, our HetMAML uses one-third or a half of trainable meta-parameters to achieve better adaptation performance, showing that HetMAML can efficiently and effectively generalize both type-specific and shared meta-parameters across heterogeneous tasks. Moreover, HetMAML can be more scalable than baselines in terms of the number of data sources, in spite that our datasets include only two modalities because existing multimodal datasets having more modalities do not have enough samples to construct meta datasets. For HetMAML, the increase of \( \Delta M \) modalities will result in \( \Delta M \) feature extractors \( \{ \theta_m | \forall m \in \Delta M \} \) added to the multi-channel backbone, while the other two modules’ parameters will remain the same.

4 RELATED WORKS

In this section, we briefly review related works in meta-learning, cross-modal domain adaption, multimodal multi-task learning, and multimodal integration.

Meta-Learning. Meta-learning approaches to few-shot learning problem mainly include gradient-based methods [10, 18, 42, 46] and metric-based methods [14, 20, 30, 34, 37]. While metric-based meta-learning learns a distance-based prediction rule over the embeddings, gradient-based methods employ a bilevel learning framework that adapts the embedding model parameters given a task’s training examples. We follows gradient-based paradigm as the bilevel framework than Multi-MAML baselines. The adaption speed of the TFN-based baseline on type-(X1,X2) tasks is very slow (see the blue dash-dotted lines), suggesting that the large-scale multimodal fusion strategy is not suitable to few-shot multimodal tasks. According to the left-2 and left-3 charts, Type-(X1,X2) adaption in Multi-MAML (BF) is faster than the type-X1 or type-X2 adaption, proving the efficiency of iterative aggregation on few-shot multimodal fusion.

In addition, HetMAML outperforms Multi-MAML (BF), especially in 5/10-way 1-shot settings. It is because HetMAML simultaneously trains all types of tasks and thus can access more data from other types of tasks during training. In contrast, each MAML in Multi-MAMLs only uses the smaller number of tasks of a specific type for training these networks. Jointly training heterogeneous tasks could capture more reliable meta-knowledge rather than separately training multiple MAMLs.

3.3.2 Adaption Speed Comparison. We also compare adaption speed in Figure 3, where we display the 10-way 1-shot classification accuracy (meta-test) with respect to the number of gradient updates, separately for each type of tasks on each of the four hetModelNet datasets. In this figure, we compare HetMAML with Multi-MAML baselines on two datasets. Other baselines are not compared here because they view different types of tasks as the same type. Solid lines denote the average results of all types of tasks; dashed lines denote the results of tasks that only have modality X1 as the input (type-X1); dotted lines denote the results of tasks that only have modality X2 as the input (type-X2); and dash-dotted lines denote tasks that have a pair of modalities as the input (type-(X1,X2)).

Overall, while the baselines need 8-9 or more gradient updates to adapt to new tasks, HetMAML can adapt to new tasks with just 2-3 updates, which demonstrates the fast adaption ability of HetMAML. We can observe that although HetMAML trains all types of tasks simultaneously, it can successfully fast adapt to each type of new tasks, suggesting that the proposed model is aware of the task-specific input structure. In addition, HetMAML outperforms baselines, especially on unimodal tasks, because the training of each types of tasks can benefit from more data from other types. Furthermore, these results are achieved with less training time and parameter size, showing that HetMAML is a more efficient.
framework has the potential mechanisms for feature selection and could be more robust to input variations rather than metric-based frameworks. Recent FSL [7, 21, 22, 41] extended to multimodal few-shot scenarios. Yet most of these methods assume homogeneous multimodal features, where input samples over tasks consist of the same set of modalities. In contrast, we focus on the case that some modalities may not be available in some tasks and thus different tasks may have different combinations of modalities. Recently, task-adaptive meta-learning [13, 27, 35, 38] considered task distribution across domains, but different tasks still share the same input space. Different from these methods that assume task homogeneity, we study a novel FSL approach under heterogeneous task distributions.

**Cross-modal Domain Adaption.** Domain Adaptation (DA) is useful in the situations where the source and target domains have the same classes but the input distribution of the target task is shifted with respect to the source task. Cross-modal DA deals with the special case that the inputs of the source and target differ in modalities [24], which is similar to the cross-modal case of our task heterogeneity. However, cross-modal DA focuses on the adaption between two domains (source and target). In contrast, this paper learns meta-knowledge for unlimited unseen tasks. There is no meta-objective in DA that optimizes "how to learn" across tasks.

**Multimodal Multi-Task Learning.** Multi-Task Learning (MTL) aims to jointly learn several related tasks such that each task can benefit from parameters or representation sharing across different tasks. Multimodal MTL [5, 23] deals with using MTL methods to solve multimodal integration problem. Our HetMAML also attempts to design a parameter-sharing strategy so that all types of tasks can be learned with a unified framework. However, we aim to obtain meta-knowledge across unlimited tasks leveraging these shared parameters, whereas MTL intends to solve a fixed number of known tasks through a single-level optimization without a meta-objective.

**Multimodal Integration.** Deep multimodal integration studies how to align or integrate different modalities (e.g., visual, language, acoustic, or others) into a joint embedding. Existing methods include early fusion [11, 19, 25, 31], late fusion that models the dynamics of multimodal interactions [4, 17, 43], and multimodal sequential learning [3, 32, 44]. A majority of existing multimodal fusion models rely on large-scale training data so that their performance may drop dramatically in the few-data scenario. Also, most works build frameworks designed for single-type tasks, which are not very robust to heterogeneous few-shot tasks. Although one can impute unavailable modalities using imputation strategies [2, 6, 29, 36] to convert the problem into homogeneity, this may introduce extra noise to the original few-data task, and also, the multimodal integration function may still vary between different types of tasks. In contrast, HetMAML directly learns with different input spaces and can be aware of the input structures of multimodal tasks to promote knowledge customization.

## 5 CONCLUSIONS

In this paper, we introduced HetMAML, a novel task-heterogeneous meta-agnostic meta-learning approach to few-shot learning under heterogeneous task distribution. To effectively capture both the type-specific and globally shared meta-parameters, we designed a three-module framework, including a multi-channel backbone to extract modality-specific embeddings, a task-aware feature aggregation network to adaptively transform these multimodal embeddings into a task-specific representation by leveraging the task-specific knowledge, and a single-channel head module for decision making. Experiment results show that HetMAML is able to effectively and efficiently capture meta-parameters across heterogeneous tasks, balance customization and generalization, and successfully fast adapt to all types of new tasks.
REFERENCES

[1] Dmytro Badanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Yoshua Bengio and Yann LeCun (Eds.) http://arxiv.org/abs/1409.0471

[2] Lei Cai, Zhengyang Wang, Hongyang Gao, Dinggang Shen, and Shuiwang Ji. 2019. Deep-adversarial learning for multi-modality missing data completion. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM.

[3] Stanislas Chambon, Matthieu N Galtier, Pierrick J Arnal, Gilles Wainrib, and Stanislas Chambon. 2020. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001. California Institute of Technology.

[4] Jaoui Chen and Aishong Zhang. 2020. HGMF: Heterogeneous Graph-based Fusion Model for multimodal Data with Incompleteness. In Proceedings of the 26th ACM International Conference on Multimedia. 108–116.

[5] Ryan Elff, Herman A Englebrecht, and Herman Karper. 2019. Multimodal one-shot learning of speech and images. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 8623–8627.

[6] Yafan Feng, Xiaoxiao Zhu, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. 3558–3565.

[7] Yafan Feng, Xizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. 2018. GCCNN: Group-view convolutional neural networks for 3D shape recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 264–272.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 1126–1135.

[9] Hatice Gunes and Massimo Piccardi. 2005. Affect recognition from face and body: An Overview of Methods, Challenges, and Prospects. IEEE Transactions on Automatic Control. 103, 9 (2015), 1233–1252.

[10] Dongni Han, Tong Zhao, Shuo Zhang, and Bingbing Ni. 2019. Few-shot learning of speech and images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 945–953.

[11] Cebeda Huang, Cherng-Hwa Kuo, and Peng-Chih Hsu. 2018. VIGAN: Missing view imputation with generative adversarial networks. In Proceedings of the 13th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 7693–7702.

[12] Qiuling Suo, Jingyuan Chou, Wei Dong, and Louis-Philippe Morency. 2020. Tadanet: Task-adaptive dual-task network for graph-enriched meta-learning. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1789–1799.

[13] Luan Tran, Xiaoming Liu, Jayyu Zhou, and Rong Jin. 2017. Missing modalities imputation via cascaded residual autoencoder. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1405–1414.

[14] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Randall Davis. 2012. Multimodal human behavior analysis: learning correlation and interaction across modalities. In Proceedings of the 14th ACM international conference on Multimodal interaction. 719–726.

[15] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Ji. 2017. VEGAN: Missing view imputation with generative adversarial networks. In Big Data (Big Data), 2017 IEEE International Conference on. IEEE.

[16] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. 2018. Tadam: Task-adaptive Meta-Learner. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net. https://openreview.net/forum?id=BiDmUWAW

[17] Louie-Philippe Morency, Rada Mihalcea, and Payal Doshi. 2011. Towards multi-modal sentiment analysis: Harvesting opinions from the web. In Proceedings of the 13th international conference on multimodal interfaces. 169–176.

[18] Fredrik Dahle, Oleksiy Ostanenko, Patrick Ja Hnichen, Tassilo Klein, and Moin Nabi. 2019. Self-paced adversarial training for multimodal few-shot learning. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 218–226.

[19] Fredrik Dahle, Mihai Puscas, Tassilo Klein, and Moin Nabi. 2020. Multimodal Prototypical Networks for Few-shot learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2644–2653.

[20] Swinivas Parathasarathy and Carlos Busso. 2017. Jointly Predicting Arousal, Valence and Dominance with Multi-Task Learning. In Interspeech 1103–1107.

[21] Costa Pereira and Nuno Vasconcelos. 2014. Cross-modal domain adaptation for text-based regularization of image semantics in image retrieval systems. Computer Vision and Image Understanding 114 (2014), 123–135.

[22] Demirvuran, B., and H. A. Imre. 2013. Utterance-level multimodal sentiment analysis. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 973–982.

[23] Andrea A. Rusu, Dushyant Rao, Jakub Szymkowiak, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Rasa Hadidzic. 2018. Meta-Learning with Latent Embedding Optimization. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net. https://openreview.net.

[24] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. IEEE transactions on Signal Processing 45, 11 (1997), 2673–2681.

[25] Stanislas Chambon, Mathieu N Galtier, Pierrick J Arnal, Gilles Wainrib, and Stanislas Chambon. 2020. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001. California Institute of Technology.

[26] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Randall Davis. 2012. Multimodal human behavior analysis: learning correlation and interaction across modalities. In Proceedings of the 14th ACM international conference on Multimodal interaction. 719–726.

[27] Xiaohui Chen, Changping Du, Changying Du, Hao Wang, Jinpeng Li, Wei-Long Zheng, Bao-Liang Lu, and Huaguang He. 2018. Semi-supervised deep generative modelling of incomplete multi-modality emotional data. In Proceedings of the 26th ACM International Conference on Multimedia. 108–116.

[28] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Ji. 2017. VEGAN: Missing view imputation with generative adversarial networks. In Big Data (Big Data), 2017 IEEE International Conference on. IEEE.

[29] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. 2005. Early versus late fusion in semantic video analysis. In Proceedings of the 13th ACM international conference on Multimedia. 399–402.

[30] Jaesik Yoon, Louis-Philippe Morency, and Randall Davis. 2012. Multimodal human behavior analysis: learning correlation and interaction across modalities. In Proceedings of the 14th ACM international conference on Multimodal interaction. 719–726.

[31] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Li. 2020. Tadanet: Task-adaptive dual-task network for graph-enriched meta-learning. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1789–1799.

[32] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. 2005. Early versus late fusion in semantic video analysis. In Proceedings of the 13th ACM international conference on Multimedia. 399–402.

[33] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE international conference on computer vision. 945–953.

[34] Zadeh, Amir Ali Bagher Zadeh, and Louis-Philippe Morency. 2018. Efficient Low-rank Multimodal Fusion With Modality-Specific Factors. In Proceedings of the 6th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2247–2256.

[35] Nikhil Doshi, Ian Fischer, and Ousmane Dia. 2021. Few-shot learning of speech and images. In Proceedings of the 35th AAAI Conference on Artificial Intelligence. 103–110.