HyperFuse: Multimodal Fusion via Hypernetworks

Anonymous Authors[1]

Abstract

Modern research in deep multimodal fusion has explored a variety of architectural designs that integrate multiple modalities into optimal fused representations. These research directions have mostly focused on embedding-based fusion layers. Our work explores an alternative approach in which a hypernetwork uses auxiliary modalities to update weights for the primary modality in a base model. Our method, HyperFuse, is empirically shown to generate more informative representations than other common fusion techniques while being model-agnostic. We evaluate the method on four different domains and a variety of base architectures, and found HyperFuse provides a consistent performance boost ranging from 0.4%-1.5% accuracy point improvement compared to state-of-the-art classification models.

1. Introduction

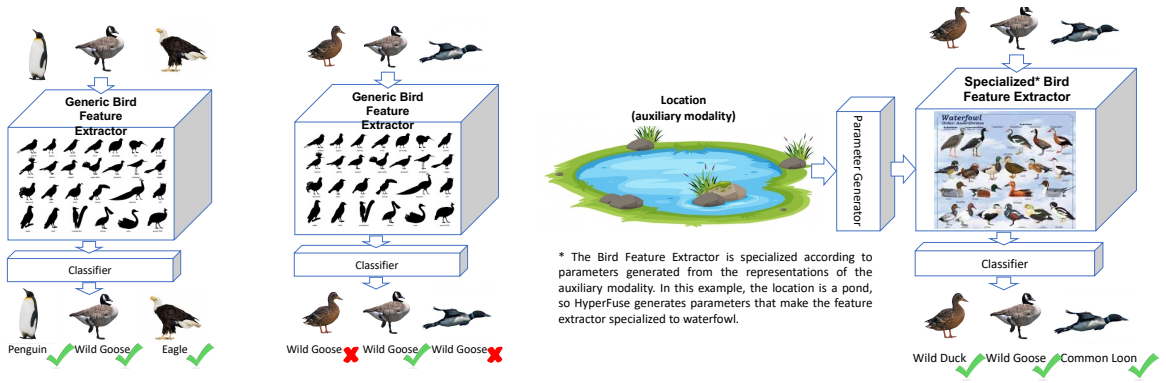
Humans perceive the world through multiple sensory modalities and integrate unimodal information to form a smooth, coherent representation of their surroundings. Research in deep multimodal fusion aims to achieve similar unified, cohesive representations from multiple unimodal sources(Baltrušaitis et al., 2018). Such research has led to significant improvements in tasks like opinion analysis (Garcia et al., 2019; Soleymani et al., 2017), language (Garbacea and Mei, 2020; Storks et al., 2019), image processing (Xu et al., 2015) and visual reasoning (Liu et al., 2020).

Traditional fusion approaches have generally focused on learning non-adaptive models for producing fused representations. However, a fixed representation function may not suffice for tasks with a large number of outputs, like those based on fine-grained classification (Li et al., 2019; Mac Aodha et al., 2019) and language modeling (Chang and McCallum, 2022). Consider a problem of fine-grained species classification with two modalities, habitat and images. While using habitat information in addition to visual features in general can improve the classification accuracy (Mac Aodha et al., 2019), simple fusion methods that combine habitat modality and visual features into a fixed fused representation do not suffice to distinguish between environmentally related and visually similar species (e.g. ducks and geese). To address this challenge, we need to incorporate an interactive relationships between the two modalities. For example, to distinguish ducks from penguins, a representation based on shape would suffice. However, to distinguish similar species likes ducks from geese, the generic shape based representation would not be enough (Figure 1a). Furthermore habitat information also would not provide enough additional information as ducks and geese often share habitat. Ideally, knowing an aquatic habitat, we want the model to learn a different representation that highlights specific image features pertinent for distinguishing aquatic birds (Figure 1b) like bill length and feather patterns . This illustrates the need for the network to map similar inputs to different representations based on auxiliary modality.

Standard fusion techniques struggle to learn such conditionally fused representations efficiently (Rath and Condurache, 2022) without using high-dimensional tensor products (Zadeh et al., 2017). To address this, we propose a filter generation mechanism in the parameter space by learning filter parameters dependent on the auxiliary information. In this way, the model can easily adapt its representation space based on the auxiliary information. Hypernetworks (Ha et al., 2016; Stanley et al., 2009), which are networks that generate parameters for another network, provide a natural way to induce such conditioning between modalities.

To this end, we propose HyperFuse, a new fusion method based on hypernetworks (Ha et al., 2016) to expand the representational capacity of the model using a generator for adaptive filter parameters. HyperFuse allows us to involve the higher-dimensional interaction between the multimodal representations without creating high dimensional tensor product features. The weights of HyperFuse are generated from the multimodal features extracted from the additional information. The hypernet structure takes a selection of auxiliary modalities and produces parameters for the primary modality network.

We test our proposal on a variety of multimodal datasets like AV-MNIST, iNat, MOSI, and MOSEI, among others. Our results show that HyperFuse-based models consistently match or outperform state-of-the-art models such as MAG-BERT(Rahman et al., 2020), EnsembleNet (Terry et al., 2020), and MFAS (Pérez-Rúa et al., 2019).



(a) (b)

Figure 1: a) Generic shape based representation can distinguish between dissimilar species like geese and penguins. However, on similar species like geese and ducks, the same representations are less accurate. b) Conditioning on the information that the habitat is aquatic, the model can produce representations more pertinent for distinguishing waterfowl and aquatic birds.

2. Preliminaries

2.1. HyperNetworks

Hypernetworks are network designs where the weights of the primary neural network are generated by another auxiliary network (Stanley et al., 2009). Commonly, the primary network is larger than the auxiliary one. Instead of learning the parameters θ_{target} of a particular function $\mathcal{F}_{\text{target}}$ directly, one learns the parameters of a primary model.

While weight-generating networks have been known for a while (Schmidhuber, 1992; Stanley et al., 2009; Bertinetto et al., 2016), they were re-introduced under this name by Ha et al. (2016) who explored their applications for CNN and RNNs. Further research studies have shown the usefulness of hypernetworks in applications around multi-task learning (Von Oswald et al., 2019; Huang et al., 2021; Ehret et al., 2020), meta-learning (Zhao et al., 2020) and model compression (Li et al., 2020). Hypernetworks are also quite effective for a variety of tasks ranging from 3D reconstruction (Littwin and Wolf, 2019), architecture search (Brock et al., 2018), exploration (Dwaracherla et al., 2020) and bioinformatics (Nachmani and Wolf, 2020).

2.2. Multimodal Fusion

Consider a dataset of N observations $\mathcal{D} = (x_i, y_i)_{i=1}^N$, in which $x_i \in \mathbb{X}$ and $y_i \in \mathbb{Y}$. In multimodal fusion, the space of inputs $\mathbb{X}$ decomposes into K different modalities $\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2 \times ... \times \mathbb{X}_K$. Provided with a loss function $L : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}$, we aim to learn a model $\mathcal{M}_\theta : \mathbb{X} \to \mathbb{Y}$ that minimizes the total loss $\mathcal{L} = \sum_i L(\mathcal{M}_\theta(x_i), y_i)$.

A common way to learn such a multimodal model is to decompose it into two components: 1) an embedding function $E : \mathbb{X} \to \mathbb{R}^d$ which transforms raw information into a d-dimension vector space, and 2) a task-specific outcome component $O : \mathbb{R}^d \to \mathbb{Y}$. Since different modalities are often not directly compatible with each other (e.g. text and image), the function E can be decomposed into a) modality-specific embedding functions $E_i : \mathbb{X}_i \to \mathbb{R}^{d_i}$, and b) a fusion function $F : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times ... \times \mathbb{R}^{d_K} \to \mathbb{R}^d$ which combines information from each unimodal embedding to create a fused d-dimension embedding of all K modalities.

3. Related Work

Tensor Fusion These methods seek to utilize higher-order multiplicative interactions for capturing complementary relationships between modalities. However, due to the high computational cost of such tensor products, several approximations have been proposed that trade-off between efficiency and flexibility (Hou et al., 2019; Liu et al., 2018). For example, LFN (Zadeh et al., 2017) combined information via pooling projections of high dimensional tensor representation of multimodal features. Multiplicative models that generalize tensor products to include learnable parameters have also been proposed (Jayakumar et al., 2020) to capture multimodal interactions. A variety of other models (Perez et al., 2018; Zhong et al., 2020) can be interpreted as restrictions of these general multiplicative models.

Multimodal gated units Gating mechanism learns representations that dynamically change for every input (Chaplot et al., 2018; Wang et al., 2019). A general gated unit can be written as a scaled version of the input vectors like $\mathbf{z}_p \odot h(\mathbf{z}_a)$, where h represents a scaling function such as sigmoid activation which amplifies and suppresses different components of $\mathbf{z}_p$. The MAGBERT/MAGXLNET models (Rahman et al., 2020) used gating to adjust the input embeddings to a language model and achieve SOTA results

on sentiment classification. Recently, Xue and Marculescu (2022) explored gating-based methods to save computation and induce data-dependent computation. Our HyperFuse approach learns to predict dynamic parameters for the fusion layer and can support instance-specific computation.

Dynamic Models Our method is directly related to ShortFuse (Fiterau et al., 2017), which improved biomedical time series models by leveraging the structured covariates in a similar fashion. Convolution layers which learn dynamic parameters for multiple kernels have been successfully used in a number of vision problems such as object detection (Sun et al., 2020; Wang et al., 2020b) and segmentation (Tian et al., 2020).Jia et al. (2016a) dynamically generates the filters conditioned on the input images. Other similar models including CondConv (Yang et al., 2019), DyNet (Zhang et al., 2020), and Dynamic Conv (Chen et al., 2020) learn dynamic parameters for multiple kernels. However, these are mostly unimodal models and derive adaptive filters from a single spatial feature map (Prakash et al., 2021; Yang et al., 2022a). Instead, ours uses a primary network whose parameters are generated by an auxiliary network conditioned on other modalities. Models like Hu et al. (2018) and Gondal et al. (2021) have also been used for dynamic inference, but these are not designed for multimodal setting and focus on multi-task learning.

Fusion Architectures Due to the wide variety of applications and tasks which require multimodal fusion, over the years a plethora of different architectures have been used. CentralNet (Vielzeuf et al., 2018) and Refnet (Sankaran et al., 2021) are multimodal fusion designs based on aggregative multi-task learning. Khattar et al. (2019) used ideas from unsupervised learning to use multimodal autoencoders to learn better representations. Architectures based on knowledge graphs have also been proposed for fusion methods in the ecological context (Nitta et al., 2020). Pérez-Rúa et al. (2019) suggest an architecture search approach to build a multimodal model by combining layers from multiple unimodal pipelines. MBT (Nagrani et al., 2021) incorporates bottlenecked fusion tokens into the multimodal transformer by (Tsai et al., 2019a). MBT is a strictly transformer-based method, while HyperFuse is an universal enhancement with hypernetworks to integrate multimodal information, that is applicable to a broader range of models.

Model Agnostic Methods A number of alignment and information based losses have also been explored to improve fusion by inducing semantic relationships across the different unimodal representations (Abavisani et al., 2019; Bramon et al., 2011; Liang et al., 2021b; Liu et al., 2021; Han et al., 2021). These are purely train-time objectives and can be generally applied to most multimodal fusion models. Wang et al. (2020a) tackle the problem of weighing modalities during learning when different unimodal networks have varying capacity. Wu et al. (2022) adresses a similar problem of balancing utilization rates. Unlike these works, we focus on learning conditional representations instead of balancing between modalities.

Using Auxiliary Geographical Data Minetto et al. (2019) introduces metadata to a geospatial land classification task, and Salem et al. (2020) integrates dense overhead imagery with location and date into a general framework by concatenating the individual representations. Mac Aodha et al. (2019) extracts location features by MLP to produce a prior distribution for fine-tuning the original predictions. In GeoNet (Chu et al., 2019), the geolocation priors, post-processing models, and feature modulation models are utilized to leverage the additional information.

4. HyperFuse

In Section 4.1, we explain the difference between the commonly used embedding-based fusion and our hypernetwork-based approach in HyperFuse, and then highlight the advantages of such hypernetwork design. Next, Section 4.2 details the implementation of the HyperFuse architecture. Finally in Section 4.3 we present details about the HyperFuse Block as part of the architecture.

4.1. Hypernetwork-based vs. Embedding-based Fusion

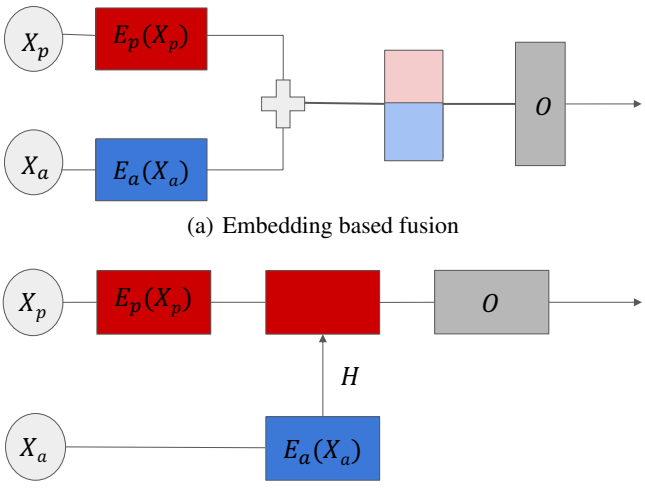


Figure 2: Two approaches to build multimodal neural nets. $\mathbf{x}_p$ and $\mathbf{x}_a$ refer to the primary and auxiliary modalities, respectively.

Consider a multimodal model $\mathcal{M} : \mathbb{X}_p \times \mathbb{X}_a \rightarrow \mathbb{Y}$ that combines a primary input $x_p \in \mathbb{X}_p$ with an auxiliary input $x_a \in \mathbb{X}_a$ and outputs $y \in \mathbb{Y}$. The embedding-based

approach builds a model of the following form:

$$\mathcal{M}(\mathbf{x}_p, \mathbf{x}_a; \theta_p, \theta_a, \theta_o) = O(E_p(\mathbf{x}_p; \theta_p), E_a(\mathbf{x}_a; \theta_a); \theta_o)$$

It is composed of a predictive neural network O and embedding modules for each modality E_p and E_a. The embeddings are combined together before applying output network O. An example is depicted in Figure 2(a). Most existing multimodal fusion approaches follow this pattern, varying modality-specific embedders and fusion functions to combine the embeddings.

Our alternate approach for fusion via hypernetworks is shown in Figure2(b). We have a primary network O, whose weights are produced by the separate hypernetwork H. This corresponds to a model of the form:

$$\mathcal{M}(\mathbf{x}_p, \mathbf{x}_a; \theta_p, \theta_a) = O(E_p(\mathbf{x}_p; \theta_p); \tilde{\theta}_p) \text{ s.t. } \tilde{\theta}_p = H(\mathbf{x}_a; \theta_a)$$

Different from the embedding approach, the hypernetwork approach formulates the mapping as a conditionally parameterized function.

Motivation Theoretically, a very wide network can extract all types of features from an image, which can then be masked with auxiliary information from other modalities(Scarselli and Tsoi, 1998). However such models have significantly high parametric and sample complexity (Galanti and Wolf, 2020; Rath and Condurache, 2022). In cases of limited embedding networks, an ideal model would switch its representations to highlight more desirable features while suppressing the less informative ones based on the auxiliary information. Galanti and Wolf (2020) proved that while an embedding method requires an increasing complexity to guarantee convergence, under certain conditions, a hypernetwork can keep the network complexity under control. This ability to effectively learn conditional functions (i.e., model a function that transforms into different functions depending on the condition X_a) is labeled as modularity. However, Galanti and Wolf (2020) consider a multi-task setting with a partitioned output space, whereas we are considering a single task setting with continuous auxiliary modalities.

Consider our earlier example of bird classification, when distinguishing between two similar waterfowls (say a duck and goose), representations that focus on specific features like feather and down shape, bill length etc. would be preferable. However, while dealing with environmentally different species like a duck and an owl, general features are sufficient. Similarly, a different set of features are important to distinguish between desert birds compared to aquatic birds. In an embedding based model, the image encoder will need to extract all such features together, leading to extremely high dimensional representations. Instead of learning such representations one can instead use conditional representations. The modularity property associated with Hypernetworks (Galanti and Wolf, 2020) suggests that this can be achieved efficiently with Hypernetworks. As such we use hypernetworks to generate filters parameterized on auxiliary modalities. Using auxiliary information such as geographical coordinates or habitat information one can generate filters for extracting more pertinent input representations [1].

4.2. HyperFuse Architecture

We propose a hypernetwork-based fusion architecture, called *HyperFuse* and presented visually in Fig. 3. Our proposed architecture keeps a primary path and an auxiliary path for processing the information from the primary and additional modalities respectively. Unlike previous works, our proposed method does fusion via one or multiple HyperFuse Blocks to modulate the primary feature pipeline with information from auxiliary modalities. The architecture fits the form of most existing multimodal networks where the input modalities are embedded into a vector, and then transformed further.

Usually hypernetworks are used to produce the entire set of parameters for a primary network. However, that approach is impractical and also incompatible with using pre-trained networks like ResNet or BERT. Instead we are going to parameterize only individual HyperFuse blocks which act as standalone components. The overall network structure is composed of multiple such HyperFuse blocks in an iterative style. This allows us to stack together fusion layers similar to how single MLP layers are stacked in a deep neural network. The blocks can also be interleaved with other standard neural networks providing flexibility.

The HyperFuse design takes $\mathbf{z}_p^0$ (primary) and $\mathbf{z}_a$ (auxiliary) as the initial inputs, where the primary feature is dynamically modulated with the corresponding auxiliary feature. We obtain the enhanced primary representation $\mathbf{z}_p^N$ after N iterative HyperFuse blocks. The multimodal output $\mathbf{z}_p^N$ matches the shape of the original input representation $\mathbf{z}_p$. A skip connection (Srivastava et al., 2015) is added between the multimodal embedding and the original representation, before applying the head prediction network O. This enables use of deeper pipelines, and allows the model to suppress the cross-modal projections (Srivastava et al., 2015).

Unlike traditional neural network layers with fixed parameters for all instances, the weights in our proposed model are dynamically generated and conditioned based on the instance-wise auxiliary information. Dynamically adjusting parameters can ease the recognition difficulty for similar inputs by extracting more pertinent representations. Experiments show that our method learns more generalized and

[1] While this might seem similar to a multi-task setting, we are still in a single-task setting with a high number of output species

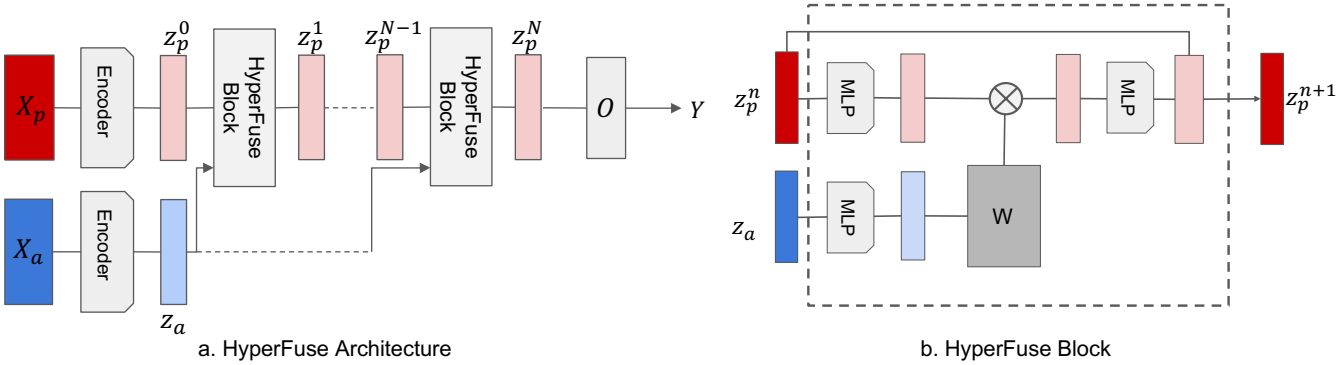


Figure 3: HyperFuse Architecture. There are two separate network pathways for the primary and auxiliary modalities. HyperFuse Block is a fusion layer between the two paths that produces fused representations. The HyperFuse block is deployed iteratively, where N denotes the number of such blocks. The final output from primary pathway is input to output net O.

distinguishing representations and surpasses existing works that utilize the traditional embedding-fusion design.

Proposition 4.1. *Consider a bounded Lipschitz and continuously differentiable function $\mathcal{Y}_{target}$ which is being learnt using a network with a single fusion block with two inputs of dimensionalities d_1 and d_2. If the fusion block is replaced by a linear HyperFuse block, then the network complexity of the HyperFuse approach is $\mathcal{O}(\varepsilon^{-d_1/r} + \varepsilon^{-d_2/r})$ compared to embedding based fusion methods $\mathcal{O}(\varepsilon^{-(d_1+d_2)/r})$.*

For derivation and more discussion we refer the readers to Appendix C.

4.3. HyperFuse Block Design

HyperFuse Block is designed to replace or add a layer in any existing model pipeline while simultaneously providing an MLP layer that transforms its inputs conditionally based on another modality. To allow a HyperFuse block to be placed at any position in a network pipeline (including pre-trained ones), we aim to ensure that there is no dimension change or information loss. To accomplish this, a HyperFuse block uses residual connection with the primary modality's latent features. The auxiliary modality encoding is kept unchanged, so that the training of the individual hypernetworks in the blocks and the downstream auxiliary pipeline are decoupled. Multiple blocks can be stacked and an MLP can be used between the blocks on the auxiliary pathway if needed.

Figure 3(b), illustrates the architecture of a single block. As mentioned earlier, we can chain these blocks where the primary input for the current layer is the output of the previous one. For the auxiliary input for each block, we pass it through to the next layer. We use a bottleneck architecture where both the primary and auxiliary inputs are projected to a smaller hidden dimension h for the intermediate HyperFuse blocks compared to the input dimension. This forces the parameter generator, within a block, to condense and extract most relevant information from each. This also allows scalability as h governs the size of the hypernetwork.

Each HyperFuse block takes as input the primary and auxiliary feature and outputs the fused feature for one iteration. The final output of the block is based upon applying a linear projector (shown as the matrix W) which is applied on the primary modality pipeline. The parameters of this projector is generated based on the auxiliary feature as $\mathbf{W} = H(\mathbf{z}_a)$ $H(\cdot)$ denotes the weight generating function whose outputs is reshaped into a 2-d matrix. A residual connection (He et al., 2016a) is also added between the input to a HyperFuse block and its output. This helps training when stacking multiple HyperFuse blocks in a deep network (He et al., 2016b). Furthermore it retains the original primary representations allowing the model to skip an individual block if needed.

If $\mathbf{z}_p^n$ is taken to be the primary representation updated n times iteratively as shown in Figure 3 by the HyperFuse layer, where $n \in \{1, 2, ..., N\}$; then the HyperFuse block can be written mathematically as:

$$\begin{aligned} \mathbf{W} &= H(\mathbf{z}_a) \\ \hat{z}_p^{n+1} &= \text{ELU}(\mathbf{W}^T \times \text{LN}(g(\mathbf{z}_p^n))) \\ \mathbf{z}_p^{n+1} &= \text{ELU}(\text{LN}(g(\hat{z}_p^{n+1}))) + \mathbf{z}_p^n \end{aligned}$$

where ELU is the exponential linear unit and LN refers to normalization layer like Layer-norm (Ba et al., 2016) or batch norm (Ioffe, 2017).

We performed ablation experiments with related block designs, by removing residual connections, combining primary and auxiliary modalities for parameter generation as well as other changes. In the Appendix E we provide details about other block variants and the corresponding results.

5. Experiments

We evaluate HyperFuse over a variety of base architectures and domains ranging from multimedia classification, species labelling, sentiment analysis and finance. For each dataset we use a standard model and replace all fusion layers in the model by HyperFuse blocks. In general we took the dominant or best single modality as the primary modality. We experiment with the number of HyperFuse blocks and hidden units of each HyperFuse block, and report the results from the best model. Details on hyperparameters can be found in the Appendix B.

5.1. Multimedia

Dataset We evaluate HyperFuse on **AV-MNIST** (Vielzeuf et al., 2018), a popular benchmark dataset used for multimodal fusion (Pérez-Rúa et al., 2019; Joze et al., 2020). It is an audio-visual dataset for a digit classification task. The data is prepared by pairing human utterances of digits obtained from FSDD dataset [2] with images of written digits from MNIST. This dataset has 55K training, 5K validation, and 10K testing examples. We use the processing stack of Cassell (2019) to prepare the dataset. The preprocessing involves adding corruption to both modalities so that more than one modality is required (Vielzeuf et al., 2018) for prediction. However, the image modality is still the dominant modality. For example, an image-only model achieves around 60% accuracy, compared to the 40% accuracy of audio-only model.

Models LF is the baseline late fusion architecture used in Vielzeuf et al. (2018), while MFM refers to the factorization method of Tsai et al. (2019b). GB and MBT are the gradient blending and vision-transformer based approaches of Wang et al. (2020a) and Nagrani et al. (2021). Similar to existing work (Liang et al., 2021a) our experiments used the aforementioned fusion designs with unimodal LeNet style feature generators (details present in the Appendix B.1). We take image as the primary modality Z_p and audio spectrogram as the auxiliary modality Z_a.

Result Table 1 demonstrates the superiority of HyperFuse over aforementioned fusion mechanisms on AV-MNIST. With the images as the primary modality and the audio records as the auxiliary modality, HyperFuse achieves the best average performance over 5 trials. HyperFuse even improves over the SOTA model MFAS [3]. On the other hand the MBT model struggles, likely due to the small data size. A larger comparison with errors and a bigger set of models in presented in the Appendix.

[2] https://www.tensorflow.org/datasets/catalog/spoken_digit

[3] We do not run a search from scratch and instead use the final architecture described in Pérez-Rúa et al. (2019)

Model	Accuracy
LF	71.4
MFM	71.4
GB	68.9
MBT	70.3
MFAS	72.1
LF + HyperFuse	**72.4**

Table 1: Accuracy results on digit classification task with AV-MNIST for various fusion architectures. The performance was averaged over five trials. HyperFuse outperforms the others by better utilizing the image and audio modality.

5.2. Fine-Grained Image Classification

Datasets We experiment with three different fine-grained classification tasks. Two of these are subsets of the classic YFCC100M(Thomee et al., 2016) dataset, the YFCC-MINI and the YFCC-GEO100 (Tang et al., 2015). The third is the iNaturalist species classification task (Van Horn et al., 2018; 2021). The images were considered the primary modality Z_p for these experiments, and any other information, such as geographic location, time, etc., was deemed the auxiliary modality Z_a. For inputs that were missing the auxiliary information, following Mac Aodha et al. (2019); Yan et al. (2021) we replaced the missing value with 0.

Models We use a pre-trained ResNet (He et al., 2016a) and apply the HyperFuse block at its penultimate layer. The design we follow is based on the model for fine-grained classification used in (Tang et al., 2015). As baseline models, we use DynamicMLP, GeoNet, and EnsembNet. GeoNet (Chu et al., 2019) and EnsembNet (Terry et al., 2020) are two common deep-learning models for this task. GeoNet incorporates geolocation priors from Mac Aodha et al. (2019) and uses feature modulation to utilize additional information. EnsembNet (Terry et al., 2020) uses an ensemble approach for species classification and is applicable only to iNat and GEO. DynamicMLP (Yang et al., 2022a) is recent model for fine-grained species classification, that uses dynamic filters (Jia et al., 2016b). We also include the concatenation-based fusion design of Tang et al. (2015) as baseline.

Result In Table 2 we present the top-1 and top-5 classification accuracy of our proposed method and other approaches. The hyperparameter information for HyperFuse are in Appendix B.3, while the rest of the results are presented from existing works. Our method achieves SOTA or near-SOTA results on multiple datasets with ResNet backbone. Our **ResNet+HyperFuse**) model achieve almost a 1 point improvement over the SOTA DynamicMLP on iNat and YFCCGEO. Similar improvements are obtained with a SK-Res2Net (Li et al., 2019) backbone for the image modality as well (see Appendix D).

	YFCC-MINI		YFCC100M-GEO100		iNat	
	Acc1	Acc5	Acc1	Acc5	Acc1	Acc5
UniModal	32.6	52.2	47.6	77.9	64.5	85.4
GeoNet	N/A	N/A	49.1	80.2	75.1	91.2
DynamicMLP	40.2	60.2	**53.2**	83.2	78.1	93.1
Concat	38.4	58.9	48.7	79.8	77.3	92.7
EnsembNet	N/A	N/A	50.8	81.9	73.7	89.9
ResNet + HyperFuse	**40.6**	**60.9**	**53.2**	**84.8**	**79.3**	**93.5**

Table 2: Results on fine-grained classification on the YFCC-MINI, YFCC100M-GEO100 and the iNaturalist datasets. Acc1 and Acc5 represent the top-k accuracy with k=1 and k=5 respectively. HyperFuse outperforms previous multimodal works.

5.3. Affective Computing

Datasets We evaluate our methods on two commonly used datasets for multimodal sentiment evaluation. **CMU-MOSI** (Wöllmer et al., 2013) is sentiment prediction tasks on a set of short youtube video clips.**CMU-MOSEI** (Zadeh et al., 2018b) is a similar dataset consisting of around 23K review videos taken from YouTube. The output in both cases is a sentiment score in $[-3, 3]$. For each dataset, three modalities are available; audio, video, and text. Preliminary features on audio, visual and textual modalities are obtained via COVAREP (Degottex et al., 2014), FACEt (iMotion) and word embeddings using Glove (Pennington et al., 2014) or BERT (Devlin et al., 2018).

	$Acc_7 \uparrow$	$Acc_2 \uparrow$	MAE $\downarrow$	CORR $\uparrow$
FLSTM	31.2	75.9	1.01	0.64
MFN	31.3	76.6	1.01	0.62
MAGBERT	40.2	83.7	0.79	0.80
MAGXLNET	43.1	85.2	0.76	**0.82**
BERT + HyperFuse	40.5	84.2	1.02	0.80
XLNET + HyperFuse	**43.8**	**85.3**	**0.76**	**0.82**

(a) CMU-MOSI

	$Acc_7 \uparrow$	$Acc_2 \uparrow$	MAE $\downarrow$	CORR $\uparrow$
FLSTM	44.1	75.1	0.72	0.52
MFN	44.3	74.7	0.72	0.52
MAGBERT	46.9	83.1	0.59	0.76
MAGXLNET	46.7	83.9	0.59	**0.77**
DynMM	N/A	79.8	0.6	N/A
BERT + HyperFuse	**52.7**	84.8	**0.58**	0.76
XLNET + HyperFuse	52.4	**85.3**	0.59	0.76

(b) CMU-MOSEI

Table 3: Results on sentiment analysis on CMU-MOSI/CMU-MOSEI. Acc_7 denotes accuracy on 7 classes and Acc_2 the binary accuracy. MAE denotes the Mean Absolute Error and Corr is the Pearson correlation. Augmenting BERT/XLNET with HyperFuse performs best.

Models MAGBERT/MAGXLNET (Rahman et al., 2020) are state of the art BERT (Devlin et al., 2018) based architecture that uses the MAG gate (Wang et al., 2019) to compute modified embeddings which are passed to a transformer based model. FLSTM (Narayanan et al., 2019) is a baseline fusion LSTM design, MFN (Zadeh et al., 2018a) is a memory based fusion, while the recent DynMM model from (Xue and Marculescu, 2022) uses dynamic gating. In this experiment we take the text embeddings as the primary modality Z_p and the audio and video features as the auxiliary modality Z_a.

Result Table 3 present the 2 and 7 class accuracies along with the MAE and correlation. We observe consistent improvements in accuracy of by using HyperFuse based embeddings (**BERT/XLNET + HyperFuse**) on the state of the art transformer models MAGBERT/MAGXLENT. These improvements range from 0.1% to 0.7%.

5.4. Financial Data

Datasets We evaluate the impact of adding the HyperFuse layer on a recently released Merger and Acquisition dataset M3A (Sawhney et al., 2021). The dataset comprises 816 conference calls with three modalities: transcript text (primary), speaker audio, and speaker information. There are two tasks: stock volatility prediction and price movement classification, both measured over 3, 7, and 15 days.

Models We experiment by adding a HyperFuse layer to the baseline architecture M3ANet. In the HyperFuse block, we have text as the primary modality Z_p^n and concatenation of audio, position embeddings, and speaker information as the auxiliary modality Z_a^n. The HyperFuse block is added to the architecture before the Attention-Fusion and Sentence-level Transformer steps. The speaker information serves as the context to determine the importance of a text or audio segment (whether the utterance comes from a decision-maker in the company), and has been used in the HyperFuse layer. We also report the results of the original M3ANet, some variations of Transformer and the Multimodal Deep Regression Model (MDRM) (Qin and Yang, 2019).

Result As shown in Table 4, **M3ANet + HyperFuse** outperforms the original M3ANet as well as Transformers in both stock volatility prediction task and price movement

Model	Volatility Prediction			Price Movement Classification					
	$MSE_3 \downarrow$	$MSE_7 \downarrow$	$MSE_{15} \downarrow$	$F1_3 \uparrow$	$F1_7 \uparrow$	$F1_{15} \uparrow$	$MCC_3 \uparrow$	$MCC_7 \uparrow$	$MCC_{15} \uparrow$
MDRM (T+A)	0.78	0.58	0.46	0.59	0.58	0.46	**0.19**	0.19	0.11
Transformer (T+A: Concat)	0.80	0.61	0.48	0.09	0.16	0.06	0.00	0.01	0.01
Transformer (T+A: Att fusion)	0.76	0.58	0.47	0.57	0.61	0.55	0.16	0.18	0.12
M3ANet	0.77	0.57	0.46	**0.59**	0.58	0.50	0.18	0.17	0.13
M3ANet + HyperFuse	**0.75**	**0.53**	**0.44**	0.51	**0.63**	**0.58**	0.16	**0.20**	**0.16**

Table 4: Mean τ-day volatility MSE and price movement prediction results (mean of 5 runs for each approach)

classification over multiple prediction periods. Besides improving the performance of the original architecture, adding the HyperFuse layer achieves better results than the MDRM model as well.

5.5 Exploratory Analysis

In this section, we present some exploratory analyses of HyperFuse using the M3A and AV-MNIST datasets. First, we test whether using an auxiliary modality always shows better performance compared to unimodal models of the primary modality. Next, we test whether the position of the HyperFuse block(s) added to the base pipeline affects the outcome performance. These results are reported in Table 5 (columns 2 and 4). The results indicate that in any configuration of the HyperFuse architecture the model that uses HyperFuse block(s) to fuse the auxiliary modality into the primary one is always more accurate. These results show that while there exists an optimal position for the fusion layer in both datasets, the boost over unimodal models is greater than inter-model differences.

Since HyperFuse architecture requires the choice of a primary modality among multiple ones, we also perform ablations to test whether switching the role of primary and auxiliary modalities impacts the model performance. These are also reported in Table 5 (the second row indicates choice of primary modality). The results (in Table 5) show significant difference in performance between the best model using image/text as primary modality and the best model using audio as primary modality.

	AV-MNIST (Accuracy ↑)		M3A ($MSE_7 \downarrow$)	
Position ↓	Image	Audio	Text	Audio
Unimodal	66.7	42.5	0.62	0.61
1	71.9	71.5	**0.53**	0.58
2	**72.6**	71.4	0.57	0.58
3	72.4	70.7	N/A	N/A

Table 5: Ablation studies of performance on AV-MNIST and M3A varying primary modalities and positions of HyperFuse blocks in the primary network. M3A pipeline is shorter so only has 2 positions for a fusion layer. The choice of primary modality is depicted on the second row

Finally, we hypothesize that a hypernetwork based fusion produces more distinguishable embeddings than embedding based fusion. To verify this we evaluate the distance between embeddings of different AV-MNIST classes for MFM and HyperFuse, and present a heatmap in Figure 10. We can see that using conditional transformations with HyperFuse produces a greater average distance between the classes, which explains why hyperfuse performs better. Similar visualization on iNat are available in Appendix E.3.

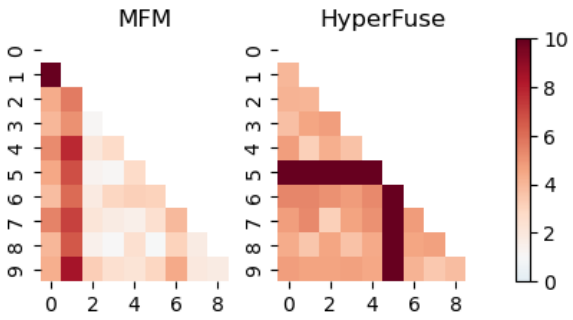


Figure 4: Heatmap of average distances between embeddings of different digits for MFM and HyperFuse. Darker colours represent greater distances. HyperFuse produces greater interclass separation which explains higher accuracy.

6. Conclusion

In this paper, we present a different approach for multimodal based on hypernetworks to integrate cross-modal information from different sources. Our method, called HyperFuse, predicts dynamic weights for a deep neural network backbone that processes the primary modality conditioned on auxiliary information. This approach naturally addresses the challenge of learning high-dimensional representations for modality conditioning. HyperFuse is a model-agnostic fusion layer that generates more discriminative representations than other common fusion techniques. Our method shows promising improvement over SOTA or near-SOTA models on a wide range of tasks, including image classification, sentiment prediction, and volatility prediction.

References

Abavisani, M., Joze, H. R. V., and Patel, V. M. (2019). Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1165–1174.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

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

Bertinetto, L., Henriques, J. a. F., Valmadre, J., Torr, P., and Vedaldi, A. (2016). Learning feed-forward one-shot learners. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 523–531. Curran Associates, Inc.

Bramon, R., Boada, I., Bardera, A., Rodriguez, J., Feixas, M., Puig, J., and Sbert, M. (2011). Multimodal data fusion based on mutual information. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1574–1587.

Brock, A., Lim, T., Ritchie, J., and Weston, N. (2018). SMASH: One-shot model architecture search through hypernetworks. In *International Conference on Learning Representations*.

Cassell, S. (2019). Mfas. `https://github.com/slyviacassell/_MFAS/`.

Chang, H.-S. and McCallum, A. (2022). Softmax bottleneck makes language models unable to represent multi-mode word distributions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8048–8073.

Chang, O., Flokas, L., and Lipson, H. (2020). Principled weight initialization for hypernetworks. In *International Conference on Learning Representations*.

Chaplot, D. S., Sathyendra, K. M., Pasumarthi, R. K., Rajagopal, D., and Salakhutdinov, R. (2018). Gated-attention architectures for task-oriented language grounding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., and Liu, Z. (2020). Dynamic convolution: Attention over convolution kernels. In *CVPR*.

Chu, G., Potetz, B., Wang, W., Howard, A., Song, Y., Brucher, F., Leung, T., and Adam, H. (2019). Geo-aware networks for fine-grained recognition. In *ICCV*.

Degottex, G., Kane, J., Drugman, T., Raitio, T., and Scherer, S. (2014). Covarep—a collaborative voice analysis repository for speech technologies. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 960–964. IEEE.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dwaracherla, V., Lu, X., Ibrahimi, M., Osband, I., Wen, Z., and Roy, B. V. (2020). Hypermodels for exploration. In *International Conference on Learning Representations*.

Ehret, B., Henning, C., Cervera, M. R., Meulemans, A., Von Oswald, J., and Grewe, B. F. (2020). Continual learning in recurrent neural networks. *arXiv preprint arXiv:2006.12109*.

Fiterau, M., Bhooshan, S., Fries, J., Bournhonesque, C., Hicks, J., Halilaj, E., Re, C., and Delp, S. (2017). Shortfuse: Biomedical time series representations in the presence of structured information. In *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pages 59–74. PMLR.

Galanti, T. and Wolf, L. (2020). On the modularity of hypernetworks.

Gao, S., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., and Torr, P. H. (2019). Res2net: A new multi-scale backbone architecture. *IEEE TPAMI*.

Garbacea, C. and Mei, Q. (2020). Neural language generation: Formulation, methods, and evaluation. *arXiv preprint arxiv.2007.15780*.

Garcia, A., Essid, S., d'Alché Buc, F., and Clavel, C. (2019). A multimodal movie review corpus for fine-grained opinion mining. *arXiv preprint arXiv:1902.10102*.

Gondal, M., Rahaman, N., Joshi, S., Gehler, P., Bengio, Y., Locatello, F., and Schölkopf, B. (2021). Dynamic inference with neural interpreters. *Advances in Neural Information Processing Systems*, 34.

Ha, D., Dai, A., and Le, Q. V. (2016). Hypernetworks. *arXiv preprint arXiv:1609.09106*.

Han, W., Chen, H., and Poria, S. (2021). Improving multimodal fusion with hierarchical mutual information maximization for multimodal sentiment analysis. In *Proceedings of EMNLP 2021*.

Hanin, B. and Sellke, M. (2018). Approximating continuous functions by relu nets of minimal width. *Arxiv*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *CVPR*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer.

Hessel, J. and Lee, L. (2020). Does my multimodal model learn cross-modal interactions? it's harder to tell than you might think!

Hou, M., Tang, J., Zhang, J., Kong, W., and Zhao, Q. (2019). Deep multimodal multilinear fusion with high-order polynomial pooling. *Advances in Neural Information Processing Systems*, 32:12136–12145.

Hu, R., Andreas, J., Darrell, T., and Saenko, K. (2018). Explainable neural computation via stack neural module networks. In *Proceedings of the European conference on computer vision (ECCV)*.

Huang, Y., Xie, K., Bharadhwaj, H., and Shkurti, F. (2021). Continual model-based reinforcement learning with hypernetworks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 799–805. IEEE.

Ioffe, S. (2017). Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. *Advances in neural information processing systems*, 30.

Jayakumar, S. M., Czarnecki, W. M., Menick, J., Schwarz, J., Rae, J., Osindero, S., Teh, Y. W., Harley, T., and Pascanu, R. (2020). Multiplicative interactions and where to find them. In *International Conference on Learning Representations*.

Jia, X., De Brabandere, B., Tuytelaars, T., and Gool, L. V. (2016a). Dynamic filter networks. *NeurIPS*.

Jia, X., De Brabandere, B., Tuytelaars, T., and Gool, L. V. (2016b). Dynamic Filter Networks. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 667–675. Curran Associates, Inc.

Joze, H. R. V., Shaban, A., Iuzzolino, M. L., and Koishida, K. (2020). Mmtm: Multimodal transfer module for cnn fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13289–13299.

Khattar, D., Goud, J. S., Gupta, M., and Varma, V. (2019). Mvae: Multimodal variational autoencoder for fake news detection. In *The world wide web conference*, pages 2915–2921.

Li, X., Wang, W., Hu, X., and Yang, J. (2019). Selective kernel networks. In *CVPR*.

Li, Y., Gu, S., Zhang, K., Gool, L. V., and Timofte, R. (2020). Dhp: Differentiable meta pruning via hypernetworks. In *European Conference on Computer Vision*, pages 608–624. Springer.

Liang, P., Lyu, Y., Fan, X., Wu, Z., Cheng, Y., Wu, J., Chen, L., Wu, P., Lee, M., Zhu, Y., et al. (2021a). Multibench: Multiscale benchmarks for multimodal representation learning.

Liang, T., Lin, G., Feng, L., Zhang, Y., and Lv, F. (2021b). Attention is not enough: Mitigating the distribution discrepancy in asynchronous multimodal sequence fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8148–8156.

Littwin, G. and Wolf, L. (2019). Deep meta functionals for shape representation. In *The IEEE International Conference on Computer Vision (ICCV)*.

Liu, J., Chen, W., Cheng, Y., Gan, Z., Yu, L., Yang, Y., and Liu, J. (2020). Violin: A large-scale dataset for video-and-language inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10900–10910.

Liu, Y., Fan, Q., Zhang, S., Dong, H., Funkhouser, T., and Yi, L. (2021). Contrastive multimodal fusion with tupleinfonce. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 754–763.

Liu, Z., Shen, Y., Lakshminarasimhan, V. B., Liang, P. P., Zadeh, A., and Morency, L.-P. (2018). Efficient low-rank multimodal fusion with modality-specific factors. *arXiv preprint arXiv:1806.00064*.

Loshchilov, I. and Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc.

Mac Aodha, O., Cole, E., and Perona, P. (2019). Presence-only geographical priors for fine-grained image classification. In *ICCV*.

Maiorov, V., Meir, R., and Ratsaby, J. (1999). On the approximation of functional classes equipped with a uniform measure using ridge functions. *J. Approx. Theory*, 99(1):95–111.

Mhaskar, H. N. (1996). Neural networks for optimal approximation of smooth and analytic functions. *Neural Comput.*, 8(1):164–177.

Minetto, R., Segundo, M. P., and Sarkar, S. (2019). Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Transactions on Geoscience and Remote Sensing*.

Nachmani, E. and Wolf, L. (2020). Molecule property prediction and classification with graph hypernetworks. *Arxiv*.

Nagrani, A., Yang, S., Arnab, A., Jansen, A., Schmid, C., and Sun, C. (2021). Attention bottlenecks for multimodal fusion. *Advances in Neural Information Processing Systems*, 34.

Narayanan, A., Siravuru, A., and Dariush, B. (2019). Temporal multimodal fusion for driver behavior prediction tasks using gated recurrent fusion units. *CoRR*, abs/1910.00628.

Nitta, N., Nakamura, K., and Babaguchi, N. (2020). Constructing geospatial concept graphs from tagged images for geo-aware fine-grained image recognition. *ISPRS International Journal of Geo-Information*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. (2018). Film: Visual reasoning with a general conditioning layer. In *AAAI*.

Pérez-Rúa, J.-M., Vielzeuf, V., Pateux, S., Baccouche, M., and Jurie, F. (2019). Mfas: Multimodal fusion architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6966–6975.

Prakash, A., Chitta, K., and Geiger, A. (2021). Multi-modal fusion transformer for end-to-end autonomous driving. In *CVPR*.

Qin, Y. and Yang, Y. (2019). What you say and how you say it matters: Predicting stock volatility using verbal and vocal cues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 390–401, Florence, Italy. Association for Computational Linguistics.

Rahman, W., Hasan, M. K., Lee, S., Zadeh, A., Mao, C., Morency, L.-P., and Hoque, E. (2020). Integrating multimodal information in large pretrained transformers. In *Proceedings of the meeting of Association for Computational Linguistics*, volume 2020, page 2359.

Rath, M. and Condurache, A. P. (2022). Improving the sample-complexity of deep classification networks with invariant integration. *arXiv preprint arXiv:2202.03967*.

Safran, I. and Shamir, O. (2017). Depth-width tradeoffs in approximating natural functions with neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2979–2987, International Convention Centre, Sydney, Australia. PMLR.

Salem, T., Workman, S., and Jacobs, N. (2020). Learning a dynamic map of visual appearance. In *CVPR*.

Sankaran, S., Yang, D., and Lim, S.-N. (2021). Multimodal fusion refiner networks.

Sawhney, R., Goyal, M., Goel, P., Mathur, P., and Shah, R. R. (2021). Multimodal multi-speaker merger & acquisition financial modeling: A new task, dataset, and neural baselines. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6751–6762, Online. Association for Computational Linguistics.

Scarselli, F. and Tsoi, A. C. (1998). Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural networks*, 11(1).

Schmidhuber, J. (1992). Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139.

Soleymani, M., Garcia, D., Jou, B., Schuller, B., Chang, S.-F., and Pantic, M. (2017). A survey of multimodal sentiment analysis. *Image and Vision Computing*, 65:3–14.

Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Highway networks.

Stanley, K. O., D'Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based encoding for evolving large-scale neural networks. *Artificial life*, 15(2):185–212.

Storks, S., Gao, Q., and Chai, J. Y. (2019). Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*.

Sun, Z., Sarma, P., Sethares, W., and Liang, Y. (2020). Learning relationships between text, audio, and video via deep canonical correlation for multimodal language

analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8992–8999.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *CVPR*.

Tang, K., Paluri, M., Fei-Fei, L., Fergus, R., and Bourdev, L. (2015). Improving image classification with location context. In *ICCV*.

Terry, J. C. D., Roy, H. E., and August, T. A. (2020). Thinking like a naturalist: Enhancing computer vision of citizen science images by harnessing contextual data. *Methods in Ecology and Evolution*.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. (2016). Yfcc100m: The new data in multimedia research. *Communications of the ACM*.

Tian, Z., Shen, C., and Chen, H. (2020). Conditional convolutions for instance segmentation. *arXiv preprint arXiv:2003.05664*.

Tsai, Y.-H. H., Bai, S., Liang, P. P., Kolter, J. Z., Morency, L.-P., and Salakhutdinov, R. (2019a). Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2019, page 6558. NIH Public Access.

Tsai, Y.-H. H., Liang, P. P., Zadeh, A., and Morency, L.-P. (2019b). Learning factorized multimodal representations. *arXiv preprint arXiv:1906.0617*.

Ukai, K., Matsubara, T., and Uehara, K. (2018). Hypernetwork-based implicit posterior estimation and model averaging of cnn. In *Proceedings of Machine Learning Research*, volume 95, pages 176–191. PMLR.

Van Horn, G., Cole, E., Beery, S., Wilber, K., Belongie, S., and Mac Aodha, O. (2021). Benchmarking representation learning for natural world image collections. *arXiv preprint arXiv:2103.16483*.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. (2018). The inaturalist species classification and detection dataset. In *CVPR*.

Vielzeuf, V., Lechervy, A., Pateux, S., and Jurie, F. (2018). Centralnet: a multilayer approach for multimodal fusion. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0.

Von Oswald, J., Henning, C., Sacramento, J., and Grewe, B. F. (2019). Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*.

Wang, W., Tran, D., and Feiszli, M. (2020a). What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12695–12705.

Wang, X., Zhang, R., Kong, T., Li, L., and Shen, C. (2020b). Solov2: Dynamic and fast instance segmentation. *arXiv preprint arXiv:2003.10152*.

Wang, Y., Shen, Y., Liu, Z., Liang, P. P., Zadeh, A., and Morency, L.-P. (2019). Words can shift: Dynamically adjusting word representations using nonverbal behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7216–7223.

Wöllmer, M., Weninger, F., Knaup, T., Schuller, B., Sun, C., Sagae, K., and Morency, L.-P. (2013). Youtube movie reviews: Sentiment analysis in an audio-visual context. *IEEE Intelligent Systems*, 28(3):46–53.

Wu, N., Jastrzebski, S., Cho, K., and Geras, K. J. (2022). Characterizing and overcoming the greedy nature of learning in multi-modal deep neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162. PMLR.

Xu, C., Tao, D., and Xu, C. (2015). Multi-view learning with incomplete views. *IEEE Transactions on Image Processing*, 24(12):5812–5825.

Xue, Z. and Marculescu, R. (2022). Dynamic multimodal fusion.

Yan, X., Hu, S., Mao, Y., Ye, Y., and Yu, H. (2021). Deep multi-view learning methods: A review. *Neurocomputing*, 448:106–129.

Yang, B., Bender, G., Le, Q. V., and Ngiam, J. (2019). Condconv: Conditionally parameterized convolutions for efficient inference. *arXiv preprint arXiv:1904.04971*.

Yang, L., Li, X., Song, R., Zhao, B., Tao, J., Zhou, S., Liang, J., and Yang, J. (2022a). Dynamic mlp for fine-grained image classification by leveraging geographical and temporal information.

Yang, L., Li, X., Song, R., Zhao, B., Tao, J., Zhou, S., Liang, J., and Yang, J. (2022b). Dynamic mlp for fine-grained image classification by leveraging geographical and temporal information.

Zadeh, A., Chen, M., Poria, S., Cambria, E., and Morency, L.-P. (2017). Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*.

Zadeh, A., Liang, P. P., Mazumder, N., Poria, S., Cambria, E., and Morency, L.-P. (2018a). Memory fusion network for multi-view sequential learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zadeh, A., Liang, P. P., Poria, S., Vij, P., Cambria, E., and Morency, L.-P. (2018b). Multi-attention recurrent network for human communication comprehension. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, Y., Zhang, J., Wang, Q., and Zhong, Z. (2020). Dynet: Dynamic convolution for accelerating convolutional neural networks. *arXiv preprint arXiv:2004.10694*.

Zhao, D., von Oswald, J., Kobayashi, S., Sacramento, J., and Grewe, B. F. (2020). Meta-learning via hypernetworks.

Zhong, V., Rocktäschel, T., and Grefenstette, E. (2020). Rtfm: Generalising to new environment dynamics via reading. In *International Conference on Learning Representations*.

Supplementary Material

A. Related Works

Tensor Fusion These methods seek to utilize higher-order multiplicative interactions for capturing complementary relationships between modalities. However, due to the high computational cost of such tensor products, several approximations have been proposed that trade-off between efficiency and flexibility (Hou et al., 2019; Liu et al., 2018). For example, LFN (Zadeh et al., 2017) combined information via pooling projections of high dimensional tensor representation of multimodal features. Multiplicative models that generalize tensor products to include learnable parameters have also been proposed (Jayakumar et al., 2020) to capture multimodal interactions. A variety of other models (Perez et al., 2018; Zhong et al., 2020) can be interpreted as restrictions of these general multiplicative models.

Multimodal gated units Gating mechanism learns representations that dynamically change for every input (Chaplot et al., 2018; Wang et al., 2019). A general gated unit can be written as a scaled version of the input vectors like $\mathbf{z}_p \odot h(\mathbf{z}_a)$, where h represents a scaling function such as sigmoid activation which amplifies and suppresses different components of $\mathbf{z}_p$. The MAGBERT/MAGXLNET models (Rahman et al., 2020) used gating to adjust the input embeddings to a language model and achieve SOTA results on sentiment classification. Recently, Xue and Marculescu (2022) explored gating-based methods to save computation and induce data-dependent computation. Our HyperFuse approach learns to predict dynamic parameters for the fusion layer and can support instance-specific computation.

Dynamic Models Our method is directly related to ShortFuse (Fiterau et al., 2017), which improved biomedical time series models by leveraging the structured covariates to produce dynamic filters. Convolution layers which learn dynamic parameters for multiple kernels have been successfully used in a number of vision problems such as object detection (Sun et al., 2020; Wang et al., 2020b) and segmentation (Tian et al., 2020).Jia et al. (2016a) dynamically generates the filters conditioned on the input images. Other similar models including CondConv (Yang et al., 2019), DyNet (Zhang et al., 2020), and Dynamic Conv (Chen et al., 2020) learn dynamic parameters for multiple kernels. However, these are mostly unimodal models and derive adaptive filters from a single spatial feature map (Prakash et al., 2021; Yang et al., 2022a). Instead, ours uses a primary network whose parameters are generated by an auxiliary network conditioned on other modalities. Models like Hu et al. (2018) and Gondal et al. (2021) have also been used for dynamic inference, but these are not designed for multimodal setting and focus on multi-task learning.

Fusion Architectures Due to the wide variety of applications and tasks which require multimodal fusion, over the years a plethora of different architectures have been used. CentralNet (Vielzeuf et al., 2018) and Refnet (Sankaran et al., 2021) are multimodal fusion designs based on aggregative multi-task learning. Khattar et al. (2019) used ideas from unsupervised learning to use multimodal autoencoders to learn better representations. Architectures based on knowledge graphs have also been proposed for fusion methods in the ecological context (Nitta et al., 2020). Pérez-Rúa et al. (2019) suggest an architecture search approach to build a multimodal model by combining layers from multiple unimodal pipelines. MBT (Nagrani et al., 2021) incorporates bottlenecked fusion tokens into the multimodal transformer by (Tsai et al., 2019a). MBT is a strictly transformer-based method, while HyperFuse is an universal enhancement with hypernetworks to integrate multimodal information, that is applicable to a broader range of models.

Optimization-Based Approaches A number of alignment and information based losses have also been explored to improve fusion by inducing semantic relationships across the different unimodal representations (Abavisani et al., 2019; Bramon et al., 2011; Liang et al., 2021b; Liu et al., 2021; Han et al., 2021). These are purely train-time objectives and can be generally applied to most multimodal fusion models. Wang et al. (2020a) tackle the problem of weighing modalities during learning when different unimodal networks have varying capacity. Wu et al. (2022) adresses a similar problem of balancing utilization rates. Unlike these works, we focus on learning conditional representations instead of balancing between modalities.

B. Hyperparam details

B.1. AV-MNIST

For the AVMNIST dataset, we used LeNet style unimodal feature generators. For the image encoder we used a 4 layer network with filter sizes $[5, 3, 3, 3]$ and max-pooling with width of 2. For the audio encoder the networks was a 6 layer networks with filter sizes $[5, 3, 3, 3, 3, 3]$ and max-pooling of width 2. The channel width was doubled after each layer. For the optimization process we tried random search on a logarithmic scale on the interval $[1e-5, 5e-2]$. We experimented with Adam, Adagrad, RMSProp, SGD optimizer with default configurations. For the HyperFuse network parameters we used models of upto 3 blocks with hidden units in $\{2, 4, .., 128, 256\}$.

For the MFAS model, we did not do architecture search but instead used the final model presented by Pérez-Rúa et al. (2019). That model is shows in Figure 5. While we have tried to stay close to the method described in Pérez-Rúa et al. (2019); Liang et al. (2021a) for creation of this dataset, our version of AVMNIST is potentially different from the earlier reported results as no standard dataset is available.

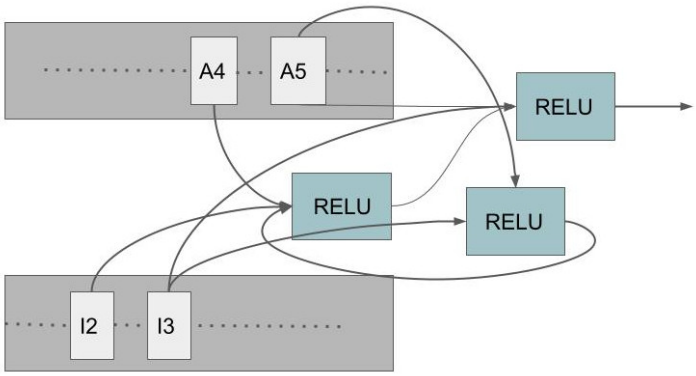


Figure 5: MFAS (Pérez-Rúa et al., 2019) based Multimodal Fusion Architecture for AVMNIST. Every arrow into the activation corresponds to a linear layer. The A4 and A5 represent the fourth and fifth layer of the audio encoder. Similarly I2 and I3 represent the second and third layer of the image encoder.

B.2. M3A

For training the model we use the M3ANet architecture, and combine the embeddings with the HyperFuse architecture. For the parameters relevant to the M3ANet model , we keep the same hyperparameters of M3ANet as in Sawhney et al. (2021) and experiment with 2 HyperFuse hyperparamters: HyperFuse MLP block type and number of hidden units in HyperFuse block. For each task and τ-day period, we run models with different combinations of 4 HyperFuse block types $\{\text{Base}, I, III, IV\}$ and range of number of hidden units $\{2, 4, 8, 16\}$.

B.3. Fine Grained Image Classification

We use the same training setting and procedure for all three datasets. For training the models, we use augmented training data as inputs. The augmented images are obtained via random corruptions like random crop of 224 by 224 pixels and flips (Szegedy et al., 2015). We further use the Mixup (Zhang et al., 2017) training process and interpolate the augmented images. Finally label smoothing is also applied with smoothing param of 0.1. All networks are trained using SGD optimizer with momentum and weight decay. The weight decay was chosen by a log scale random search from the interval $[1e-5, 1e-3]$. The auxiliary information such as coordinates, time etc. were encoded following the procedure of Mac Aodha et al. (2019). For training,we use the learning rate scheduling of Yang et al. (2022b), which combines warmup with cosine decay. Specifically tthe learning rate is set to 4×10^{-2} with a linear warmup (He et al., 2016a) for five epochs and a cosine decay schedule (Loshchilov and Hutter, 2016).

C. Proof of Proposition4.1

In this section, we briefly introduce the requisite Theorems from Galanti and Wolf (2020), before showing that these results imply Proposition 4.1. These results are extensions of known results in approximation theory (Mhaskar, 1996; Lu et al., 2017; Hanin and Sellke, 2018; Safran and Shamir, 2017) that quantify tradeoffs between the number of trainable parameters, width and depth of the neural networks as universal approximators. The notation and description here follows that of Galanti and Wolf (2020).

Notations We have a target function $Y_{\text{trgt}} : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$ available only through samples of input output pairs. Here, $x_1 \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$ are two inputs from different modalities. Let $\mathcal{W}_{r,n}$ be the Sobolev space of n dimensional multivariate functions with smoothness r. $\mathcal{W}_{r,n}$ is a vector space of functions $h : [-1, 1]^n \to \mathbb{R}$ with continuous partial derivatives of orders up to r, such that, the norm is bounded, $\|h\|_r^s := \|h\|_\infty + \sum_{1 \le |\mathbf{k}|_1 \le r} \|D^{\mathbf{k}} h\|_\infty \le 1$, where $D^{\mathbf{k}}$ denotes the partial derivative indicated by the multi–integer $\mathbf{k} \ge 1$, and $|\mathbf{k}|_1$ is the sum of the components of $\mathbf{k}$. We will assume that our target function Y_{trgt} belongs to this space.

An **embedding method** is a network of the form $h(x_1, x_2; \theta_e, \theta_q) = q(x_1, e(x_2; \theta_e); \theta_q)$, consisting of a composition of neural networks q and e parameterized with real-valued vectors $\theta_q \in \Theta_{\mathbf{q}}$ and $\theta_e \in \Theta_{\mathbf{e}}$ (resp.). $e(x_2; \theta_e)$ serves as an embedding of x_2. For two given families of embedding functions $\mathbf{e} := \{e(I; \theta_e)\}$ and prediction function $\mathbf{q} := \{q(x, z; \theta_q)\}$, let $\mathcal{E}_{\mathbf{e},\mathbf{q}} := \{q(x, e(I; \theta_e); \theta_q) \mid \theta_q \in \Theta_{\mathbf{q}}, \theta_e \in \Theta_{\mathbf{e}}\}$ denote the embedding method formed by them.

A **hypernetwork** $h(x_1, x_2) = g(x_1; f(x_2; \theta_f))$ is a pair of collaborating neural networks, $f : \mathcal{X}_2 \to \Theta_{\mathbf{g}}$ and $g : \mathcal{X}_1 \to \mathbb{R}$, such f produces the weights of g. The function $f(x_2; \theta_f)$ takes a conditioning input x_2 and returns the parameters $\theta_I \in \Theta_{\mathbf{g}}$ for g. The network g takes an input x_1 (and the parameters θ_I) and returns output $g(x_2; \theta_I)$ that depends on x_1 and via θ_I on x_2. In practice, f is typically a large neural network and g is a small neural network.

We shall denote by σ the activation function used in the nerual network. We assume that σ is a universal, piece-wise $C^1(\mathbb{R})$ activation function with $\sigma' \in BV(\mathbb{R})$ and $\sigma(0) = 0$. $BV(\mathbb{R})$ is defined to be the set of functions of bounded variation. Following Galanti and Wolf (2020) we also assume that any non-constant $Y \in \mathbb{Y}$ cannot be represented as a neural network with σ activations.

Theorem C.1. *Let $\mathcal{E}_{\mathbf{e},\mathbf{q}}$ be a neural embedding method. Assume that $\mathbf{e}$ is a class of continuously differentiable neural network e with zero biases, output dimension $k = \mathcal{O}(1)$ and $\mathcal{C}(e) \le \ell_1$ and $\mathbf{q}$ is a class of neural networks q with σ activations and $\mathcal{C}(q) \le \ell_2$. Let $\mathbb{Y} := \mathcal{W}_{1,m}$. Assume that any non-constant $y \in \mathbb{Y}$ cannot be represented as a neural network with σ activations. If the embedding method achieves error $d(\mathcal{E}_{\mathbf{e},\mathbf{q}}, \mathbb{Y}) \le \varepsilon$, then, the complexity of $\mathbf{q}$ is: $N_{\mathbf{q}} = \Theta\left(\varepsilon^{-(m_1+m_2)}\right)$.*

Theorem C.1 is a restatement of Theorem 2 and 3 from Galanti and Wolf (2020). The results in this theorem is restrict to the Sobolev space r = 1, which is the space of mean bounded, Lipschitz and continuously differentiable functions. The lower bound on the complexity in this theorem follows from well known results in of approximation theory (Mhaskar, 1996; Maiorov et al., 1999; Lu et al., 2017).

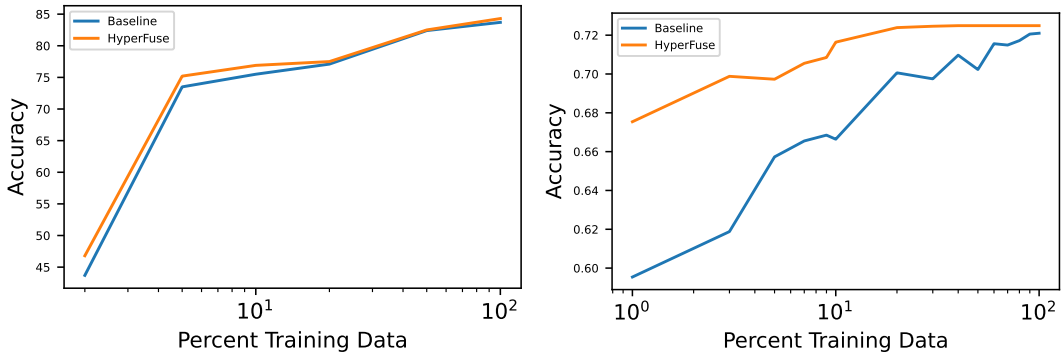
Theorem C.2. *Let σ be an activation as earlier. Let $y \in \mathbb{Y} = \mathcal{W}_{r,m}$ be a function, such that, y_{x_2} cannot be represented as a neural network with σ activations for all $x_2 \in \mathcal{X}_2$. Then, there is a class, $\mathbf{g}$, of neural networks with σ activations and a network $f(x_2; \theta_f)$ with ReLU activations, such that, $h(x_1, x_2) = g(x; f(x_2; \theta_f))$ achieves error $\le \varepsilon$ in approximating y and $N_{\mathbf{g}} = \mathcal{O}\left(\varepsilon^{-m_1/r}\right)$.*

Theorem C.2 is a restatement of Theorem 4 from Galanti and Wolf (2020). It shows that the minimal complexity required for approximating each individual smooth target function y_{x_2} is achievable by a hypernetwork based model.

Next, note that in the linear HyperFuse scenarios, the selection function $S(x_2)$ takes the form of $W \cdot h$, for some continuous function $h : \mathcal{X}_2 \to \mathbb{R}^w$ parameterized by the neural network g and W is a linear mapping (Ukai et al., 2018; Chang et al., 2020; Littwin and Wolf, 2019). If the true selector function for Y_{trgt} belongs to the family of g neural networks (or if it belongs $\mathcal{P}_{r,w,c}^{d_2,k}$) then it can be approximated by $\mathcal{O}\left(\varepsilon^{-d_2/r}\right)$ size network. Further since the selector is continuous, ε error in approximating the selector, further only leads to $\mathcal{O}\varepsilon$ error in the actual target. Correspondingly if the target function is according to Proposition 4.1, then we can apply Theorem C.2 to HyperFuse, giving an addition complexity of $\mathcal{O}\left(\varepsilon^{-d_1/r}\right)$. Combining Similarly, Theorem C.1 gives a tight bound on the complexity of the embedding based networks. Note that as per the description in Section 3, $\mathcal{X}_2 \subset \mathbb{R}^{d_2}$ and $\mathcal{X}_1 \subset \mathbb{R}^{d_1}$. Putting the corresponding dimensionalities into Theorems C.1C.2, the proposition follows

C.1. Validating Proposition 3.1

In this section, we want to empirically assess the validity of Propostion 3.1. For this purpose, we vary the number of samples available for training and measure how the model performance changes. According to Proposition 3.1, HyperFuse will have lower complexity, and hence should rise faster and saturate earlier than other models. We evaluate this hypothesis with AVMNIST and MOSI, and present the results in Figure 6. From the figure we can see that our hypothesis is correct. HyperFuse shows strong performance (even with ¡ 20% data), and has consistently greater performance improvement at low number of sampler.



(a) (b)

Figure 6: Accuracy of HyperFuse against best baseline model a) MOSI, b) AV-MNIST as the availability of training samples changes. HyperFuse has lower complexity, and has greater improvements at lower data availability.

D. Extended Results

M3A After running hyperparamter search for HyperFuse, we ran the $M3ANet + HyperFuse$ model 5 times with the optimal hyperparameters and records the mean and standard deviation of MSE_{test} below. The results for baseline models come from (Sawhney et al., 2021).

Model	Volatility Prediction			Price Prediction					
	MSE_3	MSE_7	MSE_{15}	$F1_3$	$F1_7$	$F1_{15}$	MCC_3	MCC_7	MCC_{15}
MDRM (T+A)	0.78 (0.005)	0.58 (0.003)	0.46 (0.002)	0.59	0.58	0.46	**0.19**	0.19	0.11
Transformer (T+A: Concat)	0.80 (0.0006)	0.61 (0.0006)	0.48 (0.0003)	0.09	0.16	0.06	0.00	0.01	0.01
Transformer (T+A: Att. fusion)	0.76 (0.0180)	0.58 (0.0140)	0.47 (0.0090)	0.57	0.61	0.55	0.16	0.18	0.12
M3ANet	0.77 (0.0180)	0.57 (0.0160)	0.46 (0.0110)	**0.59**	0.58	0.50	0.18	0.17	0.13
M3ANet + Hyperfuse (Ours)	**0.75 (0.0230)**	**0.53 (0.0397)**	**0.44 (0.0190)**	0.51	**0.63**	**0.58**	0.16	**0.20**	**0.16**

Table 6: Mean τ-day volatility MSE and price movement prediction results (mean and stdev. of 5 runs for each approach)

AV-MNIST We used the same parameters for the baseline as those used by Liang et al. (2021a), and repeated the experiment 5 times. The results and deviations are reported in Table 7

Model	Accuracy ↑
LFN	71.1 (0.3)
MFM	71.4 (0.6)
GB	68.9 (0.6)
Refnet	70.6 (0.5)
MFAS	72.1 (0.3)
HyperFuse	**72.4** (0.3)

Table 7: Results on digit classification task with AVMNIST for various fusion architectures.

	YFCC100M-GEO100		iNat	
	Acc1	Acc5	Acc1	Acc5
UniModal	54.5	83.4	74.2	91.7
DynamicMLP	56.8	85.9	**83.6**	95.6
GeoNet	56.1	84.9	79.0	93.8
EnsembNet	53.8	82.9	80.5	93.5
HyperFuse	**57.5**	**87.1**	83.5	**96.2**

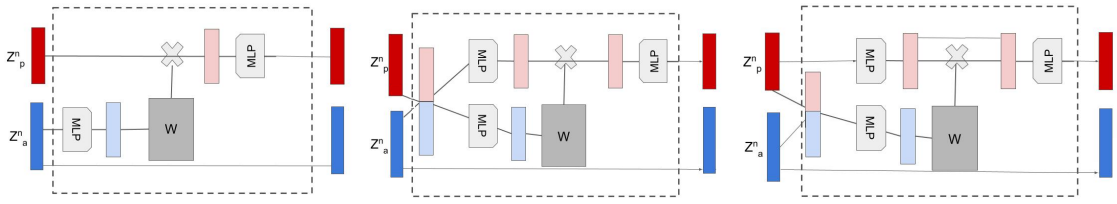
Table 8: Top-1 and Top-5 accuracies of HyperFuse against previous multimodal works on the YFCC100MGEO100 and the iNaturalist datasets with SK-Res2Net-101 (Li et al., 2019; Gao et al., 2019) backbone

Fine Grained Image Classification Due to the significantly higher cost of training on these datasets, we ran the process only twice and presented the average. The difference between the two runs in the same order as Table 2 are 0.2, 0.4, 0.3, 0.6, 0.4, and 0.6. Furthermore, since our baseline results are presented from existing literature which does not report the variance/deviation figures on these datasets, we report it only for HyperFuse.

We also present results on fine grained classification with a different image encoding network. Table 2 used a ResNet 50 based backbone network. In Table 8 we report results with Sk-Res2Net (Li et al., 2019; Gao et al., 2019).

E. Additional Experiments

E.1. Alternative Blocks



(a) Block II (b) Block III (c) Block IV

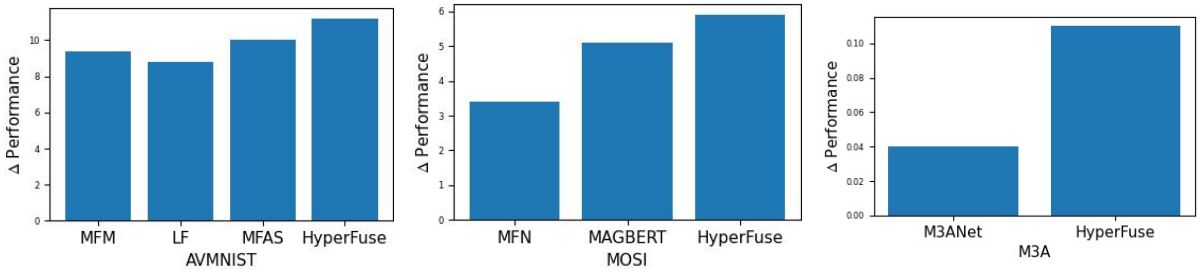
Figure 7: Alternate HyperFuse block design that we experiment with.with one auxiliary modality. The basic structure follows the design of the block in Figure 3(b), where W are parameters of a linear layer MLP applied to the primary modality.

In Figure 7, we present the design of some other HyperFuse blocks that were experimented with. Figure 7(a) present the simplest version of a hyperblock, where the auxiliary modality is directly transformed to a a matrix W which is applied to the primary modality. Figure 7(b) is a design motivated by the Dynamic MLP C design presented in Yang et al. (2022b). Note that in this block the modalities embeddings are concatenated before being transformed by MLPs and transfered back into the primary and auxiliary channels. As such this is an intermediate step between embedding based and hypernetwork based design. While this design does do well in one of the experiments, we found this to be slightly worse than the design in Figure 3(b). Moreover the concatenation between embeddings means the input modalities need to be compatible and hence this is not directly applicable on all tasks. Finally, since residual connections were found to be helpful we added them to Block III, and used purely primary modality information in the primary pipelines to make Bloc IV shown in Figure 7(c).

We present results on AV-MNIST, M3A and MOSI of the choices of different block designs. These are presented in Tables 10,12, 11 respectively. We generally see Block II to be lower performing than others. While all the other blocks perform equally well on AVMNIST, on M3A we see Block III and IV being worse. The more complex block models also perform worse on MOSI, while Block II performs similarly to MAGBERT. This can be due to the extra complexity of these blocks which allows it to overfit. From these two experiments, we choose to explore the block presented in Figure 3(b) (which we refer to also as Base Block) for the rest of the experiments.

E.2. Multimodality and Conditioning

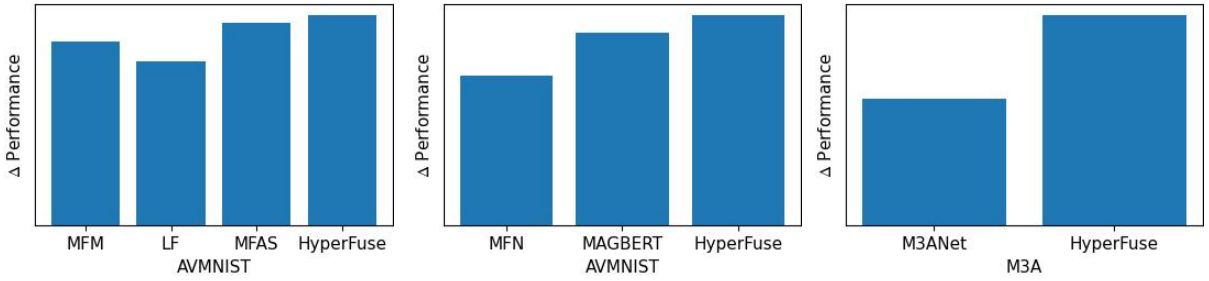
To assess the conditioning strength of HyperFuse against other models, we measure how the model performance changes when provided with wrong auxiliary information. For this purpose, we provide the model wrong inputs in the auxiliary modality by sampling from negative examples and compare the reduction in performance.



(a) (b) (c)

Figure 8: Bar graph of change in accuracy across different models for AVMNIST/MOSI/M3A when a wrong observation/features in auxiliary modality is provided as input. HyperFuse has greater sensitivity to the auxiliary modality, indicating a stronger conditioning behaviour.

We also perform the same experiment with gaussian noise corruption added to the auxiliary modalities instead of providing a negative sample. These results are presented in Figure . As expected the performance drop is lower than in the negative sample case. We also see that in this case HyperFuse has roughly the same drop as other models, indicating greater sensitivity to negative auxiliaries than noisy auxiliaries. This provides support to the hypothesis of greater conditional dependency on the auxiliary information.



(a) (b) (c)

Figure 9: Bar graph of change in accuracy across different models for AVMNIST/MOSI/M3A when a noisy features are provided as auxiliary modality. HyperFuse has greater sensitivity to the auxiliary modality, indicating a stronger conditioning behaviour.

Next we assess the multimodality behaviour of HyperFuse against other models. Hessel and Lee (2020) have demonstrated that most multimodal models often have limited cross-modal interactions. For this they develop a projection method, that isolates from the model predictions additive behaviour i.e. they extract predictions which can be expressed as a linear combination of unimodal outputs. We utilize their diagnostic criteria on HyperFuse and the best competing baseline to evaluate whether HyperFuse produces stronger cross-modal interactions. Since their criteria is more relevant to classification than regression, we focus on MOSI and AVMNIST here. As baseline we used the best-performing non-HyperFuse baseline on the corresponding dataset.

E.3. Ablation Experiments

Next, we present results for the ablation studies on AVMNIST and M3A. In these we explore the effect of choosing the position of the HyperFuse block (i.e. the fusion layer), as well as the impact to changing primary and auxiliary modalities. The results from these experiments are present in Table 13. These tables report accuracy on AVMNIST and MSE_7 for M3A.

Dataset	MOSI	AVMNIST
Baseline	1.7	1.3
HyperFuse	**2.8**	**1.9**

Table 9: Difference between model accuracy and accuracy of best additive multimodal function projection for the same model. Greater differences indicate stronger cross-modal interactions. We can see that HyperFuse has greater cross-modal interaction strength than other models

HyperFuse Block	Accuracy ↑
Base	**72.4** (0.3)
Block II	71.1 (0.4)
Block III	**72.5** (0.5)
Block IV	**72.5** (0.4)

Table 10: Results on digit classification task with AVMNIST for different HyperFuse blocks. The performance metric is Accuracy. Other than the simple Block I, others have similar performance . Base refers to the block in Figure 7

HyperFuse Block	Accuracy ↑
Base	**84.2** (0.4)
Block II	83.4 (0.5)
Block III	81.3 (0.6)
Block IV	80.1 (0.4)

Table 11: Results on sentiment detection task on MOSI for different HyperFuse blocks. The performance metric is Accuracy . Base refers to the block in Figure 7

Since the M3A data has no raw text, and instead uses text embeddings directly obtained from a pre-trained transformer, there are only two positions where one can fuse modalities. Unlike M3A, AVMNIST trains from scratch on a multilayer CNN and so has more fusion positions available. The results suggest that while there is an optimal position, the overall performance is not very sensitive to it. We also present results for the unimodal models, which were treated as the primary modality pipeline for HyperFuse, in the same table. One can see clear and substantial improvements over unimodal accuracies by using HyperFuse. Figure 10 presents a heatmap visualization of the inter-label distance on AVMNIST with our HyperFuse model. Once again, we see that HyperFuse produces greater embedding distances suggesting greater performance in discriminating between related species.

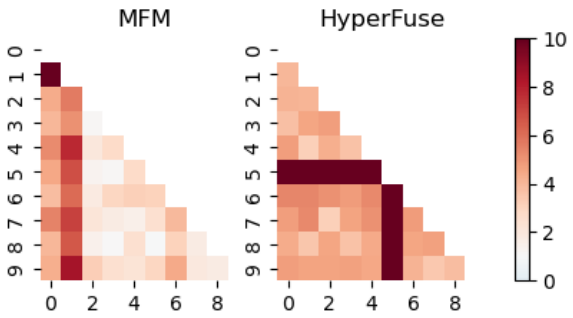
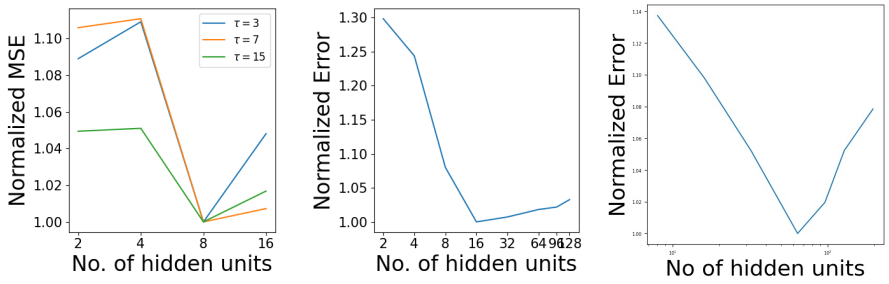


Figure 10: Heatmap of average distances between embeddings of different digits for MFM and HyperFuse. Darker colours represent greater distances. HyperFuse produces greater interclass separation which explains higher accuracy.

HyperFuse Block	MSE_3	MSE_7	MSE_{15}
Base	**0.707**	**0.546**	**0.437**
Block II	0.742	0.565	0.438
Block III	0.769	0.609	0.467
Block IV	0.779	0.548	0.443

Table 12: MSE of τ-day volatility prediction results for different HyperFuse block designs.



(a) (b) (c)

Figure 11: Relative error of HyperFuse on a) M3A, b) AV-MNIST, c) MOSI with varying size of hidden dimension h inside the HyperFuse block. Each point corresponds to the normalized error of the corresponding model with respect to the best model when the hidden dimension is set to the x-axis value.

Fusion Posn.	Acc (Image)	Acc (Audio)
1	71.9 (0.4)	71.5 (0.6)
2	**72.6** (0.3)	71.4 (0.5)
3	72.4 (0.3)	70.7 (0.3)
Unimodal	66.7 (0.6)	42.5 (0.9)

(a) AVMNIST

Fusion Posn.	MSE_7 (Text)	MSE_7 (Audio)
0	**0.53** (0.04)	0.58 (0.01)
1	0.57 (0.01)	0.58 (0.02)
Unimodal	0.62 (0.03)	0.61 (0.01)

(b) M3A

Table 13: Ablation study of performance on a) AVMNIST and b) M3A across the choice of primary modality and the position of fusion layer in the primary network. Pos -1 refers to the performance of unimodal models