Rich Semantics Improve Few-Shot Learning

Mohamed Afham [1,2]
afhamaflal9@gmail.com
Salman Khan [2,3]
salman.khan@mbzuai.ac.ae
Muhammad Haris Khan [2]
muhammad.haris@mbzuai.ac.ae
Muzammal Naseer [3,2]
muzammal.naseer@mbzuai.ac.ae
Fahad Shahbaz Khan [2,4]
fahad.khan@mbzuai.ac.ae

[1] Department of Electronic and Telecommunication Engineering, University of Moratuwa, Sri Lanka
[2] Mohamed Bin Zayed University of AI, UAE
[3] Australian National University, AU
[4] Linköping University, Sweden

Abstract

Human learning benefits from multi-modal inputs that often appear as rich semantics (e.g., description of an object's attributes while learning about it). This enables us to learn generalizable concepts from very limited visual examples. However, current few-shot learning (FSL) methods use numerical class labels to denote object classes which do not provide rich semantic meanings about the learned concepts. In this work, we show that by using 'class-level' language descriptions, that can be acquired with minimal annotation cost, we can improve the FSL performance. Given a support set and queries, our main idea is to create a bottleneck visual feature (hybrid prototype) which is then used to generate language descriptions of the classes as an auxiliary task during training. We develop a Transformer based forward and backward encoding mechanism to relate visual and semantic tokens that can encode intricate relationships between the two modalities. Forcing the prototypes to retain semantic information about class description acts as a regularizer on the visual features, improving their generalization to novel classes at inference. Further, this strategy imposes a human prior on the learned representations, ensuring that the model is faithfully relating visual and semantic concepts, thereby improving model interpretability. Our experiments on four datasets and ablations show the benefit of effectively modeling rich semantics for FSL. Code is available at: https://github.com/MohamedAfham/RS_FSL.

1 Introduction

Traditional classification models use class labels for supervision, expressed in a numerical form or as one-hot encoded vectors [13]. However, humans do not solely rely on such numerical class-labels to acquire learning. Instead, humans learn by communicating through natural language, which is grounded in a complex structure consisting of semantic attributes, relationships and abstract representations. Psychologists and cognitive scientists have argued natural language descriptions to be a central element of human learning [6, 21, 29]. The

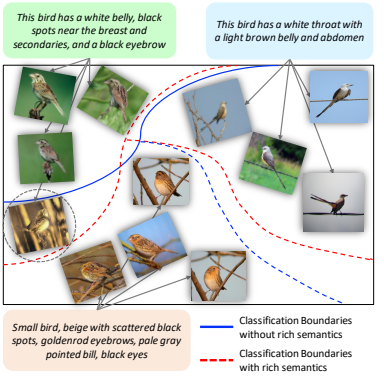

Figure 1: In FSL setting, where we require generalizability to novel classes with limited samples, modeling semantic attributes of classes can help disambiguate confusing classes. We suggest that the numerical class labels traditionally used in FSL are inadequate to represent diverse semantic attributes of an object class, which can be modeled via low-cost class-level language descriptions (colored boxes). Our approach effectively utilizes language information to learn both highly discriminative and transferable visual representations that help to avoid errors in ambiguous cases (e.g., visually similar fine-grained classes).

depiction of semantic class labels with numerical IDs leads to a *semantic gap* between the class semantic representation and the learned visual features.

We consider a few-shot learning setting where language descriptions for the seen (*base*) classes are available during training but not for the novel (*few-shot*) class that appear during inference. Remarkably, studying the potential of language descriptions has particular relevance to FSL, where a model must learn to generalize from few-samples and several categories can only be discriminated with subtle attribute-based differences (Fig. 1). We hypothesize that by predicting natural language descriptions as an auxiliary task during training, the model can learn useful representations that help transfer better to novel tasks during the inference stage. This helps the representations to explicitly model the shared semantics between the few-shot samples so that the class descriptions can be successfully generated.

The language description task while learning to classify images forces the model to attain following desirable attributes: (a) model high-level compositional patterns occurring in the visual data e.g., attributes in fine-grained bird classes; (b) avoid over-fitting on a given FSL task by imposing a regularizer demanding natural description from the class prototype; and (c) provide intuitive explanation for the learned class concepts e.g., the description of an object type, attributes, function and affordance in a human interpretable form. Importantly, the error feedback obtained from such a supervised task (natural language description) can help align a model with the '*human prior*'. Our RS-FSL approach is generic in nature and can be plugged into existing baselines models or with other multi-task objectives (e.g., equivariant self-supervised learning losses). Although we discard the language description module at inference, it is useful as a debugging tool to understand the model's behaviour in case of wrong predictions (*e.g.*, highlighting which attributes were mistaken or ambiguous).

Contributions. Our objective induces a generative task of natural language description for few-shot classes that forces the model to learn correlations between same class samples such that consistent class descriptions can be generated. The limited set of class-specific samples acts as a bottleneck that encourages extraction of shared semantics. We then introduce a novel transformer decoding approach for few-shot class description that relates the hybrid prototypes (obtained using the collective support and query image features) with the corresponding descriptions in both forward and backward directions. Our design allows modeling long-range dependencies between the sequence elements compared to previous recurrent architectures. Finally, our experiments on four datasets show consistent improvements across the FSL tasks. We extensively ablate our approach and analyze its different variants. The

proposed transformer decoder acts as a plug-and-play module and shows gains across popular FSL baselines namely ProtoNet [30], RFS [33] and Meta-baseline [4].

2 Related Work

Only a few-methods explore the potential of rich semantic descriptions in the context of FSL. A predominant approach has been the incorporation of unsupervised word embeddings or a set of manual attributes to represent class semantics. For example, [5] uses semantic embeddings of the labels or attributes to guide the latent representation of an auto-encoder as a regularizer mechanism. Xing *et al*. [38] dynamically learn the class prototypes as a convex combination of visual and semantic label embeddings (based on GloVe [24]). However, these embeddings require manual labeling (in case of attributes) or remain noisy if acquired via unsupervised learning. Additionally, representing rich semantics in a single vector remains less flexible to encode the complex semantics. In contrast, our approach flexibly learns the semantic representations with class-level language descriptions to improve upon the noisy unsupervised word embeddings. Schwartz *et al*. [28] extended [38] to exploit various semantics (category labels, text descriptions and manual attributes) in a joint framework. However, they use language descriptions as inputs rather than an extra supervision signal to train the visual backbone. Thus, these methods [28, 38] require attribute information or descriptions for novel classes during inference which can be hard to acquire for few-shot classes.

Image-level captions have been used in [8, 39] to align visual and semantic spaces with a multi-modal transformer model [14]. This can help learning from a limited set of base classes and scales to unseen classes [39]. [9] models annotator rationale as a spatial attention and the relevant attributes for a given input image. However, unlike our work, these methods do not study the FSL problem where image-level captions can cause overfitting. Furthermore, they require image-specific captions and rationales (not just "what" but also "why") which can be costly, even for a small number of base classes. Since acquiring high-quality explanations [1, 22] from experts can be expensive, efforts have been made to reduce the manual cost needed to acquire such annotations. To this end, ALICE model [19] acquires contrastive natural language descriptions from human annotators about the most informative pairs of object classes identified via active learning. In contrast, our approach only requires class-level descriptions that are easy to acquire compared to image level semantic annotations.

Andreas *et al*. [1] use the language descriptions during the pertaining stage in FSL to learn natural task structure. Once the model is pretrained to match images to natural descriptions, it can be used to learn new concepts by aligning natural descriptions with the images at inference. In contrast to inference stage alignment, LSL [22] introduced a GRU branch with language supervision to enrich the backbone features and discards the branch during inference. However, decoding mechanism in [1, 22] does not explicitly encode both forward and backward relations in the language, suffers in encoding long-term relationships and cannot relate multiple class-level descriptions with the visual class prototypes.

3 Proposed Method

3.1 Preliminaries

Problem Settings. In the standard *few-shot image classification* setting, we have access to a labelled dataset of base classes $\mathcal{C}_{base}$ with enough number of examples in each class, and the

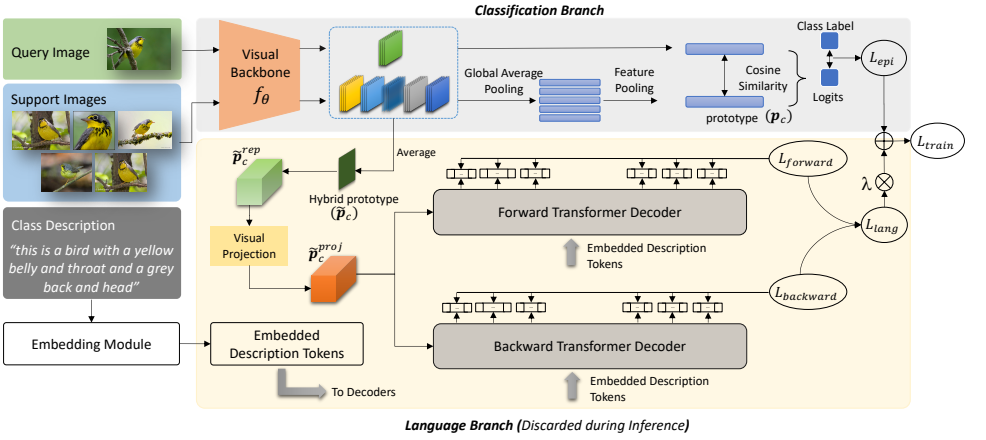

Figure 2: Overall architecture of the RS-FSL. Fundamentally, it consists of a visual backbone followed by a prototypical network to compute the classification loss (denoted by classification branch). We aim at encoding visual-semantic relationships that are in turn harnessed for enriching the visual features. To this end, we propose generating class-level language descriptions constrained on a hybrid prototype via developing a transformer based forward and backward decoding mechanism (denoted as language branch). Our method jointly trains the classification and language losses.

aim is to learn concepts in novel classes $\mathcal{C}_{novel}$ with few examples, given $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$.

Pretraining. Following the recent works in FSL [4, 36], we pretrain the *visual backbone*, parameterized by θ, on $\mathcal{C}_{base}$ in a supervised manner. We assume access to a dataset of image (x) and label $(y \in \mathcal{C}_{base})$ pairs: $\mathcal{D} = \{x_i, y_i\}_{i=1}^{M}$. The *visual backbone* maps the input image x to a feature embedding space $\mathbb{R}^d$ by an embedding function $f_\theta : x \to v$. In turn, the linear classifier $f_\Theta : v \to p$, parameterized by Θ, maps the features generated by f_θ to the label space $\mathbb{R}^L$ where L denotes the number of classes in $\mathcal{C}_{base}$. We optimize both θ and Θ by minimizing the standard cross-entropy loss:

$$\mathcal{L}_{pre}(p,y) = -\log \frac{\exp(p_y)}{\sum_j \exp(p_j)} \tag{1}$$

Episodic Training. After pretraining the visual backbone, we adopt the episodic training paradigm which has shown effectiveness for FSL. It simulates few-shot scenario faced at test time via constructing episodes by sampling small number of few-shot classes from a large labelled collection of classes $\mathcal{C}_{base}$. Specifically, each episode is created by sampling N classes from the $\mathcal{C}_{base}$ forming a support class set $\mathcal{C}_{supp} \subset \mathcal{C}_{base}$. Then two different example sets are sampled from these classes. The first is a *support-set* $S_e = \{(s_i, y_i)\}_{i=1}^{N \times K}$ comprising K examples from each of N classes, and the second is a *query-set* $Q_e = \{(q_j, y_j)\}_{j=1}^{Q}$ containing Q examples from the same N classes. The episodic training for few-shot classification boils down to minimizing, for each episode e, the loss of prediction on the examples in the query-set $(q_j, y_j) \in Q_e$, given the support set S_e:

$$\mathcal{L}_{epi} = \mathbb{E}_{(S_e, Q_e)} \sum_{j=1}^{Q} \log P_\theta(y_j | q_j, S_e). \tag{2}$$

Prototypical Networks. We develop proposed method on a popular metric-based meta-learning method named Prototypical network [30], owing to its simplicity and effectiveness.

However, ours is a plug-and-play training module which can work seamlessly with other FSL methods (as demonstrated in Sec. 4.2). Prototypical networks leverage the support set to compute a centroid (a.k.a *prototype*) for each class in a given episode, and query examples are classified based on distance to each prototype. The model is a convolutional neural network with parameters θ, that learns a d-dimensional space where examples from the same class are clustered together and those of different classes are far apart. Formally, for each episode e, a prototype $\mathbf{p}_c$ corresponding to class $c \in \mathcal{C}_{base}$ is computed by averaging the embeddings of all support samples belonging to class c:

$$\mathbf{p}_c = \frac{1}{|S_e^c|} \sum_{(s_i,y_i)\in S_e^c} f_\theta(x), \tag{3}$$

where f_θ is the pretrained visual backbone, and S_e^c is the subset of support belonging to class c. The model generates a distribution over N classes in an episode after applying softmax over cosine similarities between the embedding of the query q_j and the prototypes $\mathbf{p}_c$:

$$P_\theta(y=c|q_j,S_e) = \frac{\exp(\tau.\langle f_\theta(q_j),\mathbf{p}_c\rangle)}{\sum_k \exp(\tau.\langle f_\theta(q_j),\mathbf{p}_k\rangle)}, \tag{4}$$

where $\langle .,. \rangle$ is the cosine similarity, $k \in \mathcal{C}_{supp}$, and τ is the learnable parameter to scale the cosine similarity for computing logits [4]. The model is trained by minimizing Eq. 2. In the following section, we propose capturing rich and shared class-level semantics for FSL tasks via predicting natural language descriptions.

3.2 Capturing Rich Semantics for FSL

We show that by leveraging class-level semantic descriptions, the performance of FSL tasks can be improved. To this end, we create a bottleneck visual feature (termed hybrid prototype) to generate the language descriptions of classes as an auxiliary task. We introduce a language description branch featuring a Transformer based forward and backward decoding mechanism to connect hybrid prototypes with the corresponding descriptions both in forward and backward directions. This enforces the model to capture correlations between the same class examples so as to produce consistent class level descriptions.The hybrid prototype that is obtained using the support and query visual features facilitates the extraction of shared semantics. Furthermore, our language branch allows modelling long-range dependencies between the sequence of token vectors compared to prior recurrent architectures. We elaborate these components below.

Mapping visual features to language description. For each class $c \in \mathcal{C}_{base}$, we assume to have d_c *class-level* language descriptions $W_c = \{w_1, w_2, ..., w_{d_c}\}$. Each $w_i = (w_{i,1}, w_{i,2}, ...w_{i,T_i})$: $i \in \{1, d_c\}$ is a language description of variable length T_i tokens where $w_{i,1} = \langle s \rangle$ represents the start of the sentence token and $w_{i,T_i} = \langle /s \rangle$ denotes the end of the sentence token. Let $\tilde{\mathbf{p}}_c$ be the hybrid prototype formed after averaging the embeddings of support examples S_e^c and query examples Q_e^c of class c, as follows:

$$\tilde{\mathbf{p}}_c = \frac{1}{|S_e^c| + |Q_e^c|} \sum_{x\in(S_e^c \cup\ Q_e^c)} f_\theta(x). \tag{5}$$

Notably, the hybrid prototype $\tilde{\mathbf{p}}_c$ contrasts with $\mathbf{p}_c$ that only averages the support features. Our language module, associates the hybrid prototype $\tilde{\mathbf{p}}_c$ with the corresponding descriptions

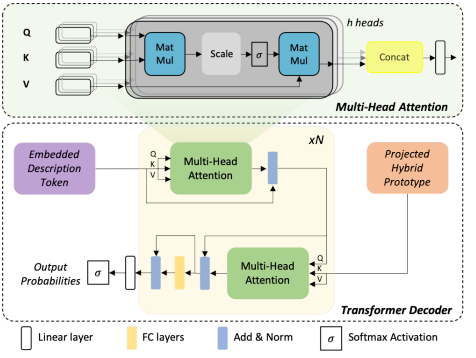

Figure 3: Transformer decoder architecture (bottom box) for generating class-level natural language descriptions based on multi-head attention (top box). We replicate the same architecture for both forward and backward decoding mechanisms. After the last Transformer layer, we apply a linear layer to get output un-normalized log probabilities over the token vocabulary.

W_c in both forward and backward directions. Specifically, the proposed transformer decoding function g_ϕ, parameterized by ϕ, takes the hybrid prototype $\tilde{\mathbf{p}}_c$ and predicts class semantic descriptions $\tilde{W}_c$. It comprises a forward and a backward model which allows it to generate each description $\tilde{w}_i$ token-by-token from left-to-right and right-to-left, respectively, and uses the following language loss for training:

$$\mathcal{L}_{lang}(\theta,\phi) = \frac{1}{2}\Big(\sum_{i=1}^{N}\sum_{t=2}^{T_i} -\log g_\phi\left(\tilde{w}_{i,t} = w_{i,t} \mid \tilde{\mathbf{p}}_c\right) + \sum_{i=1}^{N}\sum_{t=T_i-1}^{1} -\log g_\phi\left(\tilde{w}_{i,t} = w_{i,t} \mid \tilde{\mathbf{p}}_c\right)\Big).$$

We jointly train the (pretrained) visual backbone f_θ and the language description branch g_ϕ, in an episodic manner, after combining both the classification and language losses:

$$\mathcal{L}_{train} = \mathcal{L}_{epi} + \lambda \mathcal{L}_{lang}, \tag{6}$$

where λ balances the contribution of $\mathcal{L}_{lang}$ towards the joint loss $\mathcal{L}_{train}$. Our overall objective provides a learning signal that facilitates aligning the model with the human semantic prior of which representations are more transferable than others. We discard the language description branch after the training loop during inference. However, it can be beneficial towards understanding the model behaviour for incorrect predictions *e.g.* finding which attributes were mistaken during inference.

Transformer Decoder Architecture. Inspired by recent advances in language modelling, we propose to use Transformers [34], for decoding class-level descriptions in both forward and backward directions (Fig. 3). Transformers feature multi-head self-attention and not only can propagate the contextual information over sequence of description tokens but have the expressive capability to relate the hybrid prototype to semantic tokens in class-level descriptions. In a training episode, each image is accompanied with a corresponding class description. The hybrid prototype $\tilde{\mathbf{p}}_c$ of a given class is replicated to match the number of descriptions available in the episode and denoted by $\tilde{\mathbf{p}}_c^{rep}$. To reduce the complexity of the resulting feature tensor, it is projected through a linear layer. The projected hybrid prototype $\tilde{\mathbf{p}}_c^{proj}$ is then fed to the decoder module.

We embed the class-level descriptions using the Embedding Module (Fig. 2) which is initialised with pretrained GloVe [24]. The result is a set of embedded description tokens, which are fed to both forward and backward transformer decoders. The decoder first performs a multi-head self-attention over description token vectors and then applies multi-head attention between the projected hybrid prototype and descriptive token vectors. In each multi-head attention block, the inputs are transformed to query (**Q**), key (**K**) and value (**V**)

triplets using a set of transformation matrices. Attention mechanism is similar to [34] where the future elements of the description are masked to perform masked multi-head attention (see architecture in Fig. 3). It then applies a two-layer fully connected network to each vector. All these operations are followed by dropout, enclosed in a residual connection, and followed by layer normalization. After passing through transformer layers, we apply a linear layer which is common to both forward and backward decoders of each vector to produce un-normalized log probabilities over the token vectors. Following recent works [8], our transformer employs GELU activation [12] instead of ReLU.

4 Experiments

Datasets. CU-Birds [37] is an image dataset with 200 different birds species each having 40-60 images. Following [31], we split the available classes into 100 for training, 50 for validation and 50 for testing. **VGG-Flowers** is a fine-grained classification dataset comprising 102 flowers categories. Following [31], we split the dataset into 51 for training, 26 for validation and 25 for test classes. For both CUB and VGG-Flowers datasets, we acquire natural language descriptions for the images from [26] which provides 10 captions per image. Since our method leverages class-level descriptions, we sample the required number of descriptions from the (available) captions of the images belonging to each class, but those are consistently used for all the class images. **miniImageNet** [35] is a popular dataset for few-shot classification tasks. It consists of 100 image classes extracted from the original ImageNet dataset [7]. Each class contains 600 images of size 84×84. We follow the splitting protocol proposed by [35], and use 64 classes for training, 16 for validation, and 20 for testing. Since, class level descriptions for this dataset are unavailable, we manually gathered them from the web. Some representative examples of class-level descriptions for miniImageNet are shown in the supplementary material. **ShapeWorld** is a synthetic multi-modal dataset proposed by [16]. It consists of 9000, 1000, and 4000 few-shot tasks for training, validation and testing, respectively. Each task has a single support set of $K = 4$ images that are representing a visual concept with an associated natural language description, which we consider as the class-level descriptions. Each concept describes a spatial relation between two objects, and each object is optionally qualified by color and/or shape, with 2-3 distractor shapes around. The task is to predict whether a query image belongs to the concept or not.

Implementation Details. For fair comparisons with prior works, we deploy the following CNN architectures as visual backbones: 4-layer convolutional architecture proposed in [30] for CUB, ResNet-12 for miniImageNet [4, 17, 33], and ResNet-18 [31] for VGG-Flowers. For evaluation on all datasets, we use the challenging FSL setting of 5-way 1-shot and report accuracy averaged across few-shot tasks along with 95% confidence interval. During *pretraining* stage, we use SGD optimizer with an initial learning rate of 0.05, momentum of 0.9, and weight decay of 0.0005. We train the model for 100 epochs with a batch size of 64 and the learning rate decays twice by a factor of 0.1 at 60 and 80 epochs. Both forward and backward decoders are configured with a hidden layer size of 768, 12 attention heads, a feed-forward dimension of 3072 and with a 0.1 dropout probability. We use Adam optimizer with a constant learning rate of 0.0005 throughout and train the models for 600 epochs. Standard data augmentation e.g., random crop, color jittering and random horizontal flipping are applied during the meta-training stage. We fix $\lambda = 5$ in all experiments. τ is initialized as 1 for experiments with CUB and VGG-Flowers while for miniImageNet experiments it's initialized as 10. We use 2 layers of transformer decoders based on validation and study

miniImageNet			CUB		
Method	**Backbone**	**Accuracy**	**Method**	**Backbone**	**Accuracy**
ProtoNet [30]	Conv-4	55.50±0.70	MatchingNet [35]	Conv-4	60.52±0.88
Matching Net [35]	Conv-4	43.56±0.78	MAML [10]	Conv-4	54.73±0.97
MAML[10]	Conv-4	48.70±1.84	ProtoNet [30]	Conv-4	50.46±0.88
Chen *et al*. [3]	Conv-4	48.24±0.75	RFS [33]	Conv-4	41.47±0.72
Relation Net[32]	Conv-4	50.44±0.82	RelationNet [32]	Conv-4	62.34±0.94
TADAM [23]	ResNet-12	58.50±0.30	L3 [1]	Conv-4	53.96±1.06
MetaOptNet [17]	ResNet-12	62.64±0.61	LSL [22]	Conv-4	61.24±0.96
Boosting [11]	WRN-28-10	63.77±0.45	Chen *et al*. [3]	Conv-4	60.53±0.83
RFS-Simple [33]	ResNet-12	62.02±0.63	DN4-DA [18]	Conv-4	53.15±0.84
RFS-Distill [33]	ResNet-12	64.82±0.60	HP [15]	Conv-4	64.02±0.24
Meta-Baseline [4]	ResNet-12	63.17±0.23	Meta-Baseline [4]	Conv-4	59.30±0.86
RS-FSL	ResNet-12	**65.33±0.83**	RS-FSL	Conv-4	**65.66±0.90**

Table 1: Comparison with prior works on CUB and miniImageNet. Our method, RS-FSL, allows exploiting semantic information during training only.

the effect of different layers in Fig. 4. We use 20 *class-level* descriptions for both CUB and VGG-Flowers while for miniImageNet we use all 5 descriptions available per class. During inference, we discard the language description branch and rely on the visual backbone to perform few-shot classification. To be consistent with previous works [3, 31], we sample 600 few-shot tasks from the set of novel classes. For ShapeWorld dataset, following [22] we train for 50 epochs with a constant learning rate of 0.00005 with Adam optimizer. During training, we use $\lambda = 20$ and similar transformer decoder architecture parameters as the experiments in other datasets.

4.1 Comparison with state-of-the-art

We compare the performance of our method with eleven existing top performing approaches on CUB dataset in Tab. 1 (right). Our method delivers a significant improvement of 6.36% over a strong baseline method [4]. Tab. 1 (left) reports experimental results on miniImageNet. RS-FSL provides an improvement of 2.16% over the competitive baseline [4]. Compared to prior works, our method attains a higher accuracy of 65.33% and demonstrates the best performance. Experimental results for ShapeWorld dataset are reported in Tab. 2(b). RS-FSL outperforms the existing best performing method LSL [22] by a margin of 1.15%. Further, the results for VGG-Flowers dataset are shown in Tab. 2(a). RS-FSL performs favorably against all competing methods and achieves the best accuracy of 75.33%. Overall, RS-FSL consistently show gains on all four datasets. We note the improvement is more significant in CUB as compared to others since the class samples conform better with the language descriptions as compared to *e.g*., Flowers dataset. Thus our class-level descriptions show more effectiveness there. The increment on miniImageNet is relatively less pronounced due to the limited class descriptions obtained manually by us (only 5 per class). We also display some qualitative examples in supplementary material which show that RS-FSL encourages the model to focus on semantically relevant visual regions.

4.2 Analysis and Ablation Study

We perform all ablation experiments on CUB dataset with a Conv-4 visual backbone architecture following ProtoNet [30] baseline.

(a) Performance on VGG-Flowers

Method	Accuracy
ProtoNet [30]	72.38±0.98
Matching Net [33]	73.51±0.94
MAML[10]	65.46±1.05
Chen *et al.* [3]	74.09±0.84
Relation Net[31]	55.59±1.09
Meta-Baseline [4]	73.35±0.98
RS-FSL	**75.33±0.96**

(b) Performance on ShapeWorld

Method	Accuracy
ProtoNet [30]†	50.91±1.10
L3 [1]†	62.28±1.09
LSL [22]	63.25±1.06
RS-FSL	**64.40±0.99**

Table 2: Comparison on VGG-Flowers and ShapeWorld. † Results reported in [22].

The effect of different baselines and word embeddings. To demonstrate the generalizability of our approach, we show its improvements across three popular baseline FSL approaches both with and without using language prediction during training (see Tab. 3(a)). We note that the proposed language prediction mechanism constrained on a bottleneck visual feature (hybrid prototype) consistently improves the performance under all three baseline methods: ProtoNet [30], RFS [33], and Meta-Baseline [4]. Tab. 3(b) reports the impact on performance when using three different word embeddings to represent the words: Word2Vec [20] , GloVE [24], and fastText [2]. Our method retains similar accuracies under both Word2Vec and GloVe, however, it performs slightly inferior when deploying fastText. This reveals that RS-FSL is robust to the choice of word embeddings and favorable gains are obtained over the baseline method regardless of the embedding type used.

(a) Effect of different baselines

Baseline	Backbone	Without RS	With RS
ProtoNet [30]	Conv-4	57.97±0.96	63.86±0.91
RFS [33]	Conv-4	44.93±0.76	46.84±0.86
Meta-Baseline [4]	Conv-4	59.30±0.86	**65.66±0.90**

(b) Impact of Different Word embeddings

Word Embedding	Backbone	Accuracy	% gain over baseline
Word2Vec	Conv-4	63.28±0.95	5.31
GloVe	Conv-4	63.86±0.91	**5.89**
fastText	Conv-4	61.77±0.98	3.80

Table 3: (a) Performance of different baselines both with and without our rich semantic (RS) modeling and (b) performance when using three different word embeddings.

Number of class-level descriptions. Fig. 4 (left) shows that upon increasing the number of class-level descriptions from 1 to 20 the accuracy peaks to a maximum of 63.86%, however, it starts to saturate after increasing beyond 20. This could be because beyond a certain number of class-level descriptions, the semantic attributes collected from the available descriptions possibly become saturated, rendering the additional descriptions redundant.

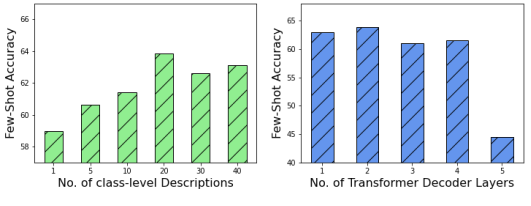

Figure 4: Performance upon varying the number of class-level descriptions (left) and the number of Transformer decoder layers (right).

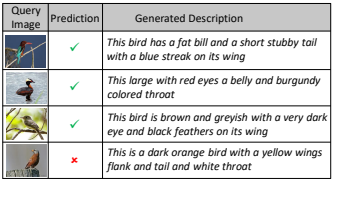

Figure 5: Class-level descriptions generated by RS-FSL for novel query images during inference.

Method	Backbone	Accuracy
Forward Decoder	Conv-4	62.03 ± 0.93
Bidirectional Decoder	Conv-4	63.86 ± 0.91

Table 4: Performance using only the forward decoding and the developed bidirectional decoding.

Fig. 4 (right) shows that our method retains the highest accuracy (63.86%) when employing two Transformer layers. However, it starts to deteriorate after further increasing the number of layers i.e. 3 to 5. The inferior performance is most likely due to over-fitting caused by the over-parameterization of the language description model given a relatively small dataset.

Language decoding mechanisms. We replace the bidirectional language decoding mechanism with just the forward decoding and observe that the former improves accuracy by 1.83% compared to latter (Tab. 4). Bidirectional decoding can better relate the visual cues with language semantics as it can model two-way interaction between the tokens, thereby facilitating the learning of generalizable visual representations vital for few-shot scenarios.

Auxiliary self-supervision. We compare the performance of different auxiliary self-supervised approaches with our proposed method (Tab. 5). The first auxiliary self-supervised task is predicting the rotation angle of the visual input [25, 27], and the others are predicting language using LSTM-GRU based recurrent architecture (ProtoNet +GRU) [27] and predicting the semantic word embedding corresponding to the prototype (ProtoNet + Word Embeddings). We observe that the proposed transformer based (bidirectional) language decoding mechanism significantly improves the performance (5.97%) over the method that is not using any auxiliary self-supervision (ProtoNet (without semantics)). Further, our approach outperforms the other auxiliary self-supervision methods, Proto+Rotation, Proto+Word Embeddings and Proto+GRU, by a margin of 4.6%, 3.61% and 2.62% respectively.

Fig. 5 shows class-level descriptions generated by RS-FSL for novel query images (in CUB dataset) during inference. We note that generated descriptions allow us interpreting the model behaviour for incorrect prediction, *e.g.* finding the bird attributes that are confused.

Method	Accuracy
ProtoNet (without semantics)	57.97±0.96
ProtoNet + Rotation	59.20±0.97
ProtoNet + Word Embeddings	60.25±0.93
ProtoNet + GRU [27]	61.24±0.96
RS-FSL + Class Descriptions	**63.86±0.91**

Table 5: Comparison between different auxiliary training methods. Average few-shot 5-way 1-shot accuracy reported with 95% confidence interval

Extra Cost vs Performance boost. We use Nvidia-RTX Quadro 6000 single-GPU for our training. We observed that training in miniImageNet dataset without auxiliary supervision took around 4 hours for training while RS-FSL took 5 hours with additional transformer decoder layers. Further our model obtains a performance boost of 2% over the baseline in miniImageNet (Tab. 1 left).

5 Conclusion

We presented a new FSL approach that models rich semantics shared across few-shot examples. This is realized by leveraging class-level descriptions, available with less annotation effort. We create a hybrid prototype which is used to produce class-level language predictions as an auxiliary task while training. We develop a Transformer based bi-directional decoding mechanism to connect visual cues with semantic descriptions to enrich the visual features. Experiments on four datasets show the benefit of our approach.

References

[1] Jacob Andreas, Dan Klein, and Sergey Levine. Learning with latent language. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2166–2179, 2018.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[3] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

[4] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9062–9071, 2021.

[5] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Multi-level semantic feature augmentation for one-shot learning. *IEEE Transactions on Image Processing*, 28(9):4594–4605, 2019.

[6] Seth Chin-Parker and Julie Cantelon. Contrastive constraints guide explanation-based category learning. *Cognitive science*, 41(6):1645–1655, 2017.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[8] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11162–11173, 2021.

[9] Jeff Donahue and Kristen Grauman. Annotator rationales for visual recognition. In *2011 International Conference on Computer Vision*, pages 1395–1402, 2011.

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

[11] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, P. Pérez, and M. Cord. Boosting few-shot visual learning with self-supervision. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8058–8067, 2019.

[12] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2020.

[13] Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah, and Mohammed Bennamoun. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, 8(1):1–207, 2018.

[14] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021.

[15] Valentin Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky. Hyperbolic image embeddings. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6417–6427, 2020.

[16] Alexander Kuhnle and Ann A. Copestake. Shapeworld - a new test methodology for multimodal language understanding. *ArXiv*, 2017.

[17] Kwonjoon Lee, Subhransu Maji, A. Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10649–10657, 2019.

[18] Wenbin Li, Lei Wang, J. Xu, Jing Huo, Y. Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7253–7260, 2019.

[19] Weixin Liang, James Zou, and Zhou Yu. Alice: Active learning with contrastive natural language explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4380–4391, 2020.

[20] Tomas Mikolov, Kai Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

[21] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[22] Jesse Mu, Percy Liang, and Noah Goodman. Shaping visual representations with language for few-shot classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[23] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, 2018.

[24] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[25] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. Self-supervised knowledge distillation for few-shot learning. *arXiv preprint arXiv:2006.09785*, 2020.

[26] Scott Reed, Zeynep Akata, Bernt Schiele, and Honglak Lee. Learning deep representations of fine-grained visual descriptions. In *IEEE Computer Vision and Pattern Recognition*, 2016.

[27] Mamshad Nayeem Rizve, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10836–10846, 2021.

[28] Eli Schwartz, Leonid Karlinsky, Rogerio Feris, Raja Giryes, and Alex M Bronstein. Baby steps towards few-shot learning with multiple semantics. *arXiv preprint arXiv:1906.01905*, 2019.

[29] Linda B Smith. Learning to recognize objects. *Psychological Science*, 14(3):244–250, 2003.

[30] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

[31] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. When does self-supervision improve few-shot learning? In *ECCV*, pages 645–666, 2020.

[32] Flood Sung, Yongxin Yang, L. Zhang, T. Xiang, P. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[33] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: A good embedding is all you need? In *ECCV*, pages 266–282, 2020.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[35] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

[36] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[37] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[38] Chen Xing, Negar Rostamzadeh, Boris Oreshkin, and Pedro O O. Pinheiro. Adaptive cross-modal few-shot learning. In *Advances in Neural Information Processing Systems*, 2019.

[39] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14393–14402, June 2021.