Learning to Hallucinate Examples from Extrinsic and Intrinsic Supervision

Liangke Gui[1*] Adrien Bardes[2*†] Ruslan Salakhutdinov[1]
Alexander Hauptmann[1] Martial Hebert[1] Yu-Xiong Wang[3]
[1] CMU [2] Facebook AI Research, Inria [3] UIUC
{liangkeg, rsalakhu, alex, hebert}@cs.cmu.edu adrien.bardes@inria.fr yxw@illinois.edu

Abstract

Learning to hallucinate additional examples has recently been shown as a promising direction to address few-shot learning tasks. This work investigates two important yet overlooked natural supervision signals for guiding the hallucination process – (i) extrinsic: classifiers trained on hallucinated examples should be close to strong classifiers that would be learned from a large amount of real examples; and (ii) intrinsic: clusters of hallucinated and real examples belonging to the same class should be pulled together, while simultaneously pushing apart clusters of hallucinated and real examples from different classes. We achieve (i) by introducing an additional mentor model on data-abundant base classes for directing the hallucinator, and achieve (ii) by performing contrastive learning between hallucinated and real examples. As a general, model-agnostic framework, our dual mentor- and self-directed (DMAS) hallucinator significantly improves few-shot learning performance on widely-used benchmarks in various scenarios.

1. Introduction

To alleviate the reliance on large, labeled datasets for learning deep models, few-shot learning has attracted increasing attention, with the goal of learning novel concepts from one, or only a few, annotated examples [20, 68, 71, 59, 21]. Existing work tries to solve this problem from the perspective of meta-learning [55, 7, 64], which is motivated by the human ability to leverage prior experiences when tackling a new task. Unlike the standard machine learning paradigm, where a model is trained on a set of examples, meta-learning is performed on a set of "simulated" tasks, each consisting of its own support and query sets [68]. The support set is used as the few-shot training data for the leaner, and the query set is used as the test data to evaluate the leaner's quality. By sampling small support and query sets from a large collection of labeled examples of base classes, meta-learning based approaches learn to extract task-agnostic knowledge, and apply it to a new few-shot learning task of novel classes.

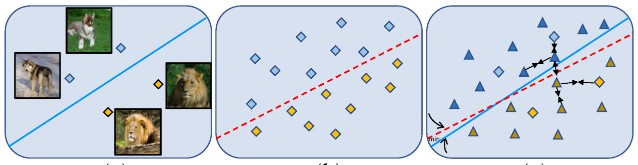


Figure 1: Learning a hallucinator to generate useful examples for few-shot learning through extrinsic and intrinsic supervision. During meta-training, we sample a few-shot task (*e.g.*, 2-way 2-shot classification) on **base classes** (Fig. 1a). **Extrinsic supervision:** The desired classifier for this task is the (dashed) one that would be learned from a large set of real examples (Fig. 1b). We *explicitly* introduce this strong classifier as "mentor" (abundant examples are available for base classes). We then learn the hallucinator in a way that minimizes the discrepancy between the (solid) "student" classifier (trained on hallucinated examples together with the few real examples) and the (dashed) mentor classifier (Fig. 1c). **Intrinsic supervision:** Through contrastive learning, clusters of hallucinated and real examples belonging to the same class are pulled together (→←), while simultaneously pushing apart (↔) clusters of hallucinated and real examples from different classes (Fig. 1c). During meta-testing, we use the *meta-trained, fixed* hallucinator to generate additional examples as augmentation for learning classifiers on **novel classes**. Real examples as light diamonds, hallucinated examples as dark triangles, and classifiers as solid or dashed lines.

One notable type of task-agnostic (or meta) knowledge comes from the shared mechanism of data augmentation or *hallucination* across categories [70, 22, 56, 84]. Since synthesizing raw images is often challenging or sometimes unnecessary, recent work has instead focused on hallucinating examples in a *learned feature space* [70, 22, 56, 84, 76, 83, 87]. This can be achieved by, for example, integrating a "hallucinator" module into a meta-learning framework, where it generates hallucinated examples guided by real ones from the support set [70]. The hallucinator captures the *intra-class variation* shared across categories, which generalizes to unseen classes. The learner then uses an augmented training set, which includes both the real and the hallucinated examples to

*equal contribution
†work done while visiting CMU

learn classifiers. The hallucinator is meta-trained end-to-end with the learner, through back-propagating a classification loss based on ground-truth labels of query data.

Despite the success of prior approaches, we argue that solely using the classification loss on the *small query set* as supervision is insufficient to adjust the hallucinator to produce effective samples in the few-shot regime. Therefore, the performance of the classifiers trained on hallucinated examples is still substantially inferior to that of the classifiers trained on real examples [16, 58]. To overcome this challenge, *our key insight* is that there are *two important yet under-explored natural signals* for guiding the data generation process – *extrinsic and intrinsic supervision*. This work explores how to leverage such supervision to enable hallucinating examples in a way that helps the classification algorithm learn better classifiers.

The first source of supervision is an *extrinsic signal from large-sample learning*. As illustrated in Figure 1, to be most helpful as a hallucinator, a classifier trained on the hallucinated examples (which are generated from a small support set of real samples) is expected to be *close* to a strong classifier that would be trained on a large amount of real examples . This extrinsic signal from large-sample learning is a natural source of supervision for few-shot learning, but it has been largely overlooked in prior work. While we have very little data on novel classes, we *do have a large number of real examples on base classes*. Therefore, on base classes we introduce a "mentor" model, which is a strong classifier pre-trained on all the available large amount of real examples. Correspondingly, the classifier trained on hallucinated examples along with few real support examples becomes the "student."

We now minimize the discrepancy between the student and mentor classifiers. A straightforward approach would be minimizing the distance between the two classifiers in the parameter space [71, 72, 11], which tends to be difficult and noisy due to the lack of suitable metrics. Hence, we instead encourage the output predictions from the student classifier (*e.g.*, the distribution of class probabilities) to be similar to those predicted by the mentor on the query set. This way of learning is reminiscent of knowledge distillation [29]. By doing so, the hallucinator explicitly learns how to produce examples that enable the student classifier to mimic the behavior of the mentor. Note that *the student-mentor pairs are only used for meta-training on base classes; there are no mentor classifiers for meta-testing on novel classes.*

In practice, the student and mentor classifiers could be quite different from each other at the beginning of the training, if the mentor is produced by a large amount of real examples while the student has access to only few real examples. To address this issue, we propose *a progressive guidance scheme* inspired by curriculum learning [8], and explore two *dual* directions – (1) we start with a mentor and a student, both trained on a small number of real examples, and we gradually *strengthen* the mentor by re-training it with increasing number of real examples; and (2) we start with a mentor and a student, both trained on a large number of real examples, and we gradually *weaken* the student by removing its real examples. During both of the processes, the hallucinator is also trained progressively.

The second source of supervision is the *intrinsic label consistency between hallucinated and real examples*. As illustrated in Figure 1, hallucinated and real examples belonging to the same class should be pulled together, while simultaneously pushing apart clusters of hallucinated and real examples from different classes. However, without appropriate constraints, the hallucinated examples might be noisy and spread over across class boundaries (*e.g.*, a hallucinated dog example resides within the cat cluster). To this end, we formulate the problem as *supervised contrastive learning*, inspired by recent progress on self-supervised learning [74, 28, 12, 30]. We treat hallucinated and real examples as different views of the data, and generate the positive and negative pairs correspondingly. For example, the positives are drawn from both hallucinated and real samples of the same class. Note that different from conventional contrastive learning that learns an embedding space (where the data augmentation is pre-defined), we use the contrastive loss to self-direct the hallucinated examples in the right class cluster or manifold (where is the feature space is pre-trained).

Our contributions are three-fold. (1) By jointly leveraging the *complementary* extrinsic and intrinsic supervision, we develop a general meta-learning with hallucination framework. (2) We not only extract shared knowledge across a collection of few-shot learning tasks, similar to most existing meta-learning methods, but also *progressively* exploit extrinsic knowledge in *large-sample models trained on base classes as mentor* to guide hallucination and few-shot learning. (3) Through a contrastive learning process, the hallucinated examples are self-directed to maintain the intrinsic label consistency with real examples. Our dual mentor- and self-directed (DMAS) hallucinator is *model-agnostic*, which can *generate data in different feature spaces and can be combined with different classification models* to consistently boost their few-shot learning performance on a variety of benchmarks, including ImageNet1K [27, 70], *mini*ImageNet [68, 49], *tiered*ImageNet [51], and CUB [69].

2. Related Work

Generative Models. Generative models have recently shown great potential as a way of data augmentation for few-shot learning [5, 70, 84, 22] and semi-supervised learning [16], but the improvement of recognition performance is still limited [58]. The generation can be performed either in image space [15] or in a pre-trained feature space [27], by using an auto-encoder architecture [56], GANlike generator [70], or the combination of GANs and auto-encoders [75, 76]. Our work is independent of these different types of generators, and we focus primarily on how to train the generator to improve its use for recognition tasks by leveraging large amounts of auxiliary data and self-supervision.

Few-Shot Learning and Meta-Learning. Meta-learning, or the ability to *learn to learn* [64], is a powerful framework for tackling the problem of learning with limited data. Most of modern approaches fall into one of the categories between optimization and metric learning based methods. Optimization based methods learn how to do fast adaptation to novel tasks, by learning appropriate parameter updates [49] or a general initialization [21]. Adaptation could be done in the original feature space [21, 4, 6] or in an embedded space [52]. Prior work on few-shot domain adaptation [53, 31] learns how to balance cross-domain clustering that is domain invariant. Metric learning methods focus on learning a similarity metric [33]. Several distance functions have been explored, from the Euclidean distance [59, 2] and the cosine distance [13, 23, 19] to more complex parametric functions and metrics [68, 62, 37, 82], or using an additional task-specific metric [46]. Most methods often treat each category separately without considering the relations between them. Graph neural networks are thus introduced to leverage those relations [54, 32, 24]. To conduct meta-learning more effectively, recent approaches often first compute a set of features of the images using a trained feature extractor network. Given that high-dimensional features have better modeling capacity but are computationally expensive to work with, each meta-learning task is then formulated as a convex optimization problem and solved in its low-dimensional dual space [9, 34]. Our hallucinator component is generic and can be integrated into different meta-learning methods.

Teacher-Student Networks. Learning a model under the guidance of a teacher or mentor model has been widely used for model compression. Compressing one cumbersome or several models into a smaller model is a classic idea [18, 10] and has been popularized by the distillation formulation in [29]. Recent work focuses on advanced techniques to guide the distillation process [42, 78, 1] and its applications to practical problems, such as object detection [77, 73] and distributed machine learning [3]. In addition, knowledge distillation has been extended to address other tasks, including multi-task learning [63] and continual learning [38, 57]. To the best of our knowledge, our work is the first to introduce a mentor network for learning recognition task oriented generative models. Importantly, different from existing work that addresses models of different capacity, we consider *models of the same capacity but trained on real or synthetic data*.

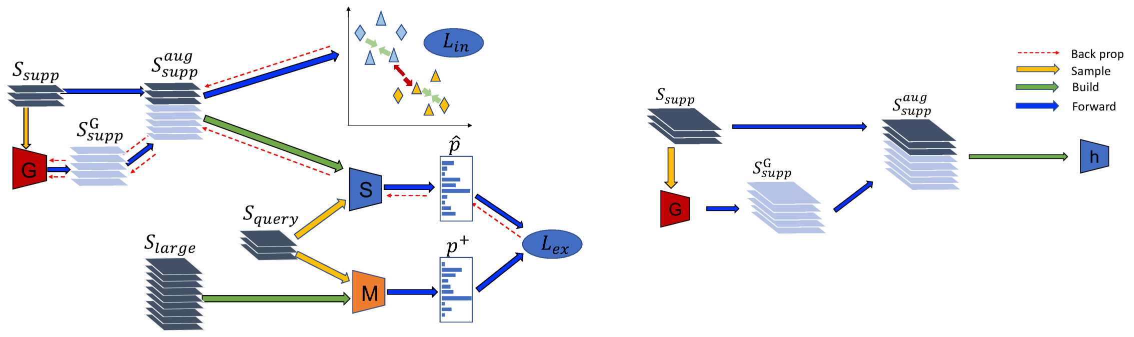
Contrastive Learning. Powerful self-supervised representation learning approaches have recently been developed in image domain via manually specified pretext tasks. Examples include auto-encoding methods which leverage contexts [47], channels [86], and colors [85] to recover the input under some corruption. Some pretext tasks form pseudo-labels by relative patch locations [17], image rotations [25], and jigsaw puzzles [44]. These pretext tasks are collected under the umbrella of the contrastive learning framework, which maintains the relative consistency between the representations of an image and its augmented views [45, 74, 80, 28, 12, 65, 26, 14, 81]. In our work, we treat hallucinated and real examples as different views of the data and use the contrastive loss to self-direct the hallucinated examples in the right class cluster or manifold.

3. Dual Mentor- and Self-Directed Hallucinator

Few-Shot Learning Setting. We are given a set of base categories $\mathcal{C}_{\text{base}}$ and a set of novel categories $\mathcal{C}_{\text{novel}}$, where $\mathcal{C}_{\text{base}} \cap \mathcal{C}_{\text{novel}} = \emptyset$. We have a base dataset $\mathcal{D}_{\text{base}}$ with a large amount of annotated training examples per class and a novel dataset $\mathcal{D}_{\text{novel}}$ with only few annotated training examples per class. Few-shot learning aims to learn a good classification model h for $\mathcal{C}_{\text{novel}}$ based on the small dataset $\mathcal{D}_{\text{novel}}$. Recent work achieves this through a meta-learning procedure [68], which learns from a collection of sampled few-shot classification tasks on $\mathcal{C}_{\text{base}}$. Given a set of categories $\mathcal{C}$ and a set of data $\mathcal{D}$, an m-way k-shot task is composed of a subset $\mathcal{C}_{\text{sub}}$ of m categories from $\mathcal{C}$, a support (training) set $\mathcal{S}_{\text{supp}}$ of k examples from $\mathcal{D}$ for each class in $\mathcal{C}_{\text{sub}}$, and a query (test) set $\mathcal{S}_{\text{query}}$ of one or few examples from $\mathcal{D}$ for each class in $\mathcal{C}_{\text{sub}}$. Meta-learning is performed in two phases as follows.

During *meta-training*, a classifier learns from a collection of m-way k-shot tasks sampled from $\mathcal{C}_{\text{base}}$ and $\mathcal{D}_{\text{base}}$. While our work is agnostic to different classification models, here we take a simple cosine classifier [13] as an example – a variant of prototypical networks [59] which uses the cosine instead of the standard Euclidean distance function. In each iteration, we compute a prototype representation for each class in $\mathcal{C}_{\text{sub}}$. Each example is fed to an embedding function f_θ with learnable parameters θ. The prototype of class c is the mean of the outputs through f_θ of examples from c in $\mathcal{S}_{\text{supp}}$. We then feed the examples in $\mathcal{S}_{\text{query}}$ to the classifier and update the parameters θ. During *meta-testing*, we use the same approach and build our previously meta-learned classifier with one unique m-way k-shot task, using $\mathcal{C}_{\text{novel}}$ instead of $\mathcal{C}_{\text{base}}$ and $\mathcal{D}_{\text{novel}}$ instead of $\mathcal{D}_{\text{base}}$. We evaluate the final classifier on unseen examples with labels from $\mathcal{C}_{\text{novel}}$.

Meta-Learning with Hallucination. Incorporating a generative model which produces additional examples for data augmentation has been shown to facilitate meta-learning [70, 22, 56]. While our approach does not rely on specific types of generative models, here we focus on the *feature hallucinator* in [70], due to its simplicity and state-of-the-art performance, which is implemented as a light-weight multi-layer perceptron (MLP) module. The hallucinator is a



(a) Meta-training on base classes (b) Meta-testing on novel classes

Figure 2: Overview of our dual mentor- and self-directed hallucinator "DMAS," learned through extrinsic and intrinsic supervision. During each iteration of meta-training, a small support set S_{supp} is augmented with a set of examples S^{G}_{supp} produced by a hallucinator G. Examples from the resulting set $S^{\text{aug}}_{\text{supp}}$ are used to build a student classifier model S. A mentor model M is built from a set S_{large} containing a large amount of real examples on *base classes*. The hallucinator G is trained by jointly leveraging the extrinsic supervision from $\mathcal{L}_{\text{ex}}$ and the intrinsic supervision from $\mathcal{L}_{\text{in}}$. That is, $\mathcal{L}_{\text{ex}}$ enforces the student classifier to mimic the distribution of class probabilities predicted by the mentor model on the query set S_{query}; meanwhile, $\mathcal{L}_{\text{in}}$ enforces the intrinsic label consistency between hallucinated and real examples from $S^{\text{aug}}_{\text{supp}}$ through contrastive learning. During meta-testing, we use the meta-trained, fixed G to generate additional examples as augmentation for learning classifiers h on novel classes; *there are no mentor classifiers*. Real examples as diamonds, hallucinated examples as triangles.

function $G(x, z; w) : \mathcal{R}^{d+d_{\text{noise}}} \to \mathcal{R}^{d}$ that produces examples in a pre-trained feature space of dimension d, where x is the feature vector of a real example, z is a random noise vector of dimension d_{noise} sampled from a Gaussian distribution, and w is the parameters of G. The hallucinated example $G(x, z; w)$ is of the same category as x.

Now the procedure of meta-learning integrated with the hallucinator G is illustrated in Figure 2. During each iteration of meta-training, the support set $\mathcal{S}_{\text{supp}}$ is first augmented by a generated set $\mathcal{S}^{G}_{\text{supp}}$. Specifically, for each class y, we sample $k^{\text{gen}}_{\text{train}}$ examples (x, y) in $\mathcal{S}_{\text{supp}}$, sample associated random noise vectors z, and then add (x', y) to $\mathcal{S}^{G}_{\text{supp}}$, where $x' = G(x, z; w)$. Our final support training set is $\mathcal{S}^{\text{aug}}_{\text{supp}} = \mathcal{S}_{\text{supp}} \cup \mathcal{S}^{G}_{\text{supp}}$. As long as G is differentiable with respect to the generated set $\mathcal{S}^{G}_{\text{supp}}$, the gradients of the final classification loss on $\mathcal{S}_{\text{query}}$ can be back-propagated into G to produce useful hallucinated examples. Through meta-training over a large amount of iterations, the hallucinator learns to capture shared modes of variation across different classes and can thus generalize to unseen classes. During meta-testing, we use the learned G to generate additional examples for recognizing categories in $\mathcal{C}_{\text{novel}}$.

Hallucination with Extrinsic Guidance from Mentor. The end-to-end optimization of the classification loss enables the hallucinator to produce useful examples in the few-shot regime. However, since the classification loss is computed on *small query sets*, such supervision solely is insufficient to adjust the hallucinator to produce *discriminative* examples that most contribute to formulating classifier decision boundaries. Hence, the resulting classifier trained on the hallucinated examples could be still far away from the desired classifier that would be learned from a large set of real examples. This makes it critical to close the gap between these two classifiers. In fact, during meta-training, a large amount of annotated examples are already available for the base categories $\mathcal{C}_{\text{base}}$, which allows us to explicitly obtain the classifier trained on a large set of examples and use it to guide the learning of the hallucinator.

Formally, we treat the classifier trained on the augmented set of the hallucinated examples and the few support examples as a *student model*, and we treat the classifier trained on a large set of real base examples as a *mentor model*. Our goal then is to learn the hallucinator by minimizing the discrepancy between the student classifier and its mentor model. While a naïve approach would be to directly characterize the difference between their model parameters, it turns out to be challenging due to the high dimensionality of the parameter space. Inspired by the teacher-student network [29], we instead enforce the student to mimic the distribution of class probabilities predicted by the mentor network, which can be viewed as a way of regularization to improve the generalization performance of the student model [43].

As shown in Figure 2, meta-training the hallucinator G is conducted in the following way. We first sample a *large* set of examples $\mathcal{S}_{\text{large}}$ with k_{large} examples per class in $\mathcal{C}_{\text{base}}$ and train a mentor classifier using all the examples in $\mathcal{S}_{\text{large}}$. During each iteration of meta-training, we augment $\mathcal{S}_{\text{supp}}$ by generating new examples using the hallucinator G. We train the student classifier on $\mathcal{S}^{\text{aug}}_{\text{supp}}$ through the knowledge distillation loss function in [29]:

$$\mathcal{L}_{\text{ex}}(s, m, y) = \mathcal{L}_{\text{CE}}(\sigma(s), e_y) + \alpha\tau_1^2 \mathcal{L}_{\text{CE}}(\sigma(\frac{s}{\tau_1}), \sigma(\frac{m}{\tau_1})), \quad (1)$$

which consists of a standard cross-entropy loss (the first term) and an additional component that measures the difference between student and mentor outputs (the second term). s and m are the logits produced by the student and the mentor, respectively, for a test example of label y in $\mathcal{S}_{\text{query}}$. σ denotes the softmax function, $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss, e_y is the one-hot encoding of y, and α is a trade-off hyper-parameter that balances the two terms. Note that $\tau_1 > 0$ is a critical *learnable* parameter called *temperature*, which smooths the probability distribution produced by the mentor and makes the corresponding decision boundary easier to learn for the student than the original one.

Self-Directed Learning with Intrinsic Label Consistency. While the hallucinated examples directed by the extrinsic mentor are useful, without other constraints they might spread over across class boundaries and thus be noisy. Inspired by supervised contrastive learning [30], we enforce intrinsic label consistency between hallucinated examples and real examples.

Formally, suppose we sample N real examples per mini-batch and generate M hallucinated examples, resulting in a batch $\mathcal{I}$ of $M+N$ examples. Given an anchor example x_i, $P(i)$ is the set of indices of all positives in the batch distinct from i and $A(i) \equiv I \backslash \{i\}$. The supervised contrastive loss is defined as $\mathcal{L}_{\text{in}} = \frac{1}{M+N} \sum_{i=1}^{M+N} \mathcal{L}_i$ and

$$\mathcal{L}_i = -\frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{exp(x_i \cdot x_p / \tau_2)}{\sum_{a \in A(i)} exp(x_i \cdot x_a / \tau_2)}, \quad (2)$$

where $\tau_2 > 0$ is a temperature parameter and $|P(i)|$ is its cardinality. This loss allows the real and hallucinated examples from the same classes to attract mutually, while they repel the other examples from different classes in the mini-batch.

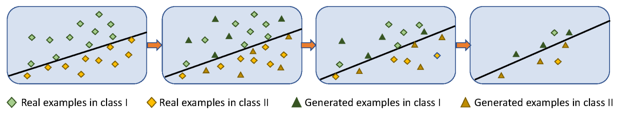
Thus, our dual mentor- and self-directed hallucinator can be derived from Eqn. 1 and Eqn. 2 as

$$\mathcal{L} = \mathcal{L}_{\text{ex}} + \beta \mathcal{L}_{\text{in}}, \quad (3)$$

where β is a trade-off hyper-parameter that balances the two terms. Minimizing Eqn. 3 over $\mathcal{S}_{\text{query}}$ thus guides the hallucinator towards producing useful examples that help the student classifier recover the decision boundary from the mentor model.

4. Progressive Guidance from Mentor Model

Under the framework of meta-learning with extrinsic guidance, a straightforward way is to build the mentor model by using k_{large} as large as possible (potentially the full set of $\mathcal{D}_{\text{base}}$) and keep it fixed, and to train the hallucinator and student classifier using only few real examples. By doing so, however, we face the problem that the decision boundaries obtained by those two models could be very far from each other at the beginning of the training, making the learning of the hallucinator difficult. To address this issue, we perform the learning process in a progressive manner with *varied* number of real examples. We start with a mentor and a student which have access to a *not too different* number of real examples, and then progressively change the number of examples, so that the decision boundaries transform in a smooth manner. Concretely, this can be achieved in the following two *dual* directions.


Real examples in class I
Real examples in class II
Generated examples in class I
Generated examples in class II
Figure 3: Illustration of progressive guidance by weakening the student classifier in the case of recognizing two classes. We start with a large number of real examples for both the student and the mentor, and learn the corresponding mentor model (the leftmost image). We then *gradually remove* the real examples for the student over the training. The hallucinator learns to generate additional examples based on the remaining real examples to *preserve* the mentor decision boundary (the middle two and rightmost images).

Progressive Guidance by Strengthening the Mentor. In this setting, both the student and the mentor start with a small number of real examples. However, the number of real examples for the mentor *gradually increases over the training*. The objective for the hallucinator then is to learn to generate additional examples so that its corresponding student can always *match* the performance of the mentor, whenever the mentor is re-trained with more samples and becomes stronger. More specifically, during meta-training, the support set $\mathcal{S}_{\text{supp}}$ of each few-shot task is composed of very few examples per class, k_{train}, as in regular meta-training. At the beginning, we sample $\mathcal{S}_{\text{large}}$, with k_{large} being set to the value of k_{train}. We then progressively sample new real examples in the same amount for each class and add them into $\mathcal{S}_{\text{large}}$. k_{large} grows from k_{train} to k_{max} in a linear or logarithmic scale, where k_{max} is the maximum available number of examples per class in $\mathcal{D}_{\text{base}}$. We re-train the mentor model every time we add new examples.

Progressive Guidance by Weakening the Student. In this setting, both the student and the mentor start with a large number of real examples. However, we *gradually remove* the real examples for the student over the training. The objective for the hallucinator then is to learn to generate the missing examples based on the remaining real examples. This allows the student to *preserve or stabilize* the original decision boundary formulated by the large set of examples (*i.e.*, the mentor boundary), when the student has access to less real examples and becomes weaker. More specifically, during meta-training, the support set $\mathcal{S}_{\text{supp}}$ of each "few-shot" task is composed of a large number of examples per class, unlike regular meta-training. This number of examples per class in $\mathcal{S}_{\text{supp}}$, k_{train}, decreases in a linear or logarithmic scale, until it reaches a small value.

5. Evaluation

We now present experiments to evaluate our dual mentor- and self-directed (DMAS) hallucinator on few-shot classification, and study the effect of progressive guidance from extrinsic and intrinsic supervision. Since DMAS is agnos-

Method	Backbone	*mini*ImageNet		*tiered*ImageNet	
		k=1	5	k=1	5
Cosine Classifier [13]	ResNet12	55.43 ± 0.81	77.18 ± 0.61	61.49 ± 0.91	82.37 ± 0.67
TADAM [46]	ResNet12	58.50 ± 0.30	76.70 ± 0.30	–	–
ECM [50]	ResNet12	$59.00 \pm$ –	$77.46 \pm$ –	$63.99 \pm$ –	$81.97 \pm$ –
TPN [40]	ResNet12	$59.46 \pm$ –	$75.65 \pm$ –	59.91 ± 0.94	73.30 ± 0.75
PPA [48]	WRN-28-10	59.60 ± 0.41	73.74 ± 0.19	–	–
ProtoNet [59]	ResNet12	60.37 ± 0.83	78.02 ± 0.57	65.65 ± 0.92	83.40 ± 0.65
wDAE-GNN [24]	WRN-28-10	61.07 ± 0.15	76.75 ± 0.11	68.18 ± 0.16	83.09 ± 0.12
MTL [61]	ResNet12	61.20 ± 1.80	75.50 ± 0.80	–	–
LEO [52]	WRN-28-10	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.05	81.44 ± 0.09
DC [39]	ResNet12	62.53 ± 0.19	79.77 ± 0.19	–	–
MetaOptNet [34]	ResNet12	62.64 ± 0.82	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
FEAT [79]	ResNet24	62.96 ± 0.20	78.49 ± 0.15	–	–
MatchingNet [68]	ResNet12	63.08 ± 0.80	75.99 ± 0.60	68.50 ± 0.92	80.60 ± 0.71
CTM [35]	ResNet18	64.12 ± 0.82	80.51 ± 0.13	68.41 ± 0.39	84.28 ± 1.73
RFS [66]	ResNet12	64.82 ± 0.60	82.14 ± 0.43	71.52 ± 0.69	86.03 ± 0.49
DeepEMD [82]	ResNet12	65.91 ± 0.82	82.41 ± 0.56	71.16 ± 0.87	86.03 ± 0.58
DMAS (Ours)	ResNet12	$\mathbf{67.42 \pm 0.28}$	$\mathbf{83.74 \pm 0.20}$	$\mathbf{73.54 \pm 0.73}$	$\mathbf{86.27 \pm 0.47}$

(a) Test accuracy (%) on the novel classes for *mini*ImageNet and *tiered*ImageNet. '±' indicates 95% confidence intervals over tasks.

Method	Backbone	k=1	5
ProtoNet [59]	ResNet12	66.09 ± 0.92	82.50 ± 0.58
RelationNet [13, 62]	ResNet34	66.20 ± 0.99	82.30 ± 0.58
DEML [88]	ResNet50	66.95 ± 1.06	77.11 ± 0.78
MAML [13]	ResNet34	67.28 ± 1.08	83.47 ± 0.59
Cosine Classifier [13]	ResNet12	67.30 ± 0.86	84.75 ± 0.60
MatchingNet [68]	ResNet12	71.87 ± 0.85	85.08 ± 0.57
DeepEMD [82]	ResNet12	75.65 ± 0.83	88.69 ± 0.50
DMAS (Ours)	ResNet12	$\mathbf{78.47 \pm 0.62}$	$\mathbf{90.67 \pm 0.39}$

(b) Test accuracy (%) on the novel classes for CUB. '±' indicates 95% confidence intervals over tasks.

Method	Backbone	k=1	2	5	10
ProtoNet [59]	ResNet10	39.3	54.4	66.3	71.2
ProtoNet *Gen* [70]	ResNet10	45.0	55.9	67.3	73.0
MatchingNet [68]	ResNet10	43.6	54.0	66.0	72.5
Logistic regression [27]	ResNet10	38.4	51.1	64.8	71.6
Logistic regression *Analogies* [27]	ResNet10	40.7	50.8	62.0	69.3
Prototype Matching Net *Gen* [70]	ResNet10	45.8	57.8	69.0	74.3
Cosine Att. Weight [23]	ResNet10	46.0	57.5	69.1	74.8
DMAS (Ours)	ResNet10	**46.5**	**58.3**	**69.7**	**75.1**

(c) Top-5 accuracy (%) for **311-way** novel-class classification on ImageNet1K. **The 95% confidence intervals for all number are of the order of 0.2%.**

Table 1: Comparisons with state of the art on four widely-benchmarked few-shot classification datasets. *With simple cosine classifiers*, our DMAS significantly and consistently outperforms all the baselines (including sophisticated classification models) across the board.

tic to the choice of classification models, we validate its generalizability to different types of features and various meta-learning models. In particular, we focus on simple cosine classifiers, which have been recently shown to achieve very competitive few-shot performance [13].

Datasets. We evaluate on four widely-used datasets: (1) *mini*ImageNet [68, 49], with 64, 16, and 20 classes for meta-training, meta-validation, and meta-testing, respectively; (2) *tiered*ImageNet [51], with 20, 6, and 8 super-classes for meta-training, meta-validation, and meta-testing, respectively; (3) ImageNet1K [27, 70], with 193 base and 300 novel classes for cross-validation and 196 base and 311 novel classes for evaluation; (4) Caltech-UCSD Birds-200-2011 (CUB) [69, 79], with 100, 50, and 50 classes for meta-training, meta-validation, and meta-testing, respectively.

Implementation Details. For a fair comparison with previous work, we employ ResNet10 as our model backbone for ImageNet1K [70] and ResNet12 as our model backbone for the other three datasets [82]. As is commonly implemented in the state-of-the-art work, we follow the feature pre-training step [82]. We first train a convolutional network based feature extractor on the base classes. Then we extract and save these features to disk, and use these pre-computed features as inputs for meta-learning. We follow the feature hallucinator architecture in [70] and use a three layer MLP with ReLU as the activation. The embedding function f_θ of our cosine classifier is a two layer MLP.

During progressive guidance by weakening the student, we start training the mentor with $k_{\text{large}} = 256$, and we decrease the number to 1 in a logarithmic scale over $12,000$ iterations. We initialize the learnable parameters including the temperature τ_1 to 7, the scale factor of the cosine distance to 75, and the temperature τ_2 to 0.07. As the performance is not sensitive to trade-off hyper-parameters α and β, we empirically set them to 5 and 1, respectively. The number of hallucinated examples is a hype-parameter ranging from $2-10$. The saturation point of hallucinated examples on improving performance is typically 6. For ImageNet1K, we follow the settings in [70] and average over 5 pre-determined k-shot (*i.e.*, $k = 1, 2, 5, 10$) tasks. We report the mean top-5 accuracy and the 95% confidence intervals for all number are of the order of 0.2%. For the other datasets, we average over $1,000$ randomly sampled tasks and report the accuracies and the 95% confidence intervals.

Comparisons with State of the Art. We compare our model with the state-of-the-art methods. We report 5-way 1-shot and 5-way 5-shot performance on three benchmarks: *mini*ImageNet, *tiered*ImageNet, and CUB, and 311-way k-shot on ImageNet1K. The results are summarized in Table 1. Under the same backbones, our model consistently achieves the best performance on all the datasets and across different sample-size regimes, even outperforming sophisticated methods, such as the attention based classifier 'Cosine Att. Weight' [23] and DeepEMD [82]. In particular, our 1-shot model outperforms state-of-the-art methods by significant margins, *e.g.*, 1.5% on *mini*ImageNet, 2% on *tiered*ImageNet, and 2.8% on CUB.

Ablation Analysis. To unpack the performance gain and understand the impact of different components, we perform a series of ablations on the challenging ImageNet1K dataset. Tables 2 summarizes the top-5 accuracies and the 95% confidence intervals for all number are of the order of 0.2%.

Robust to different types of pre-trained features and classifiers. Table 2 shows that DMAS can effectively hallucinate

Method	Feature	k=1	2	5	10
ProtoNet [59] (baseline)	Standard	39.3	54.4	66.3	71.2
ProtoNet *w/ aug* (baseline)	Standard	40.2	55.0	66.7	71.6
ProtoNet *Gen* [70] (baseline)	Standard	45.0	55.9	67.3	73.0
ProtoNet *DMAS w/ ex*	Standard	45.1	55.5	67.3	73.3
ProtoNet *DMAS (full)*	Standard	**45.9**	**56.5**	**68.2**	**73.9**
Cosine Classifier (baseline)	Standard	37.8	51.0	65.5	72.5
Cosine Classifier *Gen* (baseline)	Standard	42.6	53.9	66.4	72.6
Cosine Classifier *DMAS w/ in*	Standard	43.4	54.7	67.1	73.5
Cosine Classifier *DMAS w/ ex*	Standard	44.5	56.2	68.6	74.2
Cosine Classifier *DMAS w/ ex*↑	Standard	44.3	56.3	68.8	74.2
Cosine Classifier *DMAS w/ ex*↓	Standard	45.4	56.7	68.8	74.8
Cosine Classifier *DMAS (full)*	Standard	**46.5**	**58.3**	**69.7**	**75.1**
Cosine Classifier (baseline)	Cosine	45.8	57.0	68.9	74.3
Cosine Classifier *Gen* (baseline)	Cosine	47.0	57.8	69.1	74.3
Cosine Classifier *DMAS w/ ex*	Cosine	47.2	58.2	69.2	74.4
Cosine Classifier *DMAS (full)*	Cosine	**47.9**	**59.3**	**70.1**	**75.5**

Table 2: Ablation studies (top-5 accuracy) on ImageNet1K **311-way** classification: (1) **different pre-trained feature spaces** for hallucination – 'standard' (the feature backbone is a ResNet10 pre-trained using a standard cross-entropy linear classifier on base classes) vs. 'cosine' (the ResNet10 feature backbone is pre-trained using a cosine classifier); (2) **different types of classifiers** – prototypical net vs. cosine classifier; (3) **impact of different sources of supervision and progressive training**. *w/ aug*: with standard data augmentation. *Gen*: with a plain hallucinator [70] trained using the classification loss on the query set solely. *DMAS w/ ex*: DMAS trained only under the guidance of the mentor *without progressive training*. *DMAS w/ ex*↑: progressive guidance through *strengthening the mentor*. *DMAS w/ ex*↓: progressive guidance through *weakening the student*. *DMAS w/ in*: DMAS trained only in a self-directed way through contrastive learning. *DMAS (full)*: trained under both (progressively) extrinsic and intrinsic supervision.

Method	k=1	5
ProtoNet [59]	50.01 ± 0.82	72.02 ± 0.67
MatchingNet [68]	51.65 ± 0.84	69.14 ± 0.72
Cosine Classifier [13]	44.17 ± 0.78	69.01 ± 0.74
Linear Classifier [13]	50.37 ± 0.79	73.30 ± 0.69
KNN [36]	50.84 ± 0.81	71.25 ± 0.69
DeepEMD [82]	54.24 ± 0.86	78.86 ± 0.65
DMAS (Ours)	**63.72 ± 0.29**	**81.24 ± 0.20**

Table 3: Cross-domain evaluation (*mini*ImageNet → CUB). Our model outperforms other baseline methods by large margins, showing the generalization of our learned hallucinator.

data in different types of pre-trained feature spaces and can work with different types of classifiers. Notably, DMAS achieves the best performance in a *homogeneous* setting, where the feature is pre-trained by using a cosine classifier and the final classification model is also a cosine classifier.

Extrinsic guidance from mentor. From Table 2, we can observe that DMAS significantly outperforms baselines by benefiting from the extrinsic guidance of the mentor. There are 5.8% improvement when combining with the prototypical network and 6.7% improvement when combining with the cosine classifier. More importantly, DMAS outperforms the plain hallucinator [70] which is trained using the classification loss only. Note that both the baselines and DMAS use the same amount of data for meta-training on base classes.

Intrinsic supervision. Table 2 also shows that DMAS trained only with the intrinsic supervision already outperforms the baselines. The improvement is more pronounced when there are very few examples, *e.g.*, 5.6% improvement when $k = 1$. This implies the importance of preserving the label consistency between hallucinated and real examples. In addition, the full DMAS model achieves the best performance, demonstrating that *the extrinsic supervision and the intrinsic supervision are complementary to each other.*

Method	Backbone	k=1	5
MetaOptNet [34]	ResNet12	62.64 ± 0.61	78.63 ± 0.46
MetaOptNet + *Gen* [70]	ResNet12	63.46 ± 0.43	80.02 ± 0.28
MetaOptNet + DMAS (Ours)	ResNet12	**64.55 ± 0.64**	**80.42 ± 0.46**
S2M2 [41]	WRN-28-10	63.90 ± 0.18	81.03 ± 0.11
S2M2 + *Gen* [70]	WRN-28-10	63.37 ± 0.56	81.23 ± 0.19
S2M2 + DMAS (Ours)	WRN-28-10	**65.35 ± 0.63**	**83.55 ± 0.41**
DeepEMD [82]	ResNet12	65.91 ± 0.82	82.41 ± 0.56
DeepEMD + *Gen* [70]	ResNet12	64.73 ± 0.30	79.92 ± 0.21
DeepEMD + DMAS (Ours)	ResNet12	**67.42 ± 0.28**	**83.74 ± 0.20**

Table 4: Ablation study on the generalizability of our approach and additional comparisons with state of the art on *mini*ImageNet. Our DMAS hallucinator is **general and can work with different types of classification models and different backbone models** to *consistently* improve their performance. In addition, DMAS consistently outperforms the plain hallucinator [70].

Strengthening the mentor vs. weakening the student. We compare two directions for progressive guidance by strengthening the mentor (*w/ ex*↑) and weakening the student (*w/ ex*↓). We use a logarithmic scale when changing the number of examples on which the student or mentor model is trained [60, 72]. As shown in Table 2, both directions outperform the normal guidance without progression (*w/ ex*), and weakening the student achieves better results. It comes from the fact that, if both mentor and student start being weak, the learning problem could actually be hard due to the high variance of both mentor and student.

Comparisons with standard data augmentation. Table 2 shows that our learned data hallucination outperforms meta-learning with standard hand-crafted data augmentation ('w/ aug'), which includes random crop, random horizontal flip, and color jittering as in [13], indicating the importance of exploiting shared intra-class variation.

Cross-Domain Evaluation. So far, we have focused on the within-domain scenario. Now we consider the cross-domain scenario, which allows us to investigate the generalization of our DMAS hallucinator and understand the effects of domain shifts. Following the cross-domain setup in [13, 82], the experiment in Table 3 shows that our DMAS hallucinator trained on *mini*ImageNet is effective for never-before-seen classes on CUB *without any fine-tuning*.

DMAS as a General Plug-and-Play Module. Table 4 further shows the generalizability of our approach – the DMAS hallucinator can work with different types of classification models and different backbone models to consistently improve their performance. To fully investigate the impact of DMAS and for a fair comparison, we conduct experiments on *mini*ImageNet with the same training setups (*e.g.*, backbones, data augmentation techniques, and training

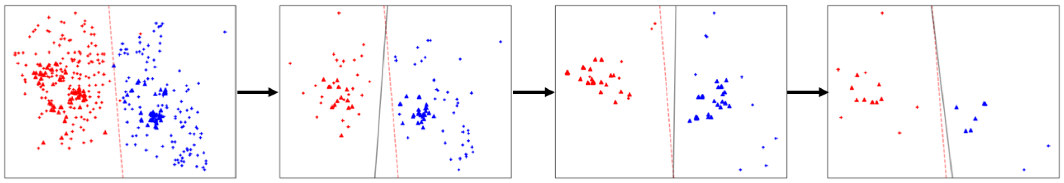


Figure 4: Visualization with t-SNE of the evolution of the decision boundary for two *novel* classes, when meta-training our DMAS hallucinator through progressive guidance by weakening the student. Real examples (small dots) are progressively removed, and hallucinated examples (triangles) are generated in a way that helps maintain the student decision boundary (black solid line) as close as possible to the desired decision boundary that would be formulated by a large set of real examples (red dashed line). We observe that PCA visualization has a similar phenomenon. **Best viewed in color with zoom.**

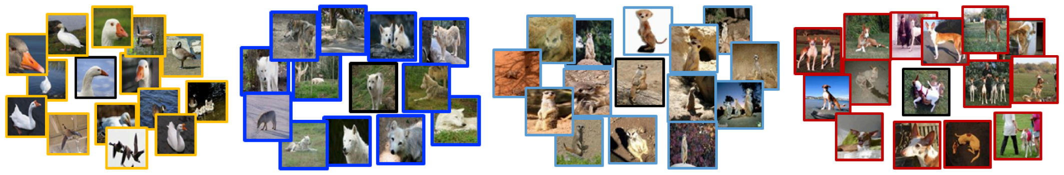


Figure 5: Visualization of nearest neighbor real images of hallucinated examples for four *novel* classes. For each class, the single black framed image comes from the original dataset and is used as a seed for generating new examples. Color framed images correspond to the nearest neighbor real images of the hallucinated examples in the feature space. **Best viewed in color with zoom.**

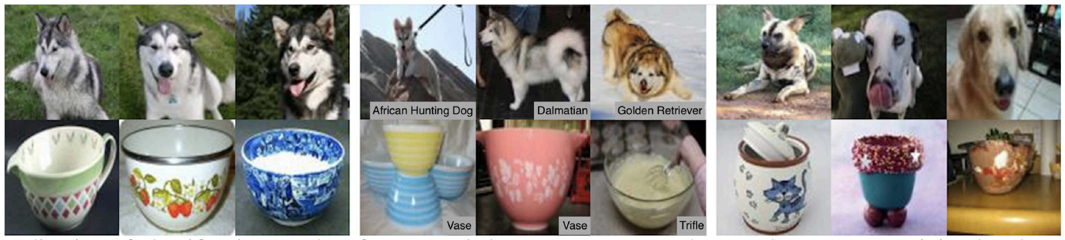


Figure 6: Visualization of classification results of two *novel* classes (Top row: malamute; bottom row: mixing bowl) and comparison between our DMAS hallucinator and the plain hallucinator [70]. The left block shows images correctly classified by both approaches. The middle block shows images that are misclassified by [70] as other classes (with predicted class names overlaid on the images), but correctly classified by our approach. The right block shows images from other classes that are misclassified by [70] as the target class, but correctly classified by our approach. In these examples, our classifier is able to recognize objects with different poses and view points, whereas [70] fails to distinguish between similar classes.

strategies) as the state-of-the-art approaches [34, 41, 82]. In all cases, DMAS can be seamlessly incorporated into these approaches (denoted as '+'), and substantially improve their performance, *e.g.*, 1.9% improvement when combining with MetaOptNet [34] and 1.5% improvement when combining with S2M2 [41] under the challenging 1-shot setting.

Comparisons with the plain hallucinator. Table 4 also shows that DMAS *consistently* outperforms the plain hallucinator [70] for different types of models (Table 2 has already shown this for ProtoNet and cosine classifier). More importantly, 'DeepEMD + [70]' is *worse* than the plain DeepEMD; a similar phenomenon is observed with S2M2 in the 1-shot case. These results suggest that, *while DMAS is general, [70] is not a general module for different few-shot models*. For more sophisticated models (S2M2 and DeepEMD), solely using the classification loss as in [70] is insufficient to adjust the hallucinator to produce effective samples. This further verifies the importance of extrinsic and intrinsic supervision.

Visualizations. To further understand how our model helps learning a classifier and refining the hallucinator, we conduct visualizations on ImageNet1K. We first visualize in Figure 4 the evolution of the decision boundary for two *novel* classes during progressive guidance by weakening the student using t-SNE [67]. We then visualize in Figure 5 the hallucinated examples in the pixel space, using their nearest neighbor real images in the feature space. Finally in Figure 6, we compare our approach with the state-of-the-art meta-learned hallucinator [70] and show that ours is able to recognize a large range of visual variations.

6. Conclusion

We present an approach to few-shot classification that uses a dual mentor- and self-directed hallucinator to generate additional examples. This is achieved by exploiting two important natural supervision signals that facilitate data hallucination in a way that most improves the classification performance, and is trained end-to-end through meta-learning. Our hallucinator can be inserted as a plug-and-play module into different classification models. The extensive experiments demonstrate our state-of-the-art performance on the widely-benchmarked few-shot datasets in various scenarios.

Acknowledgement: This work was supported in part by ONR MURI N000014-16-1-2007, AFRL Grant FA23861714660, and NSF Grant 2106825. We also thank AWS Cloud Credits for Research program.

References

[1] Angeline Aguinaldo, Ping-Yeh Chiang, Alex Gain, Ameya Patil, Kolten Pearson, and Soheil Feizi. Compressing GANs using knowledge distillation. *arXiv preprint arXiv:1902.00159*, 2019. 3

[2] Kelsey R Allen, Evan Shelhamer, Hanul Shin, and Joshua B Tenenbaum. Infinite mixture prototypes for few-shot learning. In *ICML*, 2019. 3

[3] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Róbert Ormándi, George E. Dahl, and Geoffrey E. Hinton. Large scale distributed neural network training through online distillation. In *ICLR*, 2018. 3

[4] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. In *ICLR*, 2019. 3

[5] Anthreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. In *ICLR*, 2017. 2

[6] Antreas Antoniou and Amos J. Storkey. Learning to learn via self-critique. In *NeurIPS*, 2019. 3

[7] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Université de Montréal, Département d'informatique et de recherche opérationnelle, 1990. 1

[8] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009. 2

[9] Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2018. 3

[10] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, 2006. 3

[11] Wei-Lun Chao, Han-Jia Ye, De-Chuan Zhan, Mark Campbell, and Kilian Q Weinberger. Revisiting meta-learning as supervised learning. *arXiv preprint arXiv:2002.00573*, 2020. 2

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 3

[13] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. 3, 6, 7

[14] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 3

[15] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *CVPR*, 2019. 2

[16] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan Salakhutdinov. Good semi-supervised learning that requires a bad GAN. In *NeurIPS*, 2017. 2

[17] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 3

[18] Pedro Domingos. Knowledge acquisition from examples via multiple models. In *ICML*, 1997. 3

[19] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *ICCV*, 2019. 3

[20] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *TPAMI*, 28(4):594–611, 2006. 1

[21] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 1, 3

[22] Hang Gao, Zheng Shou, Alireza, Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. In *NeurIPS*, 2018. 1, 2, 3

[23] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 3, 6

[24] Spyros Gidaris and Nikos Komodakis. Generating classification weights with GNN denoising autoencoders for few-shot learning. In *CVPR*, 2019. 3, 6

[25] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 3

[26] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 3

[27] Bharath Hariharan and Ross B. Girshick. Low-shot visual object recognition by shrinking and hallucinating features. In *ICCV*, 2017. 2, 6

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 3

[29] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 3, 4

[30] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020. 2, 5

[31] Donghyun Kim, Kuniaki Saito, Tae-Hyun Oh, Bryan A Plummer, Stan Sclaroff, and Kate Saenko. CDSP: Cross-domain self-supervised pre-training. In *ICCV*, 2021. 3

[32] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. Edge-labeling graph neural network for few-shot learning. In *CVPR*, 2019. 3

[33] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML*, 2015. 3

[34] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019. 3, 6, 7, 8

[35] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *CVPR*, 2019. 6

[36] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *CVPR*, 2019. 7

[37] Wenbin Li, Jinglin Xu, Jing Huo, Lei Wang, Yang Gao, and Jiebo Luo. Distribution consistency based covariance metric networks for few-shot learning. In *AAAI*, 2019. 3

[38] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 40(12):2935–2947, 2017. 3

[39] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. In *CVPR*, 2019. 6

[40] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho

Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR*, 2019. 6

[41] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *WACV*, 2020. 7, 8

[42] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *AAAI*, 2020. 3

[43] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *TPAMI*, 41(8):1979–1993, 2018. 4

[44] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 3

[45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[46] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018. 3, 6

[47] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 3

[48] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *CVPR*, 2018. 6

[49] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2, 3, 6

[50] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *ICCV*, 2019. 6

[51] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018. 2, 6

[52] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019. 3, 6

[53] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. In *NeurIPS*, 2020. 3

[54] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *ICLR*, 2018. 3

[55] Jürgen Schmidhuber. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook. Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*, 1987. 1

[56] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex M. Bronstein. Delta-encoder: An effective sample synthesis method for few-shot object recognition. In *NeurIPS*, 2018. 1, 2, 3

[57] Jonathan Schwarz, Jelena Luketina, Wojciech M Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *ICML*, 2018. 3

[58] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my GAN? In *ECCV*, 2018. 2

[59] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 1, 3, 6, 7

[60] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 7

[61] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, 2019. 6

[62] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 3, 6

[63] Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *NeurIPS*, 2017. 3

[64] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012. 1, 3

[65] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 3

[66] 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*, 2020. 6

[67] Laurens van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-SNE. *JMLR*, 9(11):2579–2605, 2008. 8

[68] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016. 1, 2, 3, 6, 7

[69] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical report, 2011. 2, 6

[70] Yu-Xiong Wang, Ross B. Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, 2018. 1, 2, 3, 6, 7, 8

[71] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016. 1, 2

[72] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*, 2017. 2, 7

[73] Yi Wei, Xinyu Pan, Hongwei Qin, and Junjie Yan. Quantization mimic: Towards very tiny CNN for object detection. In *ECCV*, 2018. 3

[74] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 2, 3

[75] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018. 3

[76] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-VAEGAN-D2: A feature generating framework for any-shot learning. In *CVPR*, 2019. 1, 3

[77] Jiaolong Xu, Peng Wang, Heng Yang, and Antonio M. López. Training a binary weight object detector by knowledge transfer for autonomous driving. In *ICRA*, 2019. 3

[78] Zheng Xu, Yen-Chang Hsu, and Jiawei Huang. Training shallow and thin networks for acceleration via knowledge distillation with conditional adversarial networks. In *ICLR*, 2018. 3

[79] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*, 2020. 6

[80] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, 2019. 3

[81] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. 2021. 3

[82] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, 2020. 3, 6, 7, 8

[83] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. In *CVPR*, 2019. 1

[84] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. MetaGAN: An adversarial approach to few-shot learning. In *NeurIPS*, 2018. 1, 2

[85] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 3

[86] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017. 3

[87] Weilin Zhang and Yu-Xiong Wang. Hallucination improves few-shot object detection. In *CVPR*, 2021. 1

[88] Fengwei Zhou, Bin Wu, and Zhenguo Li. Deep meta-learning: Learning to learn in the concept space. *arXiv preprint arXiv:1802.03596*, 2018. 6