Deep Learning under Privileged Information Using Heteroscedastic Dropout

John Lambert[*], Ozan Sener[*], Silvio Savarese
Department of Computer Science
Stanford University
`johnwl,osener,ssilvio@stanford.edu`

Abstract

Unlike machines, humans learn through rapid, abstract model-building. The role of a teacher is not simply to hammer home right or wrong answers, but rather to provide intuitive comments, comparisons, and explanations to a pupil. This is what the Learning Under Privileged Information (LUPI) paradigm endeavors to model by utilizing extra knowledge only available during training. We propose a new LUPI algorithm specifically designed for Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). We propose to use a heteroscedastic dropout (i.e. dropout with a varying variance) and make the variance of the dropout a function of privileged information. Intuitively, this corresponds to using the privileged information to control the uncertainty of the model output. We perform experiments using CNNs and RNNs for the tasks of image classification and machine translation. Our method significantly increases the sample efficiency during learning, resulting in higher accuracy with a large margin when the number of training examples is limited. We also theoretically justify the gains in sample efficiency by providing a generalization error bound decreasing with $\mathcal{O}(\frac{1}{n})$, where n is the number of training examples, in an oracle case.

1. Introduction

> *"Better than a thousand days of diligent study is one day with a great teacher."*
>
> — Japanese Proverb

It is a common belief that human students require far fewer training examples than any learning machine [38]. No doubt this has to do with the fact that effective teachers provide much more than the correct answer to their pupils; they provide an explanation in addition to the result.

In a typical machine learning setup, we present tuples $\{(x_i, y_i)\}_{i=1}^{n}$ to a machine learning model. One way to introduce an "explanation" to a supervised learning system would be to provide some sort of privileged information,
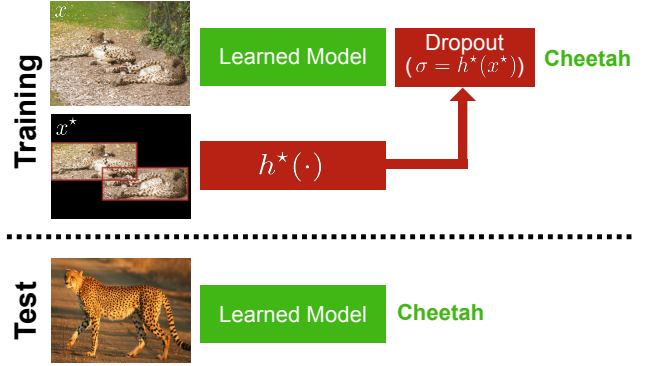
---

[*]Equal contribution

Figure 1. In Learning Under Privileged Information (LUPI) paradigm, a teacher provides additional information during training. In this work, we propose to utilize this information in order to control the variance of the Dropout. Since the Dropout's variance is not constant, we call this a *Heteroscedastic Dropout*. Our empirical and theoretical analysis suggests that *Heteroscedastic Dropout* significantly increses sample efficiency of both CNNs and RNNs resulting in higher accuracy with much less data.

which we entitle $x^\star$. In practice, one can incorporate the triplets $\{(x_i, x_i^\star, y_i)\}_{i=1}^{n}$ into a learning system at training time and the testing stage continues to make use of only x, without any access to $x^\star$. In other words, the "Student" has access to privileged information while interacting with the "Teacher" during training, but in the test stage the "Student" operates without the supervision of the "Teacher". This paradigm is called Learning Under Privileged Information (LUPI) and was introduced by Vapnik and Vashist [38].

Vapnik and Vashist [38] provide a LUPI algorithm for the Support Vector Machines (SVMs). From an algorithmic perspective, the privileged information is utilized to estimate slack values of the SVM constraints. From a theoretical perspective, this algorithm accelerates the rate at which upper bound on error drops from $\mathcal{O}\left(\sqrt{\frac{1}{n}}\right)$ to a far steeper curve of $\mathcal{O}\left(\frac{1}{n}\right)$, where n is the number of required samples.

Privileged information is ubiquitous: it usually exists for almost any machine learning problem. However, we do not see wide adoption of such methods in practice. The major obstacle is the fact that the original LUPI framework

proposed in [38] is only valid for SVM based methods. Indeed, many have shown that the privileged information can be introduced into the loss function under a multi-task or a distillation loss in an algorithm-agnostic way. However, we raise the question, could it and *should* it be fed in as an input instead of an additional task? If so, how would we go about doing so in an algorithm-agnostic way?

We define a new class of LUPI algorithms by making a structural specification. We consider a hypothesis class such that each hypothesis is a combination of two functions – namely, a deterministic function taking x as an input and a stochastic function taking $x^\star$ as an input. When $x^\star$ is not available in the test stage, the "Student" simply makes a Bayes optimal decision and marginalizes the model over $x^\star$. Our structural specification makes this marginalization straightforward while not compromising the expressiveness of the model. This structure is natural in the context of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) thanks to the dropout. Dropout is a widely adopted tool to regularize neural networks by multiplying the activations of a neural network at some layer with a random vector. We simply extend the dropout to *heteroscedastic dropout* by making its variance a function of the privileged information. In other words, dropout becomes the stochastic function taking $x^\star$ as an input and marginalizing the function corresponds to not utilizing dropout in the test phase. In order to be able to train the heteroscedastic dropout, we use Gaussian dropout instead of Bernoulli because the key technical tool we use is the reparameterization trick [21] which is only available for some specific distributions, including the Gaussian.

The rationale behind heteroscedastic dropout follows the close relationship between Bayesian learning and dropout presented by Gal and Gharamani[13]. Dropout can be considered a tool to approximate the uncertainty of the output of a neural network. In our proposed heteroscedastic dropout, the privileged information is used to estimate this uncertainty so that hard examples and easy examples are treated accordingly during training. Our theoretical study suggests that the accurate computation of a model's uncertainty can accelerate the rate at which a CNN's upper bound on error drops, from the typical rate of $\mathcal{O}\left(\sqrt{\frac{1}{N}}\right)$ to a faster $\mathcal{O}(\frac{1}{n})$, where n is the number of training examples. In an oracle case for a dataset with $600K$ training examples, this theoretical upper bound would allows us to learn a model with the same generalization error with $\sqrt{6 \times 10^8} \approx 775$ samples instead of 600K and is thus hugely significant. Although the practical gain we observe is nowhere close, it is still very significant.

We evaluate our method in experiments with both CNNs and RNNs, and show a significant accuracy improvement over two canonical problems, image classification and multi-modal machine translation. As privileged information, we offer a bounding box for image classification and an image of the scene described in a sentence for machine translation. Our method is problem- and modality-agnostic and can be incorporated as long as dropout can be utilized in the original problem and the privileged information can be encoded with an appropriate neural network.

2. Related Work

The key aspects that differentiate our work from the literature are: $i)$ our method is applicable to any deep learning architecture which can utilize dropout, $ii)$ we do not use a multi-task or distillation loss, $iii)$ we provide theoretical justification suggesting higher sample efficiency, $iv)$ we perform experiments for both CNNs and RNNs. A thorough review of the related literature is provided below.

Learning Under Privileged Information: Learning under Privileged Information (LUPI) is initially proposed by Vapnik and Vashist [38, 37]. It extends the Support Vector Machine (SVM) by empirically estimating the slack values via privileged information. This method is further applied to various computer vision problems [31, 29, 11] as well as ranking [30], clustering [10] and metric learning [12] problems. These method are based on max-margin learning and are not applicable to CNNs or RNNs.

One closely related work is [16], extending Gaussian processes to the LUPI paradigm. Hernández-Lobato *et al.* [16] use privileged information to estimate the variance of the noise in their model. Similarly, we use the privileged information to control the variance of the Dropout in CNN and RNN models. However, their method only applies to Gaussian processes, whereas we target neural networks.

Learning CNNs Under Privileged Information: The LUPI paradigm has also been studied recently in the context of CNNs. In contrast to max-margin methods, the literature on learning CNNs under privileged information heavily uses the distillation framework, following the close relationship between distillation and LUPI studied in [25].

Hoffman *et al.* demonstrated a multi-modal distillation approach to incorporating an additional modality as side information [18]. They start with a pre-trained network and distill the information from the privileged network to a main neural network in an end-to-end fashion.

Multi-task learning is a straightforward approach to incorporate privileged information. However, it does not necessarily satisfy a no-harm guarantee (*i.e.* privileged information can harm the learning). More importantly, the no-harm guarantee will very likely be violated since estimating the privileged information (*i.e.* solving the additional task) might be *even more challenging* than the original problem.

When the privileged information is binary and shares the same spatial structure as the original data, such as is the

case with segmentation occupancy or bounding box information, it can also directly be incorporated into the training of CNNs by masking the activations. Group Orthogonal neural networks [41] follow this approach. However, this approach is limited to very specific class of problems.

The loss value of a CNN can be viewed as analogous to the slack variables. Following this analogy, Yang *et al.* [40] use two networks: one for the original task, and one for estimating the loss using the privileged information. Learning occurs through parameter sharing between them.

Our method is different from aforementioned works since we do not use either a distillation or a multi-task loss.

Learning Language under Privileged Visual Information: Using images as privileged information to learn language is not new. Chrupala et al.[5] used a multi-task loss while learning word embeddings under privileged visual information. The embeddings are trained for the task of predicting the next word, as well the representation of the image. Analysis of this model [5, 19] suggests that the embeddings learned by using vision as a privileged information are significantly different than language only ones and correlate better with human judgments. Recently, Elliott *et al.* [8] collected a dataset of images with English captions as well as German translations of captions. Using this dataset, a neural machine translation *under privileged information* model is developed following the multi-task setup [9].

Dropout and its Variants: Dropout is a well studied regularization technique for training deep networks. To-the-best of our knowledge, we are the first to *specifically utilize privileged information* to control the variance of a dropout function. Here, we summarize the existing methods which control the variance of the dropout using variational inference or information theoretical tools. Although these tools have never been applied to the LUPI paradigm, we utilize some of the technical tools developed in these works.

We use multiplicative Gaussian dropout instead of Bernoulli dropout. Gaussian dropout is first introduced in [33]. Its variational extension [22] uses local re-parameterization to perform Bayesian learning.

The Information Bottleneck (IB) [34] is a powerful framework which can enforce various structural assumptions. The IB framework has been applied to CNNs and RNNs using stochastic gradient variational Bayes and the re-parametrization trick [21]. Perhaps closest to our method, Achille and Soatto [1] use the information bottleneck principle to learn disentangled representations when a CNN with Gaussian Dropout is used. The authors introduce many ideas upon which we build; specifically, our hypothesis class (Eqn. 4) is very similar to the architecture they proposed. The main architectural difference is their choice to define the variance as a function of x, whereas we make it function of $x^\star$. We also use similar distributional priors and a similar training procedure. On the other hand, we apply these ideas to a completely different problem with a different theoretical analysis. The information bottleneck has been applied to LUPI for SVMs [26]. However, this method does not apply to neural networks.

Although we use IB [35], Gaussian dropout [33] and the re-parametrization trick [21], we are the first to our knowledge to apply any of these methods to the LUPI problem.

3. Method

Consider a machine learning problem defined over a compact space $\mathcal{X}$ and a label space $\mathcal{Y}$. We also consider a loss function $l(\cdot, \cdot)$ which compares a prediction with a ground truth label. In learning under privileged information, we also have additional information for each data point defined over a space $\mathcal{X}^\star$, which is only available during the training. In other words, we have access to i.i.d. samples from the data distribution as $x_i, x_i^\star, y_i \sim p(x, x^\star, y)$ during training. However, in test we will only be given $x \sim p(x)$. Formally, given a function class $h(\cdot; \mathbf{w})$ parameterized by $\mathbf{w}$ and data $\{x_i, x_i^\star, y_i\}_{i \in [n]}$, a typical aim is to solve the following optimization problem;

$$\min_{\mathbf{w}} E_{x,y \sim p(x,y)}[l(y, h(x; \mathbf{w}))] \tag{1}$$

We propose to do so by learning a multi-view model using both x and $x^\star$ and to use the marginalized model in test when $x^\star$ is not available. Consider a parametric function class for the multi-view data $h^+ : \mathcal{X} \times \mathcal{X}^\star \to \mathcal{Y}$. The training problem becomes:

$$\min_{\mathbf{w}} E_{x,x^\star,y \sim p(x,x^\star,y)}[l(y, h^+(x, x^\star; \mathbf{w}))] \tag{2}$$

This is equivalent to a classical supervised learning problem defined over a space $\mathcal{X} \times \mathcal{X}^\star$ and any existing method like CNNs can be used. In order to solve the inference problem, we consider the following marginalization

$$h(x; \mathbf{w}) \equiv E_{x^\star \sim p(x^\star|x)}[h^+(x, x^\star; \mathbf{w})] \tag{3}$$

The major problem in this formulation is the intractability of this expectation, as $p(x^\star|x)$ is unknown. We propose to restrict the class of functions in a way that the expectation is straightforward to compute. The form we propose is a parametric family such that the privileged information controls the variance, whereas the main information (*i.e.* information available in both training and test) controls the mean. The specific form we use is:

$$h^+(x, x^\star; \mathbf{w}) = h^o(x; \mathbf{w^o}) \odot \mathcal{N}(\mathbf{1}, h^\star(x^\star; \mathbf{w}^\star)) \tag{4}$$

where $\odot$ represents the Hadamard product and the stochastic function $\mathcal{N}(\mathbf{1}, h^\star(x^\star; \mathbf{w}^\star))$ is a normal random variable with a constant mean function and a covariance function parametrized by $x^\star$ and $\mathbf{w}^\star$. We also decompose $\mathbf{w}$ as two
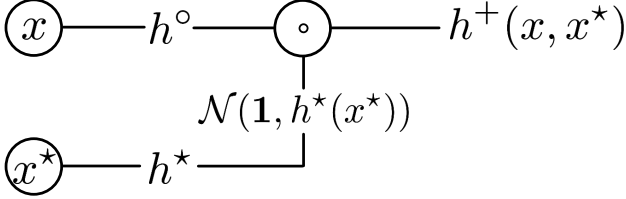
Figure 2. The structure we propose. Privileged information is only used for estimation of the variance of the heteroscedastic dropout.

disjoint vectors as $\mathbf{w} = [\mathbf{w^o}, \mathbf{w^\star}]$. Moreover, in this formulation, the expectation defined in (3) becomes straightforward and can be shown to be $h(x; \mathbf{w}) = h^o(x; \mathbf{w^o})$. We visualize this structural specification in Figure 2.

We use neural networks to represent h^o and $h^\star$ and learn their parameters using the information bottleneck. Since the output space is discrete (we address classification), we denote the representation of the data as $h(x; \mathbf{w})$ and compute the output as softmax$(h(x; \mathbf{w}))$. We explain the details of training in the following sections.

3.1. Information Bottleneck for Learning

We need to control the role of $x^\star$ in LUPI. And, the information bottleneck has already been used for this [26]; however, we do not need this explicit specification because our structural specification directly controls the role of $x^\star$. We use information bottleneck for a rather different reason, its original reason, learning a minimal and sufficient joint representation of $x, x^\star$ which capture all the information about y. This is similar to [1], and we use the same log-Normal assumption. The Lagrangian of the information bottleneck can be written as *(see [35] for details)*;

$$\mathcal{L} = H(y|z) + \beta I(x, x^\star; z) \quad (5)$$

where z is the joint representation of $x, x^\star$ computed as $z = h^+(x, x^\star; \mathbf{w})$. These terms can be computed as;

$$I(x^\star, x; z) = E_{x,x^\star \sim p(x,x^\star)}[KL(p_w(z|x, x^\star)||p_w(z))]$$
$$(6)$$
$$H(y|z) \simeq E_{x,x^\star,y \sim p}[E_{z \sim p_w(z|x,x^\star)}[-\log p_w(y|z)]]$$

where $p_w(\cdot)$ represents the distributions computed over our model with parameters $\mathbf{w}$. In order to compute the KL divergence, we need an assumption about the prior over representations $p(z)$. As suggested by [1], the log-Normal distribution follows the empirical distribution $p(z)$ when ReLu is used. Hence, we use the log-Normal distribution and compute the KL divergence as *(see supplementary materials for full derivation)*;

$$KL(p_w(z|x, x^\star)||p_w(z)) \sim \| \log h^\star(x^\star; \mathbf{w^\star}) \|. \quad (7)$$

Combining them, the final optimization problem is;

$$\min_w \frac{1}{n} \sum_{i=1}^{n} E_{z \sim p_w(z|x,x^\star)}[\log p(y_i|z)] + \beta \| \log h^\star(x_i^\star; w^\star) \|$$
$$(8)$$

This minimization is simply the cross-entropy loss with regularization over the logarithm of the computed variances of the heteroscedastic dropout, and can be performed via the re-parametrization trick in practice when h^o and $h^\star$ are defined as neural networks. We further justify the choice of IB regularization via experimental observation: without it, optimization leads to NaN loss values. We discuss the details of the re-parametrization trick in the following sections.

3.2. Implementation

In this section, we discuss the practical implementation details of our framework. We discuss it for image classification with CNNs and machine translation with RNNs. For the classification setup, we use the image as x, object localization information as $x^\star$, and image label as y. For the translation setup, we use the sentence in the source language as x, an image which is the realization of the sentence as $x^\star$, and the sentence in the target language as y.

We make a sequence of architectural decisions in order to design h° and $h^\star$. For the classification problem, we design both of them as CNNs and share the convolutional layers. The inputs are x, an image, and $x^\star$, an image with a blacked-out background. We use the VGG-Network [32] as an architecture and simply replace each dropout with our form of heteroscedastic dropout. We show the details of the architecture with the re-parameterization trick in Figure 4. We also normalize images with the ImageNet pixel mean and variance. As data augmentation, we horizontally flip images from left to right and make random crops.

We use a two-layered LSTM architecture with 500 units as our RNN cell and use the heteroscedastic dropout between layers of LSTMs. The main reason behind this choice is the fact that dropout in general is only shown to be useful for connections between LSTM layers. We use attention [2] and feed the image as a feature vector computed using the VGG[32] architecture pre-trained on ImageNet. We give the details of the LSTM with re-parametrization trick in Figure 3. For the inference, we use beam search over 12 hypotheses. Our LSTM implementation directly follows the baseline implementation provided by OpenNMT [23].

Hyperparameter Settings We used a standard learning rate across all image classification experiments, setting our initial learning rate to 1.0×10^{-3}, and tolerating 5 epochs of non-increasing validation set accuracy before decaying the learning rate by 10x. For multi-modal machine translation, we use an initial learning rate of 1.0×10^{-3} and half the learning rate every epoch after the 8th epoch. We used the ADAM [20] optimizer in PyTorch for both image classification and multi-modal machine translation. All CNN weights are initialized according to the method of He *et al.* [15] and a decay of 1×10^{-4} was used for image classification. For multi-modal machine translation, we do not use any weight decay and initialize weights according to [23].
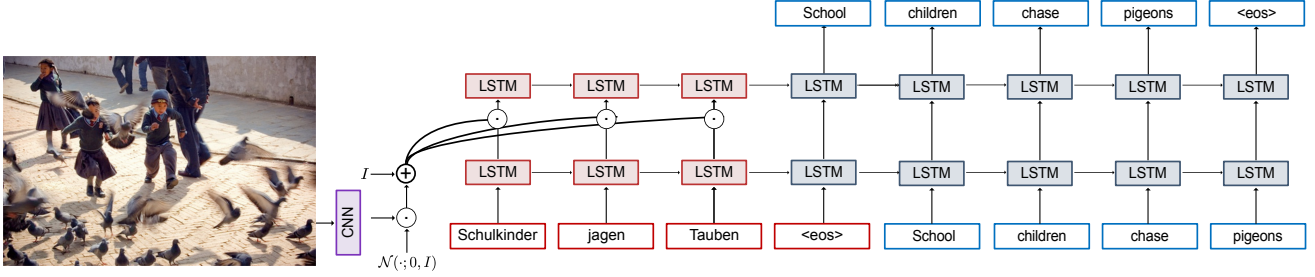
Figure 3. **Multi-Modal Machine Translation** We show the LSTM architecture we use, which incorporates the re-parameterization trick and heteroscedastic dropout connections. We use dropout only between layers and share among cells following [14]. We do not use any dropout in inference since the image is not available during test.
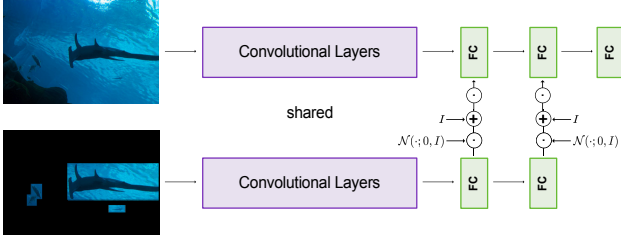

Figure 4. **Image Classification** We show the CNN architecture we used in our experiments, along with the re-parameterization trick and heteroscedastic dropout connections. We do not use any dropout in inference since localization bounding boxes are not available during test.

4. Experimental Results

In order to evaluate our method, we perform various experiments using both CNNs and LSTMs. We test our method with CNNs for the task of image classification and with LSTMs for the task of machine translation. We discuss the baselines against which we compare our algorithm, the datasets we use in the rest of this section.

Datasets: We perform our experiments using the following datasets; **ImageNet [6]:** A dataset of 1.3 million images labelled with a class over 1000 categories. We only use the subset of 600 thousand images which include localization information. **Multi-30K[8]:** A dataset of 30 thousand Flickr images which are captioned in both English and German. We use this dataset for multi-modal machine translation experiments. In Multi-30K, the English captions are generated for images; whereas, German captions are directly translated from the English captions. Hence, during the ground truth translation, the images were privileged information translators. This property makes this dataset a perfect benchmark for LUPI.

Baselines: We compare our method against the following baselines. **No-$x^\star$:** a baseline model not using any privileged information **Gaussian Dropout [33]:** A multiplicative Gaussian dropout with a fixed variance. **Multi-Task:** We perform multi-task learning as a tool to utilize privileged information. We use regression to bounding box coordinates and denote it as Multi-Task w/ B.Box

as well as direct estimation of the RGB mask and denote it as Multi-Task w/ Mask. We use this self-baseline only for CNNs since there are many published multi-task methods for machine translation with multi-modal information and we compare with them all. In addition to these self-baselines, we also compare with the following published work. **GoCNN[41]:** a method for CNNs with segmentation as a privileged information which proposes to mask convolutional weights with segmentation masks. **Information Dropout[1]:** a regularization method that utilizes injection of multiplicative noise in the activations of a deep neural network (but as a function of the input x, not $x^\star$). **MIML-FCN[40]:** a CNN-based LUPI framework designed for multi-instance problems. Our problem is not multiple instance; but, we still compare for the sake of completeness. **Modality Hallucination[18]:** Distillation-based LUPI method designed for multi-modal CNNs. **Imagination [9]:** Distillation-based LUPI method designed for multi-modal machine translation. *(see supplementary materials for implementation details.)*

4.1. Effectiveness of Our Method

We compare our method with the No-$x^\star$ baseline for image classification using the ImageNet dataset. We perform experiments by varying the number of training examples logarithmically. This is key since the main motivation behind our LUPI method is *learning with less data* rather than having higher accuracy. We report several results in Table 1 and visualize additional data points in Figure 5.

Our method is quite effective for a training dataset size of $75K$ images; however, it has no positive impact for the $200K$ and $600K$ cases. Even more importantly, the smaller the training set, the larger the improvement. For $75K$ images, it results in 5% single crop top-1 accuracy improvement; whereas, for $200K$, it matches the performance. This simply suggests that our algorithm is particularly effective for low- and mid-scale datasets. This result is quite intuitive since with increasing dataset size, all algorithms can effectively learn and reach an optimal accuracy which is possible under the model class. Hence, the role of an "intelligent teacher" is providing the privileged information to

Table 1. Classification Test Accuracy on 1000 ILSVRC Classes. Because the ILSVRC server prohibits large numbers of test submissions, which we required to evaluate at different sizes of sample data, we use a hold-out set of $50K$ images from ImageNet CLS-LOC as our test set. The authors of [32] report a 7.4% Multi-Crop, top-5 error rate when training on $\sim 1.3M$ images. Where we report "No-$x^\star$," we describe the results of a classical CNN learning method. All 1-crop evaluations below were carried out with a center crop. All 25K models diverged.

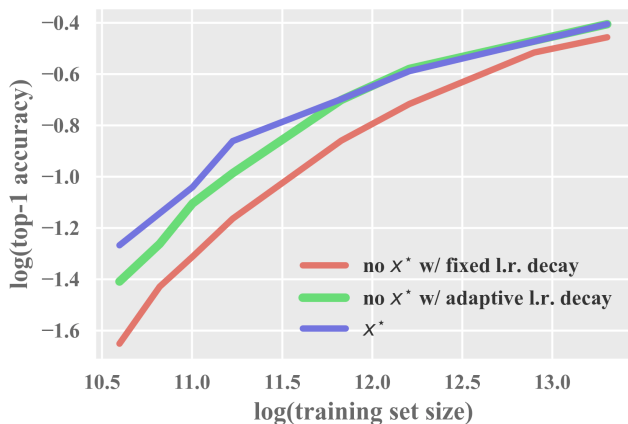|	Number of Training Images			
Model	25K	75K	200K	600K
Single Crop top-1				
No-$x^\star$	-	37.85	**55.99**	66.66
Our LUPI	-	**42.30**	55.51	**66.77**
Single Crop top-5				
No-$x^\star$	-	62.76	**79.21**	**86.90**
Our LUPI	-	**67.13**	78.89	86.88
Multi-Crop top-1				
No-$x^\star$	-	39.99	**58.7**	**69.20**
Our LUPI	-	**44.95**	58.41	69.10
Multi-Crop top-5				
No-$x^\star$	-	64.49	81.0	88.60
Our LUPI	-	**69.19**	**81.15**	**88.64**

Figure 5. Accuracy vs. training set size for ImageNet classification. Each data point denotes a VGG-16 network trained with batch normalization. The accuracies of models trained *with* $x^\star$ are depicted in **green**; those trained *without* are depicted in **red** (via an adaptive learning rate decay schedule) and **blue** (via a fixed learning rate decay schedule). **Furthermore, adaptively modifying the learning rate according to performance on a hold-out set yields massive gains in low- and mid-scale data regimes**, when compared with decaying the learning rate at fixed intervals, *e.g.* every 30 training epochs.

learn with less data. In other words, LUPI is not a way to gain extra accuracy regardless of the dataset; rather, *it is a way to significantly increase the data efficiency*. We do not perform a similar experiment for machine translation since the available dataset is a mid-scale and our LUPI method

demonstrates asymptotic accuracy increases at full dataset.

4.2. Data Efficiency of Our Method and Baselines

In order to compare the data efficiency gain of our method against baselines, we perform image classification and multi-modal machine translation experiments. We use $75K$ ImageNet images since our main goal is identify insights regarding data sample efficiency gains and using a smaller training set makes this analysis possible. We summarize the image classification experiments in Table 2 and multi-modal machine translation experiments in Table 3. Our method outperforms all baselines for both tasks, for image classification with a significant margin.

Table 2. We compare our method's performance with several baselines. We train with 75 Images per each of the 1000 ImageNet classes, leaving us with 75×10^3 images in total. We outperform each model and are competitive with GoCNN, a model specifically designed for the problem of learning with segmentation data using various architectural decisions. Evaluation is carried out on the held-out set of images from our holdout test set.

	Single Crop		Multi-Crop	
Model	top-1	top-5	top-1	top-5
No-$x^\star$ [32]	37.85	62.76	39.99	64.49
MIML-FCN [40]/ResNet	35.61	59.66	38.3	62.3
Modal. Hallucination [18]	37.66	63.15	40.45	65.95
Info. Dropout [1]	38.09	63.52	41.84	67.47
Gaussian Dropout [33]	38.80	63.64	41.0	65.3
MIML-FCN [40]/VGG	39.54	64.43	42.0	66.4
Multi-Task w/ Bbox	39.96	64.79	42.4	66.6
Multi-Task w/ Mask [27]	40.48	65.62	43.18	67.68
GoCNN	41.43	66.78	44.5	**69.3**
Our LUPI	**42.30**	**67.13**	**44.95**	69.19

Table 3. We compare our method for multi-modal machine translation with several baselines. We report BLEU[28, 24] and METEOR[7] metrics. Some baselines only report English(en)→German(de) results, and exclude de→en.

	en→de		de→en	
Model	BLEU	Meteor	BLEU	Meteor
No $x^\star$ (following [23])	35.5	54.0	40.19	55.8
Multi-Modal				
Toyama et al. [36]	36.5	56.0	—	—
Hitschler et al. [17]	34.3	56.1	—	—
Calixto et al. [3]	36.5	55.0	—	—
Calixto et al. [4]	37.3	55.1	—	—
LUPI				
Imagination [9]	36.8	55.8	40.5	56.0
Ours	**38.4**	**56.9**	**42.4**	**57.1**

Image Classification with Privileged Localization

One interesting result is that our network $h^\star$ is clearly learning much more than to predict random constant for

its output Σ, the covariance matrix used for reparameterization; in fact, our network outperforms the network that produces a Σ whose entries are drawn from pure Gaussian noise by $> 3.8\%$. We analyze our method for CNNs both theoretically and qualitatively in Section 5 and conclude that our method learns to control the uncertainty of the model and results in an order of magnitude higher data efficiency, explaining this large margin.

Furthermore, GoCNN [41], an architecture specifically designed for the problem of learning with segmentation data using various architectural decisions, results in a significant accuracy improvement competitive with our method in a small dataset regime. However, GoCNN's performance relative to other baselines begins to degrade at a dataset size of 200K images, leading to a top-1 accuracy decrease of -5.26% in comparison with Bernoulli dropout and -4.47% with our heteroscedastic dropout method. This is an intuitive result because GoCNN's rigid architectural decisions inject significant bias into the model.

Because Information Dropout relies upon sampling from a log-normal distribution with varying variance, it is heteroscedastic. However, compounded Information Dropout layers which exponentiate samples from a normal distribution lead to unbounded activations; thus, a suitable squashing function like the sigmoid must be employed to bound the activations. We find its performance can actually decrease accuracy when compared with a ReLU nonlinearity.

Multi-modal Machine Translation Our method results in a significant accuracy improvement measure by both BLEU and METEOR scores. One interesting observation is that our method outperforms various multi-modal methods which use image information in both training and test. This counterintuitive result is due to the way in which the dataset is collected. Multi-30k[8] dataset is collected by simply translating the English captions of $30K$ images into German. A LUPI model [9] was already shown to perform better than multi-modal translation models which can use both images and sentences in test time [3, 4, 17, 36]. This surprising result is largely due to the fact that the translators did not see the images while providing ground truth translations. More importantly, the effectiveness of visual information in machine translation in a privileged setting is also intuitive following the results of [5]. Chrupala et al.[5] show that when image information is used as privileged information in the learning of word representations, the quality of such representations increases. Hence, a multi-modal paradigm for learning language (e.g. with privileged visual information) and vice versa is a fruitful direction for both natural language processing and computer vision communities and our method performs quite effectively on this task.

In summary, our results overperform all baselines for both multi-modal machine translation and image classification experiments using both CNNs and RNNs. These results

Figure 6. Accuracy vs. $x^\star\%$ for multi-modal machine translation. An identical experiment on image classification is shown in supplementary material due to limited space.

suggest that our method is effective and generic.

4.3. Learning under Partial Privileged Information

Although privileged information naturally exists for many problems, it is typically not available for all points. Thus, it is common to encounter a scenario in which the entire training data is labelled; however, only a small portion includes privileged information. In other words, we typically have dataset which is the union of $\{x_i, y_i\}_{i \in [n]}$ and $\{x_j, x_j^\star, y_j\}_{j \in [m]}$ where $m \ll n$. In order to experiment with this setting, we vary the amount of $x^\star$ available. We present the result in Figure 6 for machine translation and in supplementary material for image classification.

The results in Figure 6 suggest that even when only a small portion (2% for machine translation, 4% for image classification) of the data has privileged information, our method is effective resulting in a significant accuracy increase very similar to the one we obtained with 100% of privileged information.

5. Analysis of the Algorithm

Our empirical analysis suggests a strong data-efficiency increase when privileged information is incorporated using our method. It is interesting to quantify this increase in terms of the theoretical learning rate. For the case of SVMs, Vapnik *et al.* [38] showed that utilizing the privileged information can result in a generalization error bound with rate $\mathcal{O}(\frac{1}{n})$ instead of $\mathcal{O}(\sqrt{\frac{1}{n}})$ where n is the dataset size. Our experimental results suggests a similar story for CNNs, but the theoretical justification can not be extended from [38] since their analysis is specific to SVMs. In this section, we try to answer this question for our algorithm. We show that our method is capable of converting an $\mathcal{O}(\sqrt{\frac{1}{n}})$ error rate (derived in Proposition 1) into $\mathcal{O}(\frac{1}{n})$ in an oracle setting for CNNs. We rigorously prove that it is possible to reach $\mathcal{O}(\frac{1}{n})$ rate using our structural assumptions; however, we do not provide any argument for the optimization landscape. In other words, our results are only valid with an oracle optimizer which can find the solution satisfying our assumptions. The study of the loss function and the optimization remains an open problem; however, we give a strong empirical evidence that using SGD with information bottleneck regularization is enabling faster learning.

We start by presenting a bound over the generalization error of CNNs with no privileged information. This result

directly follows from [39], and we include it here for the sake of completeness. Loss functions of CNNs based on l_2 distance are Lipschitz continuous when the non-linearity is the rectified linear unit, the pooling operation is max-pooling and softmax function is used to convert activations into logits. Moreover, any learning algorithm with a Lipschitz loss function admits the following result [39];

Proposition 1 ([39, Example 4]). *Given n i.i.d. samples drawn from $p(x,y)$ as $\{\mathbf{x}_i, y_i\}_{i \in [n]}$, if a loss function $l(y, h(x; \mathbf{w}))$ is λ^l-Lipschitz continuous function of x for all $y, \mathbf{w}$, bounded by L and $\mathcal{X} \times \mathcal{Y}$ has a covering number $N_\epsilon(\mathcal{X}, |\cdot|_2) = K$, then with probability at least $1 - \delta$,*

$$\left| E_{\mathbf{x},y \sim p(x,y)}[l(y, h(\mathbf{x}; w))] - \frac{1}{n}\sum_{i \in [n]} l(y_i, h(\mathbf{x}_i; w)) \right|$$

$$\leq \lambda^l \epsilon + L\sqrt{\frac{2K \log 2 + 2\log(1/\delta)}{n}}.$$

This proposition simply details the baseline $\mathcal{O}(\sqrt{\frac{1}{n}})$ error rate under no privileged information. In order to intuitively explain how our algorithm can accelerate this learning to $\mathcal{O}(\frac{1}{n})$, consider the following oracle algorithm. Using the privileged information, one can estimate the uncertainty (variance) of the neural network and can use the inverse of this estimate as a the variance of the heteroscedastic dropout. Since the heteroscedastic dropout is multiplicative, this results in unit variance regardless of the input. In a similar fashion, this oracle algorithm can bound the variance with an arbitrary constant. Following this oracle algorithm, we show that when the variance is properly controlled, our method can reach an $\mathcal{O}(\frac{1}{n})$ rate. Consider the population distribution of number images per class versus the empirical distribution as $\epsilon_y = E[y^\intercal y] - \frac{1}{n}\sum_i y_i^\intercal y_i$. The value ϵ_y is purely a property of the way in which a was dataset collected and must be treated independently of the learning. Hence, we do not study the rate at it vanishes. We present the following proposition and defer its proof to the supplementary material:

Proposition 2. *Given n i.i.d. samples drawn from $p(x, y)$ as $\{\mathbf{x}_i, y_i\}_{i \in [n]}$ and a loss function defined as $\|h(x; w) - y\|_2^2$ where $h(\cdot; w)$ is a CNN, assume that any path between input and output has maximum weight M_w, total number of paths between input and output is P, and for all training points x_i, $h(x_i; w) \leq M_z$ and $Var(h(x_i; w)) \leq \xi^2$. With probability at least $1 - \delta$,*

$$\left| E_{\mathbf{x},y \sim p(x,y)}[l(y, h(\mathbf{x}; w))] - \frac{1}{n}\sum_{i \in [n]} l(y_i, h(\mathbf{x}_i; w)) \right|$$

$$\leq \frac{2C\left((\xi + 1)\log\frac{1}{\delta} + M_w\left(3\xi + M_z\right)\log\frac{P}{\delta}\right)}{3n} + (2C + 1)\epsilon_y.$$

where $\xi \leq \delta$.

This proposition means that learning with sample efficiency $\mathcal{O}(\frac{1}{n})$ is indeed possible as long as one can bound the variance of the output(ξ) with an arbitrary number δ. Hence,

the full control of the output variance, makes learning with higher sample efficiency possible. One remaining question is *whether using SGD with information bottleneck regularization can learn this oracle solution or not?* Unfortunately, we have no theoretical answer for this question and leave it as an open problem. However, we study this problem empirically and show that there is a strong empirical evidence suggesting that the answer is affirmative.
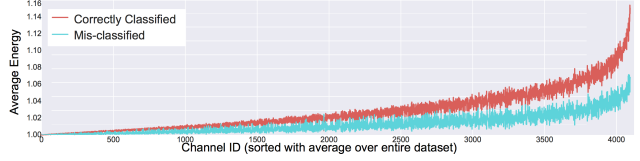

Figure 7. For 8000 random samples from the validation set that our heteroscedastic dropout algorithm mis-classifies, as well as 8000 random samples it correctly classifies, we plot the average of activations per dimension (we sort the 4096 dimensions in terms of average energy over full dataset for clarity).

A realistic estimate of variance is typically not possible without a strong parametric assumption; however, we can use the simple heuristic that the samples from validation set that our algorithm mis-classifies should have higher variance than the samples which are correctly classified. We plot the average energy of computed dropout variances per fully connected neuron for mis-classified and correctly classified examples in Figure 7. Interestingly, our method consistently assigns larger multiplier (dropout) values for correctly classified samples and significantly smaller values for mis-classified samples. This strongly supports our hypothesis since when the low heteroscedastic dropout is multiplied with the high-variance mis-classified examples, their final variance will be low, possibly bounded by the σ_0.

6. Conclusion

We described a learning under privileged information framework for CNNs and RNNs. We proposed a heteroscedastic dropout formulation by making the variance of the dropout a function of privileged information.

Our experiments on image classification and machine translation suggest that our method significantly increases the sample efficiency of both CNNs and LSTMs. We further provide an upper bound over the generalization error of CNNs suggesting a sample efficient learning (with rate $\mathcal{O}(\frac{1}{n})$) in the oracle case when privileged information is available. We make our learned models as well as the source code available [1] [2].

7. Acknowledgements

---

[1]http://svl.stanford.edu/projects/heteroscedastic-dropout
[2]https://github.com/johnwlambert/dlupi-heteroscedastic-dropout

References

[1] A. Achille and S. Soatto. Information dropout: Learning optimal representations through noisy computation. *arXiv preprint arXiv:1611.01353*, 2016. 3, 4, 5, 6

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 4

[3] I. Calixto, Q. Liu, and N. Campbell. Doubly-attentive decoder for multi-modal neural machine translation. *arXiv preprint arXiv:1702.01287*, 2017. 6, 7

[4] I. Calixto, Q. Liu, and N. Campbell. Incorporating global visual features into attention-based neural machine translation. *arXiv preprint arXiv:1701.06521*, 2017. 6, 7

[5] G. Chrupała, A. Kádár, and A. Alishahi. Learning language through pictures. *arXiv preprint arXiv:1506.03694*, 2015. 3, 7

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 5

[7] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014. 6

[8] D. Elliott, S. Frank, K. Sima'an, and L. Specia. Multi30k: Multilingual english-german image descriptions. pages 70–74, 2016. 3, 5, 7

[9] D. Elliott and Á. Kádár. Imagination improves multimodal translation. *CoRR*, abs/1705.04350, 2017. 3, 5, 6, 7

[10] J. Feyereisl and U. Aickelin. Privileged information for data clustering. *Information Sciences*, 194:4–23, 2012. 2

[11] J. Feyereisl, S. Kwak, J. Son, and B. Han. Object localization based on structural svm using privileged information. In *Advances in Neural Information Processing Systems (NIPS))*, pages 208–216. 2014. 2

[12] S. Fouad, P. Tino, S. Raychaudhury, and P. Schneider. Incorporating privileged information through metric learning. *IEEE Transactions on Neural Networks and Learning Systems*, 24(7):1086–1098, 2013. 2

[13] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016. 2

[14] Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016. 5

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, June 2016. 4

[16] D. Hernández-lobato, V. Sharmanska, K. Kersting, C. H. Lampert, and N. Quadrianto. Mind the nuisance: Gaussian process classification using privileged noise. In *Advances in Neural Information Processing Systems 27*, pages 837–845. 2014. 2

[17] J. Hitschler, S. Schamoni, and S. Riezler. Multimodal pivots for image caption translation. *arXiv preprint arXiv:1601.03916*, 2016. 6, 7

[18] J. Hoffman, S. Gupta, and T. Darrell. Learning with side information through modality hallucination. In *Computer Vision and Pattern Recognition*, June 2016. 2, 5, 6

[19] A. Kádár, G. Chrupała, and A. Alishahi. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 2017. 3

[20] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations*, 2015. 4

[21] D. Kingma and M. Welling. Auto-encoding variational bayes. In *The International Conference on Learning Representations*, 2014. 2, 3

[22] D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015. 3

[23] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*, 2017. 4, 6

[24] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007. 6

[25] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations*, 2016. 2

[26] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Information bottleneck learning using privileged information for visual recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 1496–1505, 2016. 3, 4

[27] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision*, pages 1520–1528, Dec 2015. 6

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 6

[29] V. Sharmanska and N. Quadrianto. In the era of deep convolutional features: Are attributes still useful privileged data? In *Visual Attributes*, pages 31–48. Springer, 2017. 2

[30] V. Sharmanska, N. Quadrianto, and C. H. Lampert. Learning to rank using privileged information. In *Proceedings of the International Conference on Computer Vision*, pages 825–832, 2013. 2

[31] V. Sharmanska, N. Quadrianto, and C. H. Lampert. Learning to transfer privileged information. *arXiv preprint arXiv:1410.0389*, 2014. 2

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 4, 6

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. 3, 5, 6

[34] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 3

[35] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop (ITW)*, pages 1–5, April 2015. 3, 4

[36] J. Toyama, M. Misono, M. Suzuki, K. Nakayama, and Y. Matsuo. Neural machine translation with latent semantic of image and text. *arXiv preprint arXiv:1611.08459*, 2016. 6, 7

[37] V. Vapnik and R. Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16:2023–2049, 2015. 2

[38] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(56):544 – 557, 2009. 1, 2, 7

[39] H. Xu and S. Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012. 8

[40] H. Yang, J. Tianyi Zhou, J. Cai, and Y. Soon Ong. Miml-fcn+: Multi-instance multi-label learning via fully convolutional networks with privileged information. In *Computer Vision and Pattern Recognition*, July 2017. 3, 5, 6

[41] J. F. S. Y. Yunpeng Chen, Xiaojie Jin. Training group orthogonal neural networks with privileged information. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 1532–1538, 2017. 3, 5, 7