Automatic Data Augmentation via Invariance-Constrained Learning

Ignacio Hounie [1] **Luiz F. O. Chamon** [2] **Alejandro Ribeiro** [1]

Abstract

Underlying data structures, such as symmetries or invariance to transformations, are often exploited to improve the solution of learning tasks. However, embedding these properties in models or learning algorithms can be challenging and computationally intensive. Data augmentation, on the other hand, induces these symmetries during training by applying multiple transformations to the input data. Despite its ubiquity, its effectiveness depends on the choices of which transformations to apply, when to do so, and how often. In fact, there is both empirical and theoretical evidence that the indiscriminate use of data augmentation can introduce biases that outweigh its benefits. This work tackles these issues by automatically adapting the data augmentation while solving the learning task. To do so, it formulates data augmentation as an invariance constrained learning problem and leverages Monte Carlo Markov Chain (MCMC) sampling to solve it. The result is an algorithm that not only does away with a priori searches for augmentation distributions, but also dynamically controls if and when data augmentation is applied. We validate empirically our theoretical developments in automatic data augmentation benchmarks for CIFAR and ImageNet-100 datasets. Furthermore, our experiments show how this approach can be used to gather insights on the actual symmetries underlying a learning task.

1. Introduction

Exploiting the underlying structure of data is a key principle of data analysis. Its use has been fundamental to the success of machine learning solutions, from the translational equivariance of convolutional neural networks (Fukushima and Miyake, 1982) to the invariant attention mechanism in Alphafold (Jumper et al., 2021). However, embedding invariances and symmetries in model architectures is hard in general and when possible, often incurs a high computational cost. This is the case, of rotation invariant neural network architectures that rely on group convolutions, which are feasible only for small, discrete transformation spaces or require coarse undersampling due to their high computational complexity (Cohen and Welling, 2016; Finzi et al., 2020).

A widely used alternative consists of modifying the data rather than the model. That is, to *augment* the dataset by applying transformations to samples in order to induce the desired symmetries or invariances during training. Data augmentation, as it is commonly known, is used to train virtually all state-of-the-art models in a variety of domains (Shorten and Khoshgoftaar, 2019). This empirical success is supported by theoretical results showing that, when the underlying data distribution is invariant to the applied transformations, data augmentation provides a better estimation of the statistical risk (Chen et al., 2019; Sannai et al., 2019; Lyle et al., 2020; Shao et al., 2022). On the other hand, applying the wrong transformations can introduce biases that may outweigh these benefits (Chen et al., 2019; Shao et al., 2022).

Choosing which transformations to apply, when to do so, and how often, is thus paramount to achieving good results. However, it requires knowledge about the underlying distribution of the data that is typically unavailable in learning settings. Several approaches to learning an augmentation policy or distribution over a given set of transformations exist, such as reinforcement learning (Cubuk et al., 2018), genetic algorithms (Ho et al., 2019), density matching (Lim et al., 2019; Cubuk et al., 2020; Hataya et al., 2020), gradient matching (Zheng et al., 2022), bi-level optimization (Li et al., 2020b; Liu et al., 2021), jointly optimizing over transformations using regularised objectives (Benton et al., 2020), variational bayesian inference (Chatzipantazis et al., 2021), bayesian model selection (Immer et al., 2022) and alignment regularization (Wang et al., 2022). Nevertheless, optimization based methods often require computing gradients with respect to transformations (Chatzipantazis et al., 2021; Li et al., 2020b) and several of these methods resort to computationally intensive search phases, optimization of auxiliary

[1]University of Pennsylvania [2]University of Stuttgart. Correspondence to: Ignacio Hounie <ihounie@seas.upenn.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

models, or additional data, while failing to outperform fixed user-defined augmentation distributions (Müller and Hutter, 2021).

In this work, we formulate data augmentation as an invariance-constrained learning problem. That is, we specify a set of transformations and a desired level of invariance or robustness with respect to these transformations, and recover an augmentation distribution that imposes this requirement on the learned model. We do this in a non-parametric fashion, i.e., without explicitly parametrising the distribution over transformations.

More specifically, our invariance requirement is weighted by the probability of each data point. Hence, we require the output of our model to be stable only on the support of the underlying data distribution, and more so on common samples. By imposing this requirement as a constraint on the learning task and leveraging recent duality results, the amount of data augmentation can be automatically adjusted during training. An advantage of the constrained learning formulation is that it mitigates the potential biases introduced by data augmentation without doing away with its potential benefits. We propose an algorithm that combines stochastic primal-dual methods and MCMC sampling to do away with the need for transformations to be differentiable. We evaluate our method in automatic data augmentation benchmarks in CIFAR and ImageNet-100 datasets. Furthermore, we show how our method provides insights on the actual symmetries underlying a learning task, using synthetically invariant MNIST and Fashion-MNIST datasets.

2. Data Augmentation in Supervised Learning

As in the standard supervised learning setting, let $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ denote a feature vector and $y \in \mathcal{Y} \subseteq \mathbb{R}$ its associated label or measurement. For classification tasks, we take $\mathcal{Y} \subseteq \mathbb{N}$. Let $\mathfrak{D}$ denote a probability distribution over the data pairs $(\mathbf{x}, y)$ and $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ be a non-negative, convex loss function, e.g., the cross entropy loss. Our goal is to learn a predictor $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$ in some hypothesis class $\mathcal{H}_{\boldsymbol{\theta}} = \{f_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p\}$ that minimizes the expected loss,

$$\underset{\boldsymbol{\theta}\in\Theta}{\text{minimize}} \;\; R(f_{\boldsymbol{\theta}}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}}[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), y)]. \tag{SRM}$$

We consider the distribution $\mathfrak{D}$ to be unknown, except for the dataset $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$ of n i.i.d. samples from $\mathfrak{D}$. Therefore, we rely on the empirical approximation of the objective of (SRM),

$$\hat{R}(f_{\boldsymbol{\theta}}) := \frac{1}{N}\sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i). \tag{1}$$

One of the aims of data augmentation is to improve the approximation $\hat{R}$ of the statistical risk R when dealing with a dataset that is not sufficiently representative of the data distribution. To do so, we consider transformations of the feature vector $g : \mathcal{X} \to \mathcal{X}$, taken from the (possibly infinite) transformation set $\mathcal{G}$. Common examples include rotations and translations in images. Data augmentation leverages these transformations to generate new data pairs $(g\mathbf{x}, y)$ by sampling transformations according to a probability distribution $\mathfrak{G}$ over $\mathcal{G}$, leading to the learning problem

$$\underset{\boldsymbol{\theta}\in\Theta}{\text{minimize}} \;\; \hat{R}_{\text{aug}}(f_{\boldsymbol{\theta}}) := \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{g\sim\mathfrak{G}}\left[\, \ell(f_{\boldsymbol{\theta}}(g\mathbf{x}_i), y_i)\right]. \tag{2}$$

Note that the empirical risk approximation $\hat{R}$ in (1) can be interpreted as an approximation of the data distribution $\mathfrak{D}$ by a discrete distribution that places atoms on each data point. In that sense, $\hat{R}_{\text{aug}}$ in (2) can be thought of as the Vicinal Risk Minimization (Chapelle et al., 2000) counterpart of (1), in which the atoms on $\mathbf{x}_i$ are replaced by a local distribution over the transformed samples $g\mathbf{x}_i$, i.e.,

$$\hat{R}_{\text{aug}}(f_{\boldsymbol{\theta}}) = \frac{1}{N}\sum_{i=1}^{N} \int \ell\left(f_{\boldsymbol{\theta}}(g\mathbf{x}_i), y_i\right) dP(g\mathbf{x}_i), \tag{3}$$

where the measure P over $\mathcal{X}$ is induced by the distribution $\mathfrak{G}$ over $\mathcal{G}$. As it can be seen from (3) if $\mathfrak{G}$ is not chosen adequately, $\hat{R}_{aug}$ can be a poor estimate of R, introducing biases that outweigh the benefits of data augmentation (Chen et al., 2019; Shao et al., 2022). On the other hand, if the data distribution $\mathfrak{D}$ is statistically invariant under the action of $\mathcal{G}$, and $\mathfrak{G}$ is chosen so as to induce those invariances in the solution, then learning using (2-3) has provable advantages in terms of sample complexity (Chen et al., 2019; Bietti and Mairal, 2019; Sannai et al., 2019; Lyle et al., 2020).

In this work, we tackle the choice of $\mathfrak{G}$ given a set of transformations $\mathcal{G}$, i.e., how to sample transformations so that the solution of the learning problem is approximately invariant, as defined on the next section. Unlike *invariance learning* (Jebara, 2003; Zhou et al., 2021a; Benton et al., 2020; Immer et al., 2022), we do not seek to learn the transformations $\mathcal{G}$ from the data. We also explore how to incorporate the expectation of the risk of transformed samples in the learning algorithm, so that even when the distribution is not invariant — or the hypothesis space is not rich enough to capture invariances of the data — we can still avoid introducing a bias. Note that invariance to transformations in $\mathcal{G}$ may hold for the true distribution or may also be a desirable property of the solution, for example, to achieve robustness (Kanbak et al., 2017; Volpi et al., 2018; Joshi et al., 2019).

3. From Invariance to an Augmentation Distribution

It can be straightforward to specify a set of transformations $\mathcal{G}$ to which the solution should be *approximately* invariant (e.g., image rotations and translations). However, finding a transformation distribution $\mathfrak{G}$ that leads to the desired properties in the solution can be challenging. What is more, using a fixed $\mathfrak{G}$ as in (2) prevents us from controlling when and how much augmentation is used during training, running the risk of biasing the final solution.

In the next sections, we explain how a data augmentation distribution can be obtained from imposing invariance. We first argue how *invariance* leads to a robustness requirement (Section 3.1) and then show that this invariance constrained learning problem yields an augmentation distribution (Section 3.2).

3.1. From Invariance to Robustness

If the data is invariant to action g, we want to learn a model that respects this invariance. This induces the requirement to learn a model such that $f_{\boldsymbol{\theta}}(\mathbf{x}) = f_{\boldsymbol{\theta}}(g\mathbf{x})$ for all $g \in \mathcal{G}$. Here, we focus on learning models in which invariance requirements are stated in terms of the loss function and we therefore require that the model satisfies

$$\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), y) = \ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y), \text{ for all } g \in \mathcal{G} . \tag{4}$$

If the model is indeed invariant in the sense that $f_{\boldsymbol{\theta}}(\mathbf{x}) = f_{\boldsymbol{\theta}}(g\mathbf{x})$, (4) holds as well. The advantage of (4) is that it explicitly incorporates the structure of the learning task. This is useful when the model is not perfectly invariant, in which case the difference $\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y) - \ell(f_{\boldsymbol{\theta}}(\mathbf{x}), y)$ measures the cost of not having invariance in terms of the loss function. We leverage this observation to define the invariance loss as

$$\ell_{\text{inv}}(f_{\boldsymbol{\theta}}, \mathbf{x}, y) := \max_{g \in \mathcal{G}} \ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y) - \ell(f_{\boldsymbol{\theta}}(\mathbf{x}), y). \tag{5}$$

This loss is always nonnegative because the identity action is assumed to be part of the invariant set $\mathcal{G}$. In fact, the set $\mathcal{G}$ defines an equivalence class of points $g\mathbf{x}$ that we know map to the same output y. The loss ℓ_{inv} defined in (5) is a measurement of how far the learnable model $f_{\boldsymbol{\theta}}$ is from defining a corresponding equivalence class with respect to the loss. We can therefore define an invariance risk by taking the average of the invariance loss in (5) with respect to the data distribution,

$$R_{\text{inv}}(f_{\boldsymbol{\theta}}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}} \left[\ell_{\text{inv}}(f_{\boldsymbol{\theta}}, \mathbf{x}, y)\right]. \tag{6}$$

It is interesting to substitute (5) into (6). Using the linearity of expectations we can conclude that the invariance risk is

$$\begin{aligned} R_{\text{inv}}(f_{\boldsymbol{\theta}}) \;=\; & \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}} \left[\max_{g \in \mathcal{G}} \ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)\right] \\ & - \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}} \left[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), y)\right]. \end{aligned} \tag{7}$$

The second term in (7) is the standard statistical risk without data augmentation. The first term can be interpreted as a data augmentation distribution [cf. (2)] in which all of the augmentation mass is allocated to the element $g\mathbf{x}$ of the invariance set $\mathcal{G}$ for which the loss is highest. This term can also be interpreted as a robust adversarial loss, e.g., (Madry et al., 2017), in which the adversary chooses the element $g\mathbf{x} \in \mathcal{G}$ with the highest loss.

Note that minimizing the invariance risk R_{inv} does not necessarily lead to a good model $f_{\boldsymbol{\theta}}$ but rather one that has constant loss under transformations of the input. This motivates a problem formulation in which we want to find models $f_{\boldsymbol{\theta}}$ with small statistical risk R [cf. (SRM)] and small invariance risk R_{inv} [cf. (7)]. We therefore combine these two terms to propose the constrained learning problem

$$\begin{aligned} P^\star \;=\; \min_{\boldsymbol{\theta}\in\Theta} \quad & \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}} \left[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), y)\right] \\ \text{s. to} \quad & \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}} \left[\max_{g\in\mathcal{G}} \left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)\right]\right] \leq \epsilon . \end{aligned} \tag{CSRM'}$$

Observe that in the constraint in (CSRM') we keep the first term in (7) only. We do that because to reduce (7) in a problem in which the standard risk is being minimized, it suffices to reduce the first term of (7).

The formulation in (CSRM') tackles the two terms forming the invariant risk bound in (7), but instead of combining them directly, it incorporates the data augmentation term as a constraint in the typical statistical risk minimization problem (SRM). This formulation has the advantage that if a solution to the unconstrained problem is feasible, i.e., satisfies the invariance constraint in (CSRM'), the presence of that constraint has no effect on the statistical problem. Still, it can be beneficial when approximating the solution of (CSRM') empirically. We will explore this fact in the next section, where we tackle the practical challenges involved in solving (CSRM').

3.2. From Robustness to an Augmentation Distribution

Solving the maximization in the constraint can be difficult when $\mathcal{G}$ is not finite or $f_{\boldsymbol{\theta}}$ is a deep neural network. Even when the transformation space is low dimensional, as is the case of translations and rotations, the highly non-convex loss landscape of these models makes the maximization over $\mathcal{G}$ challenging (Engstrom et al., 2017). As shown by (Robey et al., 2021a), the maximisation of the loss over transformations can be written as the semi-infinite constrained optimization problem

$$\begin{aligned} \max_{g\in\mathcal{G}} \ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y) = \sup_{\lambda\in\mathcal{L}_+^2} & \int_{\mathcal{G}} \lambda(g)\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)dg \\ \text{s. to} & \int_{\mathcal{G}} \lambda(g)dg = 1. \end{aligned} \tag{8}$$

Notice that the solution of this optimization problem $\lambda^\star(g)$ is a non-negative, normalized function and can therefore be interpreted as a distribution over transformations that depends on the sample point $(\mathbf{x}, y)$ as well as the model $f_{\boldsymbol{\theta}}$. This allows us to re-interpret the maximization over $\mathcal{G}$ as an expectation, i.e.,

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}}\left[\max_{g\in\mathcal{G}} \ell(f_{\boldsymbol{\theta}}\left(g\mathbf{x}\right), y)\right] = \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}}\left[\mathbb{E}_{g\sim\lambda^\star}\left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)\right]\right]. \tag{9}$$

Note that this risk upper bounds the risk under any augmentation distribution defined on $\mathcal{G}$.

Then we can re-write problem (CSRM') replacing its robustness constraint with one using data augmentation according to $\lambda^\star$, namely,

$$\begin{aligned} P^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}}\left[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), y)\right] \qquad \text{(CSRM)} \\ \text{s. to} \quad & \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}}\left[\mathbb{E}_{g\sim\lambda^\star}\left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)\right]\right] \leq \epsilon\,. \end{aligned}$$

For conciseness in (CSRM) we have included only one invariance constraint associated with a single set of transformations $\mathcal{G}$. However, our approach can be extended to an arbitrary number of constraints defined by transformation sets $\mathcal{G}_i,\ i = 1, \ldots, m$ (each inducing an augmentation distribution $\lambda_i^\star$), and constraint levels ϵ_i. All of the following derivations still hold, regardless of the number of constraints.

4. Algorithm Development

Solving (CSRM) presents two challenges. First, it is a constrained statistical learning problem, which involves the unknown data distribution $\mathfrak{D}$. We address this by resorting to an empirical dual problem as explained in Section 4.1. Second, it can be hard to sample from $\lambda^\star$. We address this by introducing a smooth approximation that leverages MCMC methods in Section 4.2.

4.1. Dual Empirical Constrained Learning

To tackle the invariance-constrained statistical risk minimization problem, we leverage recent duality results in constrained learning theory (Chamon and Ribeiro, 2020; Chamon et al., 2021), that approximate (CSRM) by its empirical dual,

$$D^\star_{\text{emp}} = \max_{\gamma\geq 0}\min_{\boldsymbol{\theta}\in\Theta} \frac{1}{n}\sum_{i=1}^{n}\big[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i) + \gamma\left(\mathbb{E}_{g\sim\lambda^\star}\left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}_i), y_i)\right] - \epsilon\right)\big]. \tag{DE-CRM}$$

The advantage of (DE-CRM) is that it is an unconstrained problem that, provided we have enough samples and the parametrization is rich enough, can approximate the constrained statistical problem (CSRM). Namely, the difference between the optimal value of the empirical dual $D^\star_{\text{emp}}$ and the statistical primal $P^\star$, i.e., the empirical duality gap is bounded (Chamon et al., 2021) with high probability.

As in regular data augmentation, we will also approximate the expectation over $\lambda^\star$ by sampling transformations as discussed in Section 4.2. Then, the problem (DE-CRM) becomes an unconstrained deterministic problem that can be solved using the algorithm described in Section 4.3.

Note that finding a Lagrangian minimizer for a fixed value of the dual variable (γ) is equivalent to minimising the risk under a fixed mixture augmentation distributiona penalised or regularised learning objective. However, solving the constrained problem, namely maximising over γ, has fundamental differences.

First, constraints explicit the requirement they represent. While the degree of invariance imposed should depend only on the statistical problem at hand, the value of γ needed to achieve it will depend on the sample size, the parametrization and the learning algorithm. In contrast, constrained learning dynamically adjusts the amount of augmentation — dictated by γ — to a particular learning setup. Second, the optimal dual variable can give information about the trade-off minimising the loss over training samples and satisfying the invariance constraint. In penalised approaches, on the contrary, this trade-off is fixed. Lastly, the aforementioned informativeness and interpretability can facilitate hyper-parameter tuning. The insights gathered from optimal dual variables can be leveraged a posteriori, for instance, to manually choose appropriate transformations, relax the invariance constraint levels, or change the learning setup (e.g., increase the capacity of the model class).

4.2. Sampling towards invariance

If the optimal distribution $\lambda^\star$ is not smooth, it is challenging to sample from it with sufficient accuracy (Homem-de Mello and Bayraksan, 2014). Consequently, obtaining an unbiased estimator of $\mathbb{E}_{g\sim\lambda^\star}\left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)\right]$ may not be possible. Therefore, we add an L_2 norm penalisation, which promotes smoothness, to leverage MCMC methods.

We then define the *c-smoothed distribution* $\lambda_c^\star$ as a solution to the regularised problem

$$\begin{aligned} \lambda_c^\star \subseteq & \\ & \operatorname{argmax}_{\lambda\in\mathcal{L}_+^2} \int_{\mathcal{G}} \lambda(g)\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)dg + c\int_{\mathcal{G}} \lambda(g)^2 dg, \\ & \text{s. to} \int_{\mathcal{G}} \lambda(g)dg = 1 \end{aligned}$$

The regularization term introduces an optimality gap with respect to worst case perturbations, i.e., $\mathbb{E}_{g\sim\lambda_c^\star}\left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}),y)\right] \leq \max_{g\in\mathcal{G}} \ell\left(f_{\boldsymbol{\theta}}\left(g\mathbf{x}\right),y\right)$. However, for particular values of c the regularized problem has a closed form solution (Robey et al., 2021a) that allows us to sample from it easily. Namely, there exists a constant $c \geq 0$ such that $\lambda_c^\star(\mathbf{x},y,g) \propto \ell(f_{\boldsymbol{\theta}}(g\mathbf{x}),y)$.

We need not find its partition function to sample from $\lambda_c^\star$, and can instead leverage Monte Carlo Markov Chain methods (MCMC). MCMC methods (Hastings, 1970) are based on constructing a Markov chain that has the target distribution as an equilibrium distribution. *Independent* Metropolis Hastings uses a state independent — usually fixed — proposal for each step. In our case, it only requires applying a transformation and computing a forward pass of the neural network to evaluate the loss. This enables the use of non-differentiable transformations, and has the advantage that the density at consecutive proposals can be evaluated in parallel, allowing speedups in the sampling step.

MH methods thus allow to sample the proposal distribution with low computational cost. It also allows the computation of transformations on CPU, unlike gradient based methods, thus requiring minimal modifications to typical training pipelines. Although MH methods exhibit random walk behaviour, which leads to slow convergence in high dimensional settings (Dellaportas and Roberts, 2003; Holden et al., 2009), the space of augmentation transformations is typically low dimensional. For example, $d = 18$ for the experiments in Section 5. Furthermore, even coarse sampling approximations can yield useful augmentation distributions in practice (see Section 5).

However, nothing precludes our method from being used with other sampling algorithms. For example, Hamiltonian Monte Carlo methods (see e.g. (Neal, 2011)), namely Langevin Monte Carlo (Rossky et al., 1978; Bubeck et al., 2015) samplers, are suitable when using differentiable transformations.

Then, we can obtain a set of m samples drawn from $\lambda_c^\star$ and approximate the expectation over the group by the sample mean

$$\mathbb{E}_{g\sim\lambda_c^\star}\left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}_i),y_i)\right] \approx \frac{1}{m}\sum_{j=1}^{m} \ell(f_{\boldsymbol{\theta}}(\mathbf{g_j}\mathbf{x_i}),y_i),$$

where $g_1,\ldots,g_m \overset{\text{i.i.d.}}{\sim} \ell(f_{\boldsymbol{\theta}}(g\mathbf{x}_i),y_i)$ are m transformations sampled from the smoothed distribution $\lambda_c^\star(f_{\boldsymbol{\theta}},\mathbf{x}_i,y_i)$.

Algorithm 1 describes the implementation of independent-MH with a uniform proposal. By keeping only one sample ($m = 1$) we recover the usual augmentation setting, that yields one augmentation per sample in the training batch. In our experiments we address this setting, because it is extensively used in practice. However, simply keeping more samples from the chain ($m > 1$) allows to extend the method to the batch augmentation setting (Hoffer et al., 2020), which creates several augmented samples from the same instance in each batch.

Algorithm 1 Independent MH sampler

$g^{(0)} \sim \mathcal{U}(\mathcal{G})$ ▷ Sample initial State
$\ell^{(0)} = \ell\left(f_{\boldsymbol{\theta}}\left(g^{(0)}\mathbf{x}\right),y\right)$ ▷ Evaluate loss
for $t = 1,\ldots,n_{steps}$ **do**
  $g_{\text{prop}} \sim \mathcal{U}(\mathcal{G})$ ▷ Sample next proposal
  $\ell_{\text{prop}} = \ell\left(f_\theta\left(g_{\text{prop}}\mathbf{x}\right),y\right)$ ▷ Evaluate Loss
  $p = \min\left(1, \frac{\ell_{\text{prop}}}{\ell^{(t-1)}}\right)$ ▷ Acceptance Prob
  w.p. p: ▷ Accept/Reject
    $g^{(t)} = g_{\text{prop}}$, $\ell^{(t)} = \ell_{\text{prop}}$
  else:
    $g^{(t)} = g^{(t-1)}$, $\ell^{(t)} = \ell^{(t-1)}$

4.3. Primal-Dual Algorithm

Since the cost of the inner minimization, i.e. training the model, can be high, we adopt an alternating update scheme (K. J. Arrow and Uzawa., 1958) for the primal and dual variables, as in (Chamon et al., 2021; Fioretto et al., 2020).

A bounded empirical duality gap *does not* guarantee that the primal variables obtained after running the alternating primal-dual Algorithm 2 and solving the saddle point problem approximately are near optimal or approximately feasible. Although stronger *primal recovery* guarantees can be obtained by randomizing the learning algorithm (Chamon et al., 2021), it requires storing model parameters $\boldsymbol{\theta}$ at each iteration and there is empirical evidence (Chamon et al., 2021; Robey et al., 2021a; Elenter et al., 2022; Shen et al., 2022; Cervino et al., 2022; Zhang et al., 2022) that good solutions can still be obtained without randomization.

Algorithm 2 Primal-Dual Augmentation

$\lambda = 0,\ \boldsymbol{\theta} = \boldsymbol{\theta}_0$.
for Batch in $\{(\mathbf{x}_i,y_i)\}_{i=1}^n$ **do**
  for $(\mathbf{x}_i,y_i) \in$ Batch **do** ▷ Sample transformations
    $g_{i1},\ldots,g_{im} \sim^{iid} \ell(f_{\boldsymbol{\theta}}(g\mathbf{x}_i),y_i)$
  $\ell_{\text{c}} = \frac{1}{|\text{Batch}|}\sum_{(\mathbf{x}_i,y_i)\in\text{Batch}}\left[\frac{1}{m}\sum_{j=1}^m \ell(f_{\boldsymbol{\theta}}(g_{ij}\mathbf{x}_i),y_i)\right]$
  $s = \ell_{\text{c}} - \epsilon$ ▷ Slack
  $\ell = \frac{1}{|\text{Batch}|}\sum_{(\mathbf{x}_i,y_i)\in\text{Batch}} \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i),y_i)$ ▷ Loss
  $\hat{L} = \ell + \gamma s$ ▷ Lagrangian
  $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta_p \nabla_{\boldsymbol{\theta}}\hat{L}$ ▷ Primal update
  $\gamma = [\gamma + \eta_d s]_+$ ▷ Dual Update

	Standard			Wide		
	TA	DeepAA	**OURS**	TA	DeepAA	**OURS**
CIFAR10						
Wide-ResNet-40-2	96.55 ± 0.11	96.43 ± 0.09	$\mathbf{96.76 \pm 0.14}$	96.24 ± 0.19	96.27 ± 0.19	$\mathbf{97.05 \pm 0.18}$
Wide-ResNet-28-10	97.46 ± 0.10	97.57 ± 0.15	$\mathbf{97.74 \pm 0.10}$	97.51 ± 0.20	97.27 ± 0.10	$\mathbf{97.85 \pm 0.17}$
CIFAR100						
Wide-ResNet-40-2	79.92 ± 0.13	79.45 ± 0.42	$\mathbf{80.83 \pm 0.31}$	79.96 ± 0.45	79.36 ± 0.27	$\mathbf{81.19 \pm 0.34}$
Wide-ResNet-28-10	83.40 ± 0.16	$\mathbf{83.77 \pm 0.29}$	83.53 ± 0.16	84.11 ± 0.24	83.09 ± 0.30	$\mathbf{84.89 \pm 0.12}$
SVHNcore						
Wide-ResNet-28-10	98.05 ± 0.03	98.04 ± 0.08	$\mathbf{98.15 \pm 0.09}$	$\mathbf{98.07 \pm 0.03}$	97.93 ± 0.03	98.01 ± 0.04

Table 1. Image Classification accuracy for WideResnet architectures (Zagoruyko and Komodakis, 2016) trained using different augmentation policies, defined on standard (Cubuk et al., 2018) and wider (Müller and Hutter, 2021) augmentation search spaces. We include state-of-the-art methods TA (Müller and Hutter, 2021), DeepAA (Zheng et al., 2022), and 95% confidence intervals computed over five independent runs.

5. Experiments

5.1. Automatic Data Augmentation

This section showcases Algorithm 2 in common image classification benchmarks. We compare it to state-of-the-art data augmentation methods in terms of classification accuracy. Furthermore, we discuss other advantageous properties of our method through ablations. Namely, we focus on the ability to control the effect of data augmentation by modifying the constraint level, the informativeness of dual variables, and the benefits of adapting the augmentation distribution throughout training. We follow the setup (including the transformation sets) used in recent automatic data augmentation literature (Müller and Hutter, 2021). A complete list of transformations together with other hyperparameters and training settings can be found on Appendix C.1. Note that four out of the sixteen transformations used are non-differentiable. Whereas other works have introduced gradient approximations for transformation operations with discrete parameters (Li et al., 2020b; Hataya et al., 2020), our approach does not require such approximations.

Throughout these experiments, we fixed the number of steps of the MH sampler (Algorithm 1) to two, which has the added advantage of reducing the computational cost of evaluating proposals. That is, the added computational cost of running our algorithm with respect to uniform augmentation at each iteration is essentially that of computing an additional forward pass through the neural network (see Appendix D.2 for a runtime analysis). The constraint level was determined by a grid search targeting cross-validation accuracy. As shown in Table 1, in both transformation sets considered, we find that our approach improves or closely matches existing approaches.

The failure to achieve large improvements in accuracy over baselines, which has been attributed to a stagnation in data augmentation research (Müller and Hutter, 2021), can also reflect the limits of the benchmarking setup. In order to investigate whether larger gains can be obtained when less data is available and tasks are more challenging, we test our approach on subsampled versions of the Imagenet-100 dataset. That is, we consider all of the classes in Imagenet-100 but use only a fraction of all the training samples in the dataset to train the model. As sown in Table 2, our method shows larger performance gains in this scenario, regardless of the size of the training set.

Samples	TA	**OURS**
0.01	19.17 ± 0.38	$\mathbf{20.11 \pm 0.69}$
0.025	34.67 ± 0.34	$\mathbf{37.85 \pm 0.23}$
0.05	48.77 ± 0.46	$\mathbf{51.01 \pm 1.74}$
0.1	60.77 ± 0.47	$\mathbf{62.45 \pm 0.58}$

Table 2. Image Classification accuracy for subsampled ImageNet-100 using Resnet50 (He et al., 2015) and wide (Müller and Hutter, 2021) augmentation search space. We vary the fraction of samples kept from the original training set (first column). We include TA (Müller and Hutter, 2021) as a baseline and standard deviation computed over three independent runs.

Moreover, our approach yields improvements in test accuracy over a baseline model without augmentation for a wide range of constraint levels (Figure 1). This illustrates the robustness of the solution to this hyperparameter. Observe also that as the constraint is relaxed (by increasing ϵ), the training error decreases while the generalization gap, i.e., the difference between train and test errors, increases. In other words, by loosening the invariance requirement the model can fit better to training samples at the cost of worse generalization. Eventually, only the generalization gap increases while the training error stagnates ($\epsilon > 2.1$ for CIFAR100 and $\epsilon > 0.8$ for CIFAR10). This transition occurs at the same point at which the final value of the dual variable γ from (DE-CRM) essentially vanishes (darker color). This showcases the infromativeness of the dual variable. In the case of CIFAR10, even the training error begins to increase at that point, suggesting that the invariance requirement

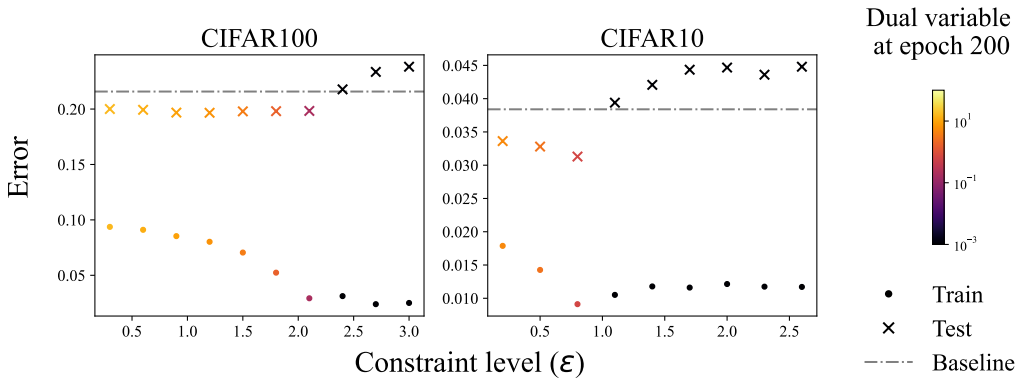

Figure 1. Constraint level ablation for WideResnet-40-2 in CIFAR datasets. We plot error rates computed over the train and test set and averaged over five runs, for different constraint levels. We include the test error of the unconstrained baseline (without augmentation) for comparison. The color of markers denotes the final value of the dual variable.

need not be at odds with accuracy.

Not only does Algorithm 2 tune the effect of data augmentation on the solution (by adapting γ in step 9), but also modifies the distribution over transformations during training (step 4). To showcase the benefits of this over the use of a fixed distribution, Figure 2 compares the results obtained using our approach (sampling according to $\lambda_c^\star$) and one where step 4 is replaced by a uniform sampling over transformations. For the same constraint levels, lower test errors are obtained by sampling transformations according to $\lambda_c^\star$, i.e., promoting invariance. Note also that for $\epsilon = 2.1$ in CIFAR100 and $\epsilon = 0.8$ for CIFAR10, the performance gap is quite large. Once again, this occurs at the point in which γ vanishes (darker color) for the uniform distribution, i.e., no data augmentation occurs by the end of training. At this point, however, there is still value in promoting invariance by sampling from $\lambda_c^\star$ as evidenced by the positive value of the dual variable (lighter color) in this approach.

Furthermore, to assess the impact of the sampling approximation on the performance of the solution we conduct an ablation on the number of steps of the MH sampler (Algorithm 1), keeping the constraint level (ϵ) fixed. Using more steps of the chain allows samples to deviate further from the uniform distribution, which as shown in Figure 3 is reflected on the lower entropy of sampled transformations (right). As shown in the left plot, this results in higher loss on the training set (represented by blue circular markers). However, we find that increasing the number of steps does not affect significantly test loss (denoted by orange crosses). In Appendix D.1.3 we show how the number of sampling steps affects the evolution of dual variables. In Appendix D.1.4 we also analyze how it affects the frequency and strength of different transformations.

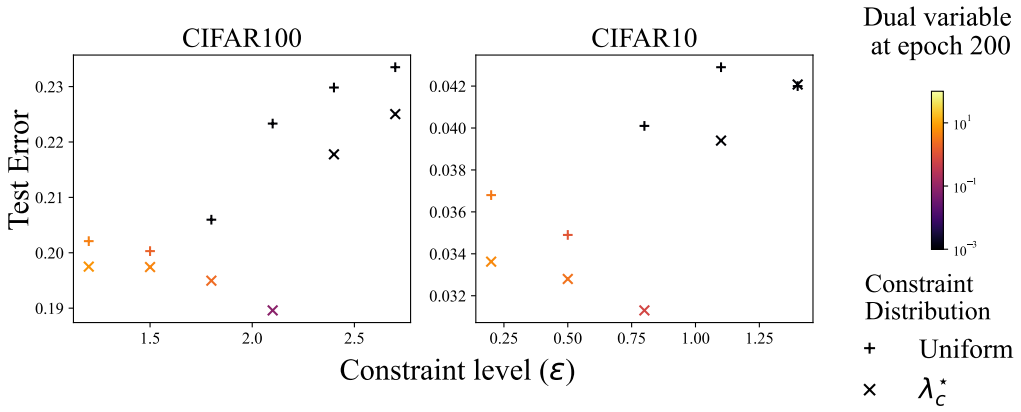

Figure 2. We compare our approach to a constraint on the uniform distribution, for WideResnet-40-2 in CIFAR datasets, at different constraint levels. We plot error rates computed over test set and averaged over five runs. Markers denote the augmentation distribution. The color of markers denotes the final value of the dual variable.

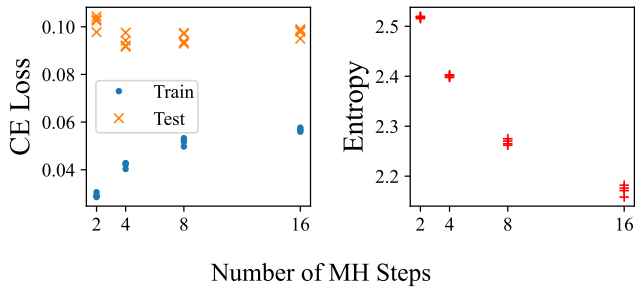

Figure 3. Number of Metropolis Hastings steps ablation for WideResnet-40-2 in CIFAR-10. The constraint level is fixed ($\epsilon = 0.8$). The first plot shows the cross-entropy loss averaged over the train and test sets. The second plot shows the entropy of the augmentations sampled at the last epoch of training. Each point represents an independent run.

5.2. Synthetic Invariances

Although our approach does not aim to learn the set of symmetries or invariant transformations from the data, but rather to impose it on the predictor, dual variables can be used to gather insights on the actual invariances underlying a learning task. We showcase this on datasets with artificial invariances, following the setup of (Immer et al., 2022). Namely, we apply rotations, translations or scalings, independently drawn from the uniform distributions, to each sample in the MNIST (LeCun et al., 2010) and FashionMNIST (Xiao et al., 2017) datasets. We use the same MLP and CNN architectures and hyperparameters as (Immer et al., 2022). The distributions used to generate the datasets and additional experimental details can be found in Appendix C.2.

We run our algorithm constraining the loss on transformation spaces which (except for the fully rotated dataset) are larger than the true transformation range used to construct the synthetic dataset. Note that we use the same transformation sets and constraint levels (ϵ) for all synthetic datasets, regardless of which transformations were used to construct the dataset. As shown in table 3, except for scalings in FashionMNIST, the dual variables (γ) associated with transformations corresponding to the true synthetic invariances in the dataset are considerably smaller. This indicates that when the transformations in the constraint correspond to a true invariance of the dataset, the constraint is easier to satisfy. Therefore dual variables can give information about the invariances in a dataset or alert about the misspecification of transformation sets and constraint levels. Appendix D.5 illustrates how dual variables can be leveraged to adjust the constraint specification (ϵ) using simple heuristics.

6. Conclusion

In this paper, we have proposed a constrained learning approach for automatic data augmentation, which instead of using augmented samples as a modified learning objective, imposes an invariance-constraint. We have shown that this yields an augmentation distribution that adapts during training, and found that coarse sampling approximations based on MCMC methods exhibit competitive performance in small and medium scale benchmarks. Furthermore, our experiments showed that dual variables can give insights about the resulting augmentation distribution. We also found that strictly feasible solutions were obtained for a wide range of constraint levels, with different generalization gaps, and that in some cases tightening the constraint even led to a lower training error. Analysing the interplay between the learning problem and the optimization algorithm, and evaluating our approach in more complex or larger scale datasets, as well as in batch-mode augmentation, are promising future work directions. Finally, constrained optimization can enable further theoretical developments in data augmentation.

			Synthetic Invariance			
Dataset	Architecture	γ	Full Rot.	Partial Rot.	Translation	Scale
MNIST	MLP	Rotation	**0.000**	**0.004**	3.224	0.035
		Translation	1.344	0.038	**0.289**	0.032
		Scale	1.800	0.045	4.206	**0.004**
	CNN	Rotation	**0.000**	**0.002**	2.724	0.012
		Translation	1.218	0.009	**0.439**	0.006
		Scale	2.026	0.049	4.029	**0.003**
F-MNIST	MLP	Rotation	**0.000**	**0.037**	4.470	1.599
		Translation	3.572	1.934	**0.939**	**0.717**
		Scale	4.144	2.653	3.472	0.754
	CNN	Rotation	**0.000**	**0.107**	3.301	1.352
		Translation	3.572	1.426	**0.515**	**0.441**
		Scale	4.144	2.332	2.725	0.904

Table 3. Value of dual variables (after 400 epochs) for different transformation constraints and synthetic invariant datasets. Columns correspond to different transformations of the dataset, and rows to dual variables associated with different transformations. Except for scaling in FashionMNIST, for all architectures and datasets the dual variable associated with the constraint corresponding to the transformations applied to the dataset is considerably lower. The smallest dual variable for each dataset and architecture is bolded.

Acknowledgements.

The work of A. Ribeiro and I. Hounie is supported by NSF-Simons MoDL, Award 2031985, NSF AI Institutes program, Award 2112665, and NSF HDR TRipods Award 1934960. The work of Dr. Chamon is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy (EXC 2075-390740016).

References

Benton, G., Finzi, M., Izmailov, P., and Wilson, A. G. (2020). Learning invariances in neural networks from training data. *Advances in Neural Information Processing Systems*, 33:17605–17616.

Bertsekas, D. (2015). *Convex optimization algorithms*. Athena Scientific.

Bietti, A. and Mairal, J. (2019). Group invariance, stability to deformations, and complexity of deep convolutional representations. *The Journal of Machine Learning Research*, 20(1):876–924.

Blaas, A., Suau, X., Ramapuram, J., Apostoloff, N., and Zappella, L. (2021). Challenges of adversarial image augmentations. In *I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop*.

Bubeck, S., Eldan, R., and Lehec, J. (2015). Finite-time analysis of projected langevin monte carlo. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Caruana, R., Lawrence, S., and Giles, C. (2000). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Cataltepe, Z., Abu-Mostafa, Y. S., and Magdon-Ismail, M. (1999). No free lunch for early stopping. *Neural Computation*, 11(4):995–1009.

Cervino, J., Ruiz, L., and Ribeiro, A. (2022). Training stable graph neural networks through constrained learning. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4223–4227.

Chamon, L. F. O., Paternain, S., Calvo-Fullana, M., and Ribeiro, A. (2021). Constrained learning with non-convex losses. *CoRR*, abs/2103.05134.

Chamon, L. F. O. and Ribeiro, A. (2020). Probably approximately correct constrained learning. *CoRR*, abs/2006.05487.

Chapelle, O., Weston, J., Bottou, L., and Vapnik, V. (2000). Vicinal risk minimization. *Advances in neural information processing systems*, 13.

Chatzipantazis, E., Pertigkiozoglou, S., Dobriban, E., and Daniilidis, K. (2021). Learning augmentation distributions using transformed risk minimization. *arXiv preprint arXiv:2111.08190*.

Chen, S., Dobriban, E., and Lee, J. H. (2019). A group-theoretic framework for data augmentation.

Cheung, T.-H. and Yeung, D.-Y. (2022). Adaaug: Learning class- and instance-adaptive data augmentation policies. In *International Conference on Learning Representations*.

Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA. PMLR.

Cubuk, E. D., Zoph, B., Mané, D., Vasudevan, V., and Le, Q. V. (2018). Autoaugment: Learning augmentation policies from data. *CoRR*, abs/1805.09501.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Dellaportas, P. and Roberts, G. O. (2003). An introduction to mcmc. In *Spatial statistics and computational methods*, pages 1–41. Springer.

DeVries, T. and Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Duvenaud, D., Maclaurin, D., and Adams, R. (2016). Early stopping as nonparametric variational inference. In Gretton, A. and Robert, C. C., editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1070–1077, Cadiz, Spain. PMLR.

Elenter, J., Naderializadeh, N., and Ribeiro, A. (2022). A lagrangian duality approach to active learning. *ArXiv*, abs/2202.04108.

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. (2017). Exploring the landscape of spatial robustness.

Esteves, C., Allen-Blanchette, C., Makadia, A., and Daniilidis, K. (2017). 3d object classification and retrieval with spherical cnns. *CoRR*, abs/1711.06721.

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. (2020). Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data.

Fioretto, F., Hentenryck, P. V., Mak, T. W., Tran, C. D., Baldo, F., and Lombardi, M. (2020). Lagrangian duality for constrained deep learning. In *ECML/PKDD*.

Fukushima, K. and Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Hataya, R., Zdenek, J., Yoshizoe, K., and Nakayama, H. (2020). Faster autoaugment: Learning augmentation strategies using backpropagation. In *European Conference on Computer Vision*, pages 1–16. Springer.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Ho, D., Liang, E., Chen, X., Stoica, I., and Abbeel, P. (2019). Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741. PMLR.

Hoffer, E., Ben-Nun, T., Hubara, I., Giladi, N., Hoefler, T., and Soudry, D. (2020). Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138.

Holden, L., Hauge, R., and Holden, M. (2009). Adaptive independent Metropolis–Hastings. *The Annals of Applied Probability*, 19(1):395 – 413.

Homem-de Mello, T. and Bayraksan, G. (2014). Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19(1):56–85.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Immer, A., van der Ouderaa, T. F. A., Fortuin, V., Rätsch, G., and van der Wilk, M. (2022). Invariance learning in deep neural networks with differentiable laplace approximations. *CoRR*, abs/2202.10638.

Jebara, T. (2003). Convex invariance learning. In Bishop, C. M. and Frey, B. J., editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume R4 of *Proceedings of Machine Learning Research*, pages 149–156. PMLR. Reissued by PMLR on 01 April 2021.

Ji, Z., Li, J. D., and Telgarsky, M. (2021). Early-stopped neural networks are consistent. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

Joshi, A., Mukherjee, A., Sarkar, S., and Hegde, C. (2019). Semantic adversarial attacks: Parametric transformations that fool deep classifiers.

Jumper, J. M., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D. A., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596:583 – 589.

K. J. Arrow, L. H. and Uzawa., H. (1958). *Studies in linear and non-linear programming*. Stanford University Press.

Kanbak, C., Moosavi-Dezfooli, S.-M., and Frossard, P. (2017). Geometric robustness of deep networks: analysis and improvement.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kondor, R. and Trivedi, S. (2018). On the generalization of equivariance and convolution in neural networks to the action of compact groups.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, Canadian Institute for Advanced Research.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Li, M., Soltanolkotabi, M., and Oymak, S. (2020a). Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pages 4313–4324. PMLR.

Li, Y., Hu, G., Wang, Y., Hospedales, T., Robertson, N. M., and Yang, Y. (2020b). Dada: differentiable automatic data augmentation. *arXiv preprint arXiv:2003.03780*.

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. (2019). Fast autoaugment. *Advances in Neural Information Processing Systems*, 32.

LingChen, T. C., Khonsari, A., Lashkari, A., Nazari, M. R., Sambee, J. S., and Nascimento, M. A. (2020). Uniformaugment: A search-free probabilistic data augmentation approach. *ArXiv*, abs/2003.14348.

Liu, A., Huang, Z., Huang, Z., and Wang, N. (2021). Direct differentiable augmentation search. *CoRR*, abs/2104.04282.

Lyle, C., van der Wilk, M., Kwiatkowska, M., Gal, Y., and Bloem-Reddy, B. (2020). On the benefits of invariance in neural networks.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mallat, S. (2016). Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203.

Müller, S. G. and Hutter, F. (2021). Trivialaugment: Tuning-free yet state-of-the-art data augmentation. *CoRR*, abs/2103.10158.

Neal, R. M. (2011). Mcmc using hamiltonian dynamics. *arXiv: Computation*, pages 139–188.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems (NIPS)*.

Olah, C., Cammarata, N., Voss, C., Schubert, L., and Goh, G. (2020). Naturally occurring equivariance in neural networks. *Distill*. https://distill.pub/2020/circuits/equivariance.

Petrini, L., Favero, A., Geiger, M., and Wyart, M. (2021). Relative stability toward diffeomorphisms indicates performance in deep nets. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

Püschel, M. and Moura, J. M. F. (2006). Algebraic signal processing theory. *CoRR*, abs/cs/0612077.

Rice, L., Wong, E., and Kolter, Z. (2020). Overfitting in adversarially robust deep learning. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8093–8104. PMLR.

Robey, A., Chamon, L. F. O., Pappas, G. J., Hassani, H., and Ribeiro, A. (2021a). Adversarial robustness with semi-infinite constrained learning. *ArXiv*, abs/2110.15767.

Robey, A., Pappas, G. J., and Hassani, H. (2021b). Model-based domain generalization. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20210–20229. Curran Associates, Inc.

Rockafellar, R. and Wets, R. J.-B. (1998). *Variational Analysis*. Springer Verlag, Heidelberg, Berlin, New York.

Rossky, P. J., Doll, J. D., and Friedman, H. L. (1978). Brownian dynamics as smart monte carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Sannai, A., Imaizumi, M., and Kawano, M. (2019). Improved generalization bounds of group invariant / equivariant deep networks via quotient feature spaces.

Shao, H., Montasser, O., and Blum, A. (2022). A theory of pac learnability under transformation invariances.

Shen, Z., Cervino, J., Hassani, H., and Ribeiro, A. (2022). An agnostic approach to federated learning with class imbalance. In *International Conference on Learning Representations*.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S. (2018). Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31.

Wang, H., Huang, Z., Wu, X., and Xing, E. P. (2022). Toward learning robust and invariant representations with alignment regularization and data augmentation.

Weiler, M. and Cesa, G. (2019). General e(2)-equivariant steerable cnns. *CoRR*, abs/1911.08251.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

Xu, Y., Noy, A., Lin, M., Qian, Q., Hao, L., and Jin, R. (2021). Wemix: How to better utilize data augmentation.

Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *CoRR*, abs/1605.07146.

Zhang, H., Zhang, Y.-F., Liu, W., Weller, A., Schölkopf, B., and Xing, E. P. (2022). Towards principled disentanglement for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8024–8034.

Zhang, X., Wang, Q., Zhang, J., and Zhong, Z. (2020). Adversarial autoaugment. In *International Conference on Learning Representations*.

Zheng, Y., Zhang, Z., Yan, S., and Zhang, M. (2022). Deep autoaugment. In *International Conference on Learning Representations*.

Zhou, A., Knowles, T., and Finn, C. (2021a). Meta-learning symmetries by reparameterization. In *International Conference on Learning Representations*.

Zhou, F., Li, J., Xie, C., Chen, F., Hong, L., Sun, R., and Li, Z. (2021b). Metaaugment: Sample-aware data augmentation policy learning. In *AAAI*.

A. Additional Related work

A.1. Constrained Learning and Data augmentation

(Xu et al., 2021) also formulate data augmentation as a constrained learning problem. They impose a constraint on the excess risk, i.e. the difference between the statistical risk and its optimal value, on augmented data. Thus the constraint level on the augmented risk is also determined by the data distribution, augmentations considered, and model class, and the existence of a strictly feasible point is guaranteed.

$$\min_{\boldsymbol{\theta}\in\Theta} R(f_{\boldsymbol{\theta}}) \quad \text{s.t.} \quad R_{aug}\left(f_{\boldsymbol{\theta}}\right) - \min_{\hat{\boldsymbol{\theta}}\in\Theta} R_{aug}(f_{\hat{\boldsymbol{\theta}}}) \leq \epsilon,\ \epsilon > 0,$$

where R and R_{aug} are the statistical risk under the original and augmented distribution, exlipicity

$$R(f_{\boldsymbol{\theta}}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}}[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), y)],$$
$$R_{\text{aug}}(f_{\boldsymbol{\theta}}) = \mathbb{E}_{\substack{(\mathbf{x},y)\sim\mathfrak{D},\\ g\sim\mathfrak{G}}}[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)].$$

Unlike our formulation, this formulation assumes a fixed distribution of augmentations $\mathfrak{G}$ is given.

By formulating it as a constrained problem, they aim to avoid introducing a bias when the data distribution is not invariant to augmentations. Two types of biases induced by augmentation are explicitly addressed, covariate shift (i.e. label-preserving augmentations) and concept shift (i.e. label mixing augmentations).

Interestingly, they show that under some conditions on the risk, augmented risk and constraint level, by utilizing the augmented data to constrain the solution to a small region SGD can achieve lower error (Xu et al., 2021, Proposition 1).

Instead of resorting to constrained optimization algorithms, they propose a two stage algorithm that consists of first finding an approximate minimizer of the augmented risk and then using that solution as an initialisation to the (unconstrained) statistical risk minimization problem. The first stage obtains a feasible point, and then under some conditions the SGD iterates obtained when solving the second problem remain feasible (Xu et al., 2021, Theorem 1).

A.2. Adversarial Data Augmentation

(Zhang et al., 2020) have shown that using adversarial transformations - which as already mentioned is related to promoting invariance - can give competitive results with respect to other automatic augmentation methods in image classification. However, (Blaas et al., 2021) have since evidenced the importance of two factors: the implicit learning curricula and the suboptimality of the adversarial used by (Zhang et al., 2020), which mitigates the biases introduced by worst-case transformations. Furthermore, (Blaas et al., 2021) report that an explicit cyclic curricula in which augmentations are *mild* at first, then get harder as training progresses, and finally revert to milder augmentations at the end of training, performs better empirically. We note some interesting commonalities with our approach and experimental results. First, the dynamics of our primal-dual algorithm resemble the aforementioned heuristically defined curricula. Second, the suboptimality with respect to worst case perturbations can be related to the smoothed approximation used in our approach.

A.3. Constrained Learning and Domain Generalization

Domain Generalization (DG) involves training the model in different but related data distributions, and evaluating in an unseen *domain*. For example, a common benchmark consists of domains created by rotating images in the MNIST dataset by different angles. Constrained formulations have been proposed in this context (Robey et al., 2021b; Zhang et al., 2022), that enforce invariance under learnt *domain translation* transformations. In contrast, our approach addresses pre-defined transformations that are commonly used in data augmentation pipelines in order to improve generalization in the same domain used to train the model.

A.4. Group invariance

The relationship between convolutional structure and equivariance has been long known in algebraic signal processing

theory (Püschel and Moura, 2006). Recently, necessity results have re-gained attention in the context of neural network architecture design (Kondor and Trivedi, 2018). Several Group Convolutional neural network architectures that generalize CNNs to different groups by leveraging group convolutions have been proposed (Cohen and Welling, 2016; Esteves et al., 2017; Finzi et al., 2020). In the case of images, it has been shown that that SE2 equivariant layers can be implemented efficiently using regular 2D convolutions (Weiler and Cesa, 2019). This approach allows to derive a parametrization for CNN filters under finite subgroups of SE2. Among other works (Bietti and Mairal, 2019; Sannai et al., 2019) give theoretical analyses of the benefits of group invariance in learning settings.

As already mentioned, achieving invariance through architecture design is both challenging and limited in the sense that it relies on transformations having a specific structure (e.g: a group). The goal of exact invariance over the whole input space is more strict than the approximate invariance notion that our work addresses. As argued by (Mallat, 2016), CNNs can learn *locally* invariant features with respect to arbitrary transformation groups, which could explain their generalization properties. Furthermore, empirical studies evidence modern CNN architectures learn approximately equivariant features to transformations such as scaling and rotations (Olah et al., 2020), or diffeomorphisms (Petrini et al., 2021), even when trained without direct augmentation.

However, commonly used augmentations do not form a group. Our approach does not require this structure.

B. Additional Theoretical Details

B.1. Related Notions of invariance

There are several definitions of invariance that capture different properties of the solution or data distribution under the action of transformations in $\mathcal{G}$. In the context of supervised learning, the data distribution is said to be *exactly invariant* iff it does not change when transformations are applied to the covariates, i.e.,

$$(\mathbf{x}, y) =_d (g\mathbf{x}, y), \quad \text{for all} \quad g \in \mathcal{G}, \tag{10}$$

Note that this is equivalent to the distribution of inputs x being conditionally invariant on the label y,

$$(gX \mid Y = y) =_d (X \mid Y = y). \tag{11}$$

If the data distribution $\mathfrak{D}$ is statistically invariant under the action of $\mathcal{G}$, invariant solutions have provable advantages in terms of sample complexity (Chen et al., 2019; Bietti and Mairal, 2019; Sannai et al., 2019; Lyle et al., 2020).

These notions of invariance do not explicitly contemplate the task at hand, that is, not all changes in y equally affect performance. Thus, we use the loss to encode meaningful differences in labels with respect to the task, as described in section 3.1. Throughout this work, we thus refer to the invariant risk R_{inv} defined on equation 6 as the degree of invariance, unless otherwise noted.

B.2. Data Augmentation and Penalised Formulations

Augmented risk minimisation formulations usually do not include the loss on clean data explicitly on the objective. However, several common choices for the augmentation distribution $\mathfrak{G}$ have positive mass at the identity transformation $e(\mathbf{x}) = \mathbf{x}$, i.e. $P_{\mathfrak{G}}(g = e) = \gamma > 0$.

Therefore, the risk under the augmentation distribution $\mathfrak{G}$ is statistically equivalent to a regularised objective, explicitly

$$\begin{aligned}\mathbb{E}_{\substack{(\mathbf{x},y)\sim\mathfrak{D},\\ g\sim\mathfrak{G}}}[\ell(h(g\mathbf{x}), y)] =& \gamma\mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}}[\ell(h(\mathbf{x}), y)] \\ &+ \mathbb{E}_{\substack{(\mathbf{x},y)\sim\mathfrak{D},\\ g\sim\tilde{\mathfrak{G}}}}[\ell(h(g\mathbf{x}), y)],\end{aligned}$$

where $\tilde{\mathfrak{G}}$ is a distribution over $\mathcal{G}$ with no mass at the identity.

The bias-variance tradeoff arising from averaging over transformations of training samples while introducing a distribution shift (Chen et al., 2019), can thus be controlled through the probability of the identity transformation in the augmentation distribution.

B.3. Worst case transformations as an Augmentation Distribution

In this section we show that finding the transformation that maximises the loss can be interpreted as an augmentation distribution. We thus extend the particular case of bounded additive perturbations in (Robey et al., 2021a) to a transformation set $\mathcal{G}$.

Given a predictor f_θ and a data instance (x, y), we want to solve the optimization problem

$$P^* = \max_{g\in\mathcal{G}} \ell\left(f_{\boldsymbol{\theta}}\left(g\mathbf{x}\right), y\right) \tag{12}$$

Assuming $\ell \in L^2$, we begin by writing the problem in epigraph form

$$\begin{aligned} P^* = \min_{t\in L^2} & \; t(g) \\ \text{s.t. } & \ell(f_\theta(gx), y) \leq t(g) \text{ for all} g \in \mathcal{G} \end{aligned} \tag{EPI}$$

Then, the Lagrangian associated to this problem can be

written as

$$L(\lambda, t) = t(g) + \int_{\mathcal{G}} \lambda(g)(\ell(f_\theta(gx), y) - t(g))dg$$
$$= t(g)\left[1 - \int_{\mathcal{G}} \lambda(g)dg\right]$$
$$+ \int_{\mathcal{G}} \lambda(g)(\ell(f_\theta(g\mathbf{x}), y))dg.$$

Problem (EPI) can thus be written in Lagrangian form as:

$$P^* = \sup_{\lambda \in L^2_+} \min_{t \in L^2} L(\lambda, t),$$

where L^2_+ is the space of non negative functions.

Since L is linear in t, if P^* is finite, then strong duality holds (Bertsekas, 2015, Chapter 4). Then we can solve the dual problem:

$$P^* = D^* := \sup_{\lambda \in \mathcal{P}^q} \min_{t \in L^p} L(\lambda, t),$$

Since t is unconstrained, the inner minimization yields:

$$\min_{t \in L^2} L(\lambda, t) = \begin{cases} -\infty & \text{if } \int_{\mathcal{G}} \lambda(g)dg \neq 1 \\ \int_{\mathcal{G}} \lambda(g)\ell(f_\theta(g\mathbf{x}), y)dg & \text{otherwise} \end{cases}$$

Maximizing over λ we get the desired result:

$$P^* = \sup_{\lambda \in L^2_+} \int_{\mathcal{G}} \lambda(\mathbf{x}, y, g)\ell(f_\theta(g\mathbf{x}), y)dg,$$
$$\text{s. t.} \int_{\mathcal{G}} \lambda(\mathbf{x}, y, g)dg = 1$$

Since any solution of this optimization problem, which we will denote $\lambda^\star(g)$, is non-negative almost everywhere and integrates one, it defines a distribution over the set of transformations $\mathcal{G}$. This allows us to re-interpret the maximization over $\mathcal{G}$ as an expectation, namely

$$\max_{g \in \mathcal{G}} \ell\left(f_{\boldsymbol{\theta}}\left(g\mathbf{x}\right), y\right) = \mathbb{E}_{g \sim \lambda^\star}\left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)\right].$$

B.4. Derivation of the smoothed distribution $\lambda^\star_c$

In this section we show that for a particular value of c, the smoothed distribution $\lambda^\star_c$ has a closed form solution that enables MCMC sampling methods. We defined the $\mathcal{L}^2$ regularised problem as

$$\sup_{\lambda \in \mathcal{L}^2_+} \int_{\mathcal{G}} \left[\lambda(\mathbf{x}, y, g)\ell(h(g\mathbf{x}), y) + c\lambda^2(\mathbf{x}, y, g)\right] dg.$$
$$\text{s . t.} \int_{\mathcal{G}} \lambda(\mathbf{x}, y, g)dg = 1$$

We will begin by finding the optima of the unconstrained problem.

Since $\mathcal{L}^2$ is decomposable, and the integral is Careothodory, the supremum can be found by taking pointwise supremum in $\mathbb{R}_+$ inside the integral. See (Rockafellar and Wets, 1998) chapter 14, theorem 14.60.

$$\sup_{\lambda \in \mathcal{L}^2_+} \int_G \left[\lambda(\mathbf{x}, y, g)\ell(f_\theta(gx), y)\right.$$
$$\left. - c\int_{\mathcal{G}} \lambda^2(\mathbf{x}, y, g)\right]dg$$
$$= \int_{\mathcal{G}} \sup_{\lambda(\mathbf{x}, y, g) \in \mathcal{R}_+} \left[\lambda(\mathbf{x}, y, g)\ell(f_\theta(gx), y)\right.$$
$$\left. - c\lambda^2(\mathbf{x}, y, g)\right]dg$$

Note that if $\ell(f_\theta(gx), y)) < 0$ then the pointwise maximum is attained at $\lambda^*(x, y, g) = 0$.

Otherwise, we can find the pointwise maxima by setting the partial derivative w.r.t. λ to zero.

$$\frac{\partial}{\partial \lambda(\mathbf{x}, y, g)}\left[\lambda(\mathbf{x}, y, g)\ell(f_\theta(gx), y) - c\lambda^2(\mathbf{x}, y, g)\right]$$
$$= \ell(f_\theta(gx), y) - 2c\lambda(\mathbf{x}, y, g)$$

Then, if $\ell(f_\theta(gx), y)) \geq 0$ the pointwise maximum is attained at $\lambda^*(x, y, g) = \frac{\ell(f_\theta(gx), y)}{2c}$. Then the optimum of the unconstrained problem can be written as

$$\lambda^*(x, y, g) = \left[\frac{\ell(f_\theta(gx), y)}{2c}\right]_+$$

For the solution of the unconstrained problem λ^* to be feasible for the constrained problem, we can choose c so that it satisfies the normalization constraint:

$$\int_{\mathcal{G}} \lambda^*(\mathbf{x}, y, g)dg = \frac{1}{2c}\int_{\mathcal{G}} \left[\ell(f_\theta(gx), y)\right]_+ dg = 1$$
$$\Leftrightarrow$$
$$c = \frac{1}{2}\int_{\mathcal{G}} \left[\ell(f_\theta(gx), y)\right]_+ dg$$

If we further assume the loss is non-negative, which holds for many commonly used losses such as cross-entropy and mean squared error, we can drop the projection into the non-negative orthant.

B.5. Constrained Statistical Learning Guarantees

In this section we provide an overview of the results that bound the empirical duality gap of problem (DE-CRM) under uniform convergence assumptions shown by (Chamon and Ribeiro, 2020; Chamon et al., 2021) in more general settings and in (Robey et al., 2021a) for the particular case of robustness (invariance) objectives. The bound goes to show that the sample complexity of constrained learning is not

larger than that of unconstrained ERM. However, unlike the unconstrained formulation of traditional data augmentation in (2), our constrained formulation has stronger guarantees over the loss of transformed samples.

A1. (Bounded Distance to Convex Hull) The parametrization f_θ is rich enough so that for each $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta$ and $\beta \in [0,1]$, there exists $\theta \in \Theta$ such that $\sup_{\mathbf{x}\in\mathcal{X}} |\beta f_{\theta_1}(\boldsymbol{x}) + (1-\beta) f_{\theta_2}(\boldsymbol{x}) - f_\theta(\boldsymbol{x})| \leq \alpha$.

A2.(Slater's Condition) There exists $\boldsymbol{\theta}' \in \Theta$ such that $\mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{D}} \left[\mathbb{E}_{g\sim\lambda^\star} \left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}_n), y_n)\right]\right] - \epsilon > M\nu > 0$.

A3.(Uniform convergence) There exists $\zeta_0(N,\delta), \zeta_1(N,\delta) \geq 0$ monotonically decreasing in N such that w.p $1-\delta$ for all$\theta \in \Theta$:

$$\left|\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}} \left[\mathbb{E}_{g\sim\lambda^\star} \left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)\right]\right] - \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{g\sim\lambda^\star} \left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}_n), y_n)\right]\right| \leq \zeta_1(N,\delta)$$

$$\left|\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}} \left[\ell\left(f_{\boldsymbol{\theta}}(\mathbf{x}), y\right)\right] - \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}_n\right), y_n\right)\right| \leq \zeta_0(N,\delta).$$

A4. The loss function $\ell(\cdot, y)$ is convex, M-Lipchitz and $[0, B]$ bounded.

The first two assumptions state that the hypothesis space $\mathcal{H}_{\boldsymbol{\theta}} = \{f_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p\}$ is sufficiently expressive in the sense that the pointwise distance to its convex hull $\text{conv}(\mathcal{H}_{\boldsymbol{\theta}})$ is bounded (A1), and there exists a function that $f_\theta \in \mathcal{F}$ that strictly satisfies the constraint (A2). Note that in the case of neural networks, universal approximation results (Hornik et al., 1989) imply the existence of hypothesis spaces that are sufficiently expressive to satisfy these assumptions. However, this may be at odds with uniform convergence (A3), which limits the complexity of the hypothesis space.

Empirical Duality Gap Bound (Chamon et al., 2021) Under Assumptions $A1 - A4$, there exists a solution $(\gamma^*, f_{\hat{\boldsymbol{\theta}}^\star})$ of the empirical dual problem (DE-CRM) such that, w. p. $1 - 5\delta$, it holds

$$\left|P^\star - D^\star_{\text{emp}}\right| \leq (1+\overline{\gamma})(M\nu + \max(\zeta_0(N,\delta), \zeta_1(N,\delta)), \text{ and } \mathbb{E}_{\mathfrak{D}} \left[\mathbb{E}_{g\sim\lambda^\star} \left[\ell(f_{\boldsymbol{\theta}}(g\mathbf{x}), y)\right]\right] \leq \epsilon + \zeta_1\left(N,\delta\right),$$

where $P^\star$ is the value of (CSRM), $\bar{\zeta} = \max_i \zeta_i\left(N_i, \delta\right)$, and $\overline{\gamma}$ is lower bounded by γ^*.

The optimal dual variable γ^* depends not only on the data distribution, loss and hypothesis class, but also on the constraint level. As a result, if satisfying the invariance constraint is overly restrictive for a learning task and results in large optimal dual variables, it also leads to looser guarantees for our empirical dual algorithm.

C. Experimental Setup Details

C.1. Automatic Data Augmentation

C.1.1. Transformations

As in recent automatic augmentation literature (Ho et al., 2019; Lim et al., 2019; Hataya et al., 2020; Zhang et al., 2020; Cubuk et al., 2020; LingChen et al., 2020; Zhang et al., 2020; Müller and Hutter, 2021; Zhou et al., 2021b; Zheng et al., 2022; Cheung and Yeung, 2022), we focus on image classification datasets and employ a transformation search space comprising 14 operations, introduced by (Cubuk et al., 2018). For those that have parameters, their magnitudes are discretized in thirty levels, which does not compromise performance and greatly reduces the search space (Cubuk et al., 2020). Most approaches compose transformations, i.e. applying more than one transformation to the same image. However, recently (Müller and Hutter, 2021) have shown that applying only one transformation at a time, defined over a wider magnitude space (noted Wide in Table 4) can outperform other approaches. We thus use the same transformation space as (Müller and Hutter, 2021).

Table 4 from (Müller and Hutter, 2021) lists the operations and their magnitude ranges. In our experiments we used the wide search space, the standard ranges from (Cubuk et al., 2020) are included for comparison. We extend the codebase provided by (Müller and Hutter, 2021), which uses the Pillow[1] implementation of all transformations except for cutout, and refer to its documentation for further details about image operations.

C.1.2. Training Setup

In order to enable comparisons and reproducibility we use the same training pipeline as in previous works (Müller and Hutter, 2021) . We apply the vertical flip and the pad-and-crop augmentations and a 16 pixel cutout (DeVries and Taylor, 2017) after any augmentation method. We trained Wide-ResNet (Zagoruyko and Komodakis, 2016) models in the 40-2 and 28-10 configurations For CIFAR (Krizhevsky, 2009) and SVHN (Netzer et al., 2011) datasets, and ResNet-50 (He et al., 2015) for Imagenet-100 (Russakovsky et al., 2015).

Except for epoch ablation and Imagenet-100 experiments, we use SGD with Nesterov Momentum and a learning rate of 0.1, a batch size of 128, a 5e-4 weight decay. In Imagenet-100 experiments, we use Adam (Kingma and Ba, 2014) and a learning rate of 5e-4. We use a cosine learning rate decay schedule and train for 270 epochs for tinyImagenet and 200 epochs for all other datasets. In ablation experiments we also trained for 600 epochs and used a custom learning rate schedule. For the first 185 epochs we followed the same

[1] https://github.com/python-pillow/Pillow

Operation	Magnitude	
	Standard	Wide
Identity	–	–
ShearX	$[-0.3, 0.3]$	$[-0.99, 0.99]$
ShearY	$[-0.3, 0.3]$	$[-0.99, 0.99]$
TranslateX	$[-10, 10]$	$[-32, 32]$
TranslateY	$[-0.45, 0.45]$	$[-32, 32]$
Rotate	$[-30, 30]$	$[-135, 135]$
AutoContrast	–	–
Invert	–	–
Equalize	–	–
Solarize	$[0, 256]$	
Posterize	$[4, 8]$	$[2, 8]$
Contrast	$[0.1, 1.9]$	$[0.01, 2]$
Color	$[0.1, 1.9]$	$[0.01, 2]$
Brightness	$[0.1, 1.9]$	$[0.01, 2]$
Sharpness	$[0.1, 1.9]$	$[0.01, 2]$
Flips	–	–
Cutout	$16(60)$	$16(60)$
Crop	–	–

Table 4. Pillow image operations in the data augmentation search space and the range of magnitudes corresponding to the standard (Cubuk et al., 2020) and wide (Müller and Hutter, 2021) search spaces. Some operations do not use magnitude parameters.

cosine learning rate decay schedule, and then switching to a custom step learning rate scheduler detailed on Table 5. This schedule was implemented after observing that just scaling the cosine learning rate schedule to 600 epochs resulted in slow convergence, thus yielding solutions similar to training for 600 epochs, and failing reflect the effects of early stopping which this experiment addressed.

We used 2 MH steps for the MH sampler (Algorithm 1) unless stated otherwise, and a learning rate of 10^{-3} for the dual ascent step.

No other hyperparameters were tuned or modified with respect to standard settings. The constraint levels were set for Wide-ResNet-40-2 in the wide augmentation space by maximising three fold cross validation over the grid specified on Table 6. Then the constraint levels were adjusted for other architectures and search spaces so that the dual variables at the end of training were small but not zero (of the order of 10^{-1}), which empirically showed good results. The resulting constraint levels corresponding to the results in 1 are detailed in table 7.

We used the provided code and reported hyperparameters for running TrivialAugment (Müller and Hutter, 2021) and DeepAA (Zheng et al., 2022). For the latter, unlike the results reported in (Zheng et al., 2022), we kept all evaluation hyperparameters (including weight decay) consistent with that of other methods. Results for the wide augmentation space were not included in (Zheng et al., 2022). We thus performed the augmentation policy search for the wide search space using the same hyperparameters (except for the augmentation space) reported in (Zheng et al., 2022) for CIFAR datasets. We also run the search for both augmentation spaces in SVHN datasets, and evaluated the policy with the same setup as other methods.

Epochs	LR Scheduler Step
180-230	10
230-430	20
430-600	40

Table 5. Learning Rate Custom schedule used when training for 600 epochs. We use the standard scheduler for the first 180 epochs, and then update the LR only every n epochs, where n is the number indicated in the second column. This hand designed schedule outperforms using a cosine learning rate schedule for 600 epochs, but could improvements convergence speed and performance by exploring other LR-schedulers or tuning it.

Dataset	Constraint Level Grid Range
CIFAR10	[0.2, 2.3]
CIFAR100	[0.3, 2.7]
SVHNcore	[0.1, 1.5]

Table 6. Constraint level grid search space. For each dataset, we evaluated three fold cross validation accuracy for 8 evenly spaced constraint levels in the ranges given in the second column. The one with the highest cross validation score was then selected and used to train the model with the full dataset.

C.2. Syntetic Invariances

We showcase our approach on datasets with artificial invariances, following the setup of (Immer et al., 2022). Explicitly, we generate the synthetic datasets, by applying either rotations, translations or scalings, to each sample in the MNIST (LeCun et al., 2010) and FashionMNIST (Xiao et al., 2017) datasets. The transformations are sampled from uniform distributions over the ranges detailed in Table 8. We use the same MLP and CNN architectures and hyperparameters as (Immer et al., 2022), except that we use only 1 augmentation per sample instead of 31 during training.

For each transformation set (rotations, translations and scalings), we constraint the expected loss over samples augmented with the transformations sampled from uniform

	Constraint Level	
	Standard	Wide
CIFAR10		
Wide-ResNet-40-2	0.6	0.8
Wide-ResNet-28-10	0.4	0.8
CIFAR100		
Wide-ResNet-40-2	0.9	0.8
Wide-ResNet-28-10	0.9	1.2
SVHNcore		
Wide-ResNet-28-10	0.1	0.2

Table 7. Constraint levels for different datasets and architectures, for the results presented in Table 1

distributions over the ranges detailed in Table 9. Note that there is a mismatch between the distribution used to generate the data and that used in the constraints. That is, except for the fully rotated dataset, the constraints are larger than the true transformation range used to construct the synthetic dataset (Table 8). We use the same transformation sets and constraint specification (ϵ) for all synthetic versions of the datasets. We only vary the constraint level depending on the architecture, to contemplate how model capacity affects the relative difficulty of satisfying the constraint. In order to do so, we evaluate the loss of a model trained without constraints. Constraint levels are specified in Table 10.

Synthetic invariance	Parameter	Distribution
Full Rotation	Angle in radians.	$\mathcal{U}\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$
Partial Rotation	Angle in radians.	$\mathcal{U}[-\pi, \pi]$
Translation	Translation in pixels.	$\mathcal{U}[-8, 8]^2$
Scale	Exponential Scaling factor.	$\mathcal{U}[-log(2), log(2)]$

Table 8. Sampling parameters for transformations used to obtain synthetically invariant datasets, from (Immer et al., 2022).

Constraint Set	Parameter	Range
Rotations	Angle in radians.	$[-\pi, \pi]$
Translation	Translation in pixels.	$[-16, 16]^2$
Scale	Exponential Scaling factor.	$[-1.5, 1.5]$

Table 9. Transformation sets used as invariance constraints. All sets are used simultanously, with the same constraint level (ϵ) for all transformations.

D. Additional Experimental Results

D.1. Automatic Data augmentation

The following section contains additional ablations and discussions about our algorithm. As in previous sections, we use the wide (Müller and Hutter, 2021) augmentation space and CIFAR image classification benchmarks. Our main motivation is to analyse how the different hyperparameter

	MLP	CNN
MNIST	0.6	0.35
F-MNIST	0.8	0.6

Table 10. Constraint levels used for different architectures and base datasets. We use the same constraint levels for all synthetically invariant versions of the same base dataset.

choices and the learning algorithm affect the generalization and invariance of the obtained solutions. First, we show how the dual variables adapts to different constraint levels during training, and link its dynamics to heuristically defined learning curricula. We then evaluate the effect of training for more epochs and link our observations to the known properties of early stopping in unconstrained learning, in section D.1.2. In section D.1.3, we analyse how the sampling approximation affects regularisation, by performing an ablation on the number of MH steps. Lastly, we include the observed frequencies of sampled transformations for different setups, so as to obtain further insights in how the distribution adapts throughout training (Section D.1.4).

D.1.1. Dual Variable Dynamics

As already mentioned, the dual variables control the weight of augmented samples during training, thus balancing the trade-off between fitting the primal objective (i.e. loss over training samples) and satisfying the constraints (i.e. loss over transformed samples). In penalised approaches, on the contrary, this trade-off is fixed. In Figure 4 we show how the dual variable adapts to different constraint levels, for Wide-ResNet-40-2 in CIFAR datasets using the standard setup. Note that for stricter constraint levels, the algorithm has not converged when it reaches 200 epochs. We also observe that the dynamics of our primal-dual algorithm resembles the augmentation learning curricula proposed by (Blaas et al., 2021).

D.1.2. Early Stopping

Our approach has slower convergence than unconstrained approaches and stopping training arbitrarily - after a fixed number of epochs - can result in solutions that are unfeasible or sub-optimal. However, early stopping is a popular regulariser, particularly for neural networks trained through gradient descent. Several empirical (Caruana et al., 2000; Rice et al., 2020) and theoretical (Ji et al., 2021; Li et al., 2020a; Duvenaud et al., 2016) results show its advantages in terms of generalisation and robustness to noisy training data. In general, the advantages of early stopping do not lie in the sub-optimality of the solution in terms of training error, but on nature of the regularization or prior imposed, which leads to non-uniform model selection among models with a given training error (Cataltepe et al., 1999).

To the best of our knowledge, there is no literature that ex-

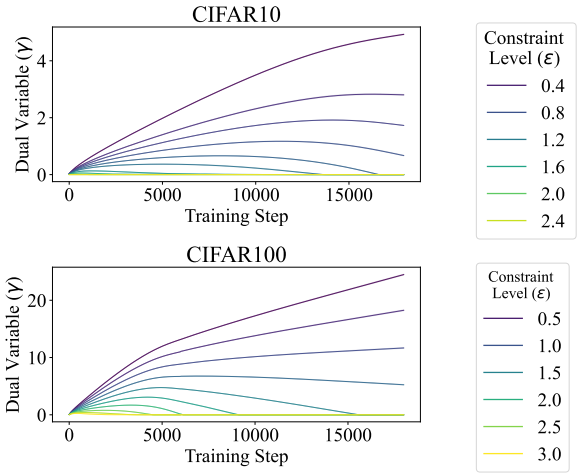

Figure 4. Evolution of dual variables during training for Wide-ResNet-40-2 in CIFAR datasets. For most levels, augmentation increases until the constraint becomes feasible and then decreases towards the end of training. For stricter levels, the algorithm does not converge in 200 epochs using the standard learning settings. However, the solutions obtained still show good properties.

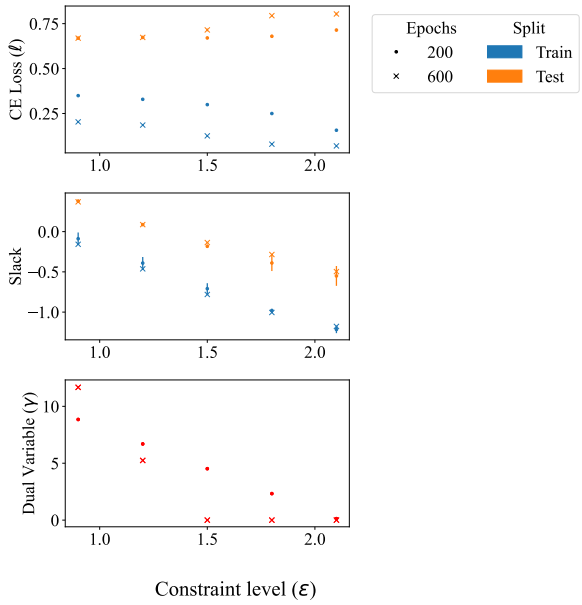

Figure 5. Training WideResnet-40-2 for 200 (standard) and 600 epochs in the CIFAR100 dataset, with different constraint levels.

plicitly addresses early stopping in empirical primal-dual learning. Whether the generalisation gap in constraint satisfaction can be reduced by early stopping regularisation, in the same manner early stopping regularisation can reduce the generalisation gap in unconstrained learning, thus remains unclear.

Figure 5 shows an ablation on the number of epochs. Non-zero dual variables and strict feasiblility show the algorithm has not yet converged at 200 epochs. At 600 epochs whereas constraint satisfaction shows little change, training loss decreases and the generalization gap increases. Thus, we observe early stopping has a larger impact on the primal objective than on the constraint. that although for stricter levels of the constraint the algorithm has not converged when training is stopped at 200 epochs, it can yield solutions that are still feasible and have a smaller generalization gap.

D.1.3. Sampling Steps Ablation

As already mentioned in Section 4.3, we used the Metropolis-Hastings algorithm with independent uniform proposals. As shown in Figure 6, using more steps of the chain allows samples to deviate further from the uniform distribution, as measured by the decrease in entropy. As already mentioned, dual variables give information of the trade-off between fitting clean and augmented data. We observe the final value of the dual variable is highly correlated with the entropy of sampled transformations. That is, transformations that deviate further from uniformity result in larger dual variables at the end of training. Furthermore, in 7 we show the evolution of dual variables for different sampling steps. Sampling distributions that are closer to worst case perturbations results in more stringent requirements, and thus dual variables grow more rapidly.

D.1.4. Sampled Transformations

We now include plots of the empirical transformation distributions. As shown in Figure 8, the frequency with which transformations are sampled varies depending on the dataset, which is a desirable property. Similar to (Zhang et al., 2020), we observe a prevalence of geometric transformations, unlike (Cubuk et al., 2018). In SVHN color transformations are less frequent than in CIFAR datasets, and the frequency of geometric transformations increases, which is interesting due to the perceptual importance of shape in the digit recognition task.

As already noted, using more steps allows the chain to deviate further from the uniform proposal distribution. We thus plot the frequency of sampled transformations across training epochs on Figure 9. We observe that as training progresses, the observed frequencies also deviate further from uniformity. This suggests that, due to the dataset and architecture, some transformations may be harder to fit than others.

Similarly, we include histograms for the sampled transformation levels in Figure 10. Extreme levels are sampled more frequently, but the empirical distributions vary depending on the transformation. The deviation from uniformity and the differences between transformations are accentuated as

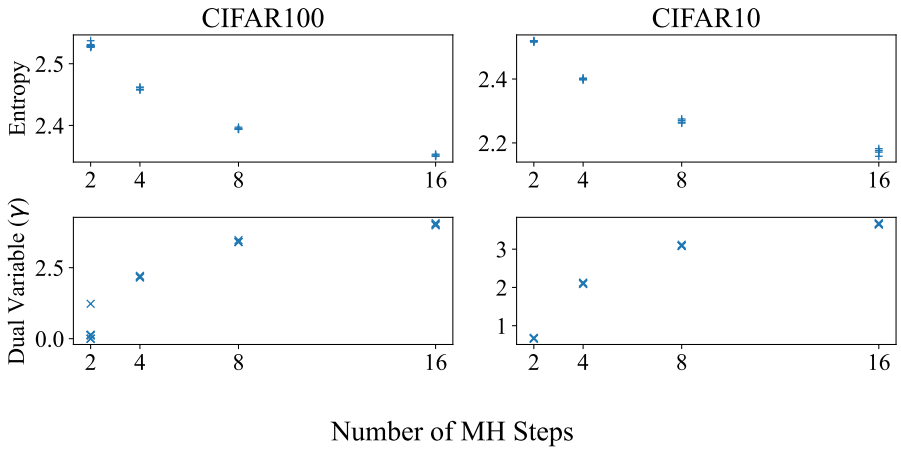

Figure 6. Number of Metropolis Hastings steps ablation for WideResnet-40-2 in CIFAR datasets. The constraint level is fixed for each dataset (0.8 in CIFAR10 and 2.1 for CIFAR100). We compute the entropy of the augmentations sampled at the last epoch of training. The value of the dual variable at epoch 200 increases with the number of steps, whereas the augmentation distribution entropy decreases. Each point represents an independent run. Experiments were repeated four times.

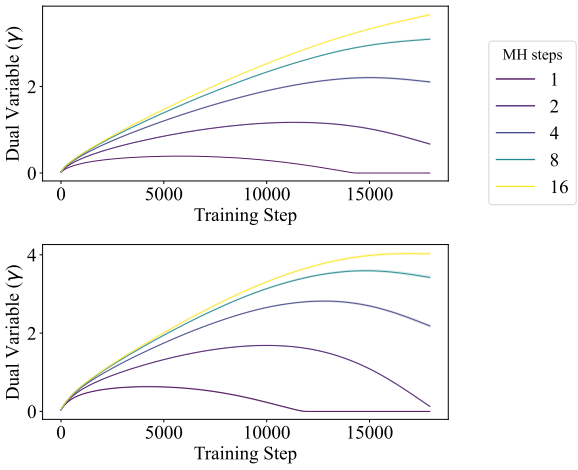

Figure 7. Evolution of dual variables during training for Wide-ResNet-40-2 in CIFAR datasets. The constraint level is fixed for each dataset (0.8 in CIFAR10 and 2.1 for CIFAR100). As the number of MH sampler steps increases so does the growth of dual variables, which reflects harder to satisfy constraints.

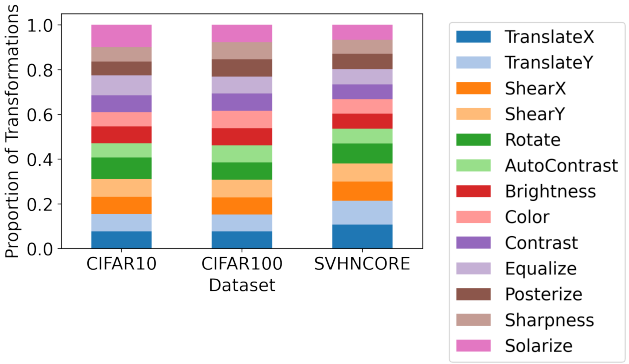

Figure 8. Frequency of sampled transformations for CIFAR and SVHN datasets using two MH sampler steps, for Wide-Resnet-28-10.

sampling steps increase (Figure 11).

D.2. Runtime analysis

As mentioned in section 4.3, the added computational cost of our algorithm is that of computing a forward pass for each MCMC step. As a result, trade-off between sampling $\lambda_c^\star$ accurately and computation arises. In Table 11 we provide an empirical runtime analysis for our method for different numbers of MH steps, and compare it with the training time of baseline methods. These times correspond solely to training, and it is difficult to account for the time taken to tune the hyperparameters of each method, which hinders direct comparisons. In the case of DeepAA (Zheng et al., 2022), it requires running a data augmentation policy search that takes 11 hours (more than $5\times$ training time) using the same hardware.

	TA	DeepAA	Ours	
			2 Steps	4 Steps
Epoch time (s)	12.6	13.3	32.5	51.2

Table 11. Time per epoch for WideResnet 40-2 in CIFAR 10 dataset, on a workstation with one NVIDIA RTX 3090 GPU and AMD Threadripper 3960X (24 cores, 3.80 GHz) CPU.

m	Accuracy
2	81.58
4	81.34
8	81.25
16	80.33

Table 12. Test accuracy (second column) for WideResnet 40-2 trained on CIFAR100 using different values of augmentations per sample (denoted as m, first column).

D.3. Batch mode Augmentation

All of the experiments were conducted using a single augmentation per training sample in the batch, that is $m = 1$ in Algorithm 1. As already mentioned, our approach is also suitable for batch mode augmentation, i.e. $m > 1$. Therefore, conducted a preliminary ablation study on m. As shown in table 12, increasing m resulted in a slight decrease in accuracy. However, increasing m leads to larger batch sizes, which affects optimization dynamics. Consequently, adjusting the learning rate and other optimization hyperparameters for each value of m could lead to performance improvements, as shown by (Zheng et al., 2022). Therefore, providing further ablations and analyses regarding batch mode sampling is an interesting future research direction.

D.4. The bias of data augmentation can hinder performance

As already mentioned in section A.2 there is empirical evidence that certain distributions over commonly used transformations can introduce biases that are detrimental to model performance and generalization (Blaas et al., 2021). We provide another simple experiment to show that in practice there exist transformation distributions over commonly used augmentation spaces that deteriorate model performance. We also showcase that balancing the amount of data augmentation (e.g. by including the original data) is important to mitigate and overcome this issue.

We restrict the transformations in the wide augmentation space (Müller and Hutter, 2021) to their maximum magnitude. We sample transformations according to $\lambda_c^\star$ using two MH steps as previously described. We compare against training without augmentation, and training using both augmentation and the original data (i.e. adding the identity) equally weighted. While sampling transformations according to $\lambda_c^\star$ deteriorates performance with respect to the model without augmentation, including both the identity and the augmented samples achieves a superior performance, as shown in Table 13.

No augmentation	$\lambda_c^\star$	$\lambda_c^\star$ + Identity
78.42 ± 0.31	75.19 ± 0.54	80.01 ± 0.26

Table 13. Image Classification test accuracy for WideResnet 40-2 in CIFAR100, trained using different augmentation policies defined over the wide (Müller and Hutter, 2021) augmentation space. The first column corresponds to using the training data without applying any transformations, and the second column to sampling transformations according to $\lambda_c^\star$, which results in lower accuracy. The third column corresponds to using both the original and augmented data equally weighted, which leads to an improvement in accuracy. We report the mean and confidence intervals computed over five independent runs.

D.5. Constraint Level Specification.

This experiment illustrates how dual variables can be used to adjust epsilon using simple heuristics. In the synthetic invariances setup, we first train the model with the same constraint level for all transformations, regardless of whether the transformation is a "true" invariance of the data distribution. That is, there is a mismatch between the constraints and the invariances in the datasets, which can result in poor performance. We then proceed to:

(i) Tighten the constraints associated with zero dual variables by dividing epsilon by a factor of two.

(ii) Relax the remaining constraints, which showed large associated dual variables, by multiplying ϵ by a factor of two.

As shown in Table 14, this simple heuristic leads to an improvement in test accuracy over the initial (misspecified) constraint levels. Therefore, adapting the constraint level by leveraging dual dynamics while training is a promising future work direction.

Dataset	Architecture	Invariance	Same ϵ	Adjusted ϵ
MNIST	MLP	Translation	93.5 ± 0.2	94.9 ± 0.1
		Rotation (90)	96.95 ± 0.03	97.28 ± 0.07
		Rotation (180)	93.18 ± 0.03	95.35 ± 0.03
	CNN	Translation	96.00 ± 0.02	97.79 ± 0.09
		Rotation (90)	98.15 ± 0.09	98.5 ± 0.1
		Rotation (180)	96.3 ± 0.3	97.2 ± 0.3
F-MNIST	MLP	Translation	76.5 ± 0.5	81.2 ± 0.2
		Rotation (90)	81.8 ± 0.1	84.78 ± 0.06
		Rotation (180)	76.1 ± 0.7	83.85 ± 0.21

Table 14. Classification accuracy for synthetically invariant datasets using the same constraint level for all transformations (third column) and adjusting constraint levels according to dual variables (fourth column).

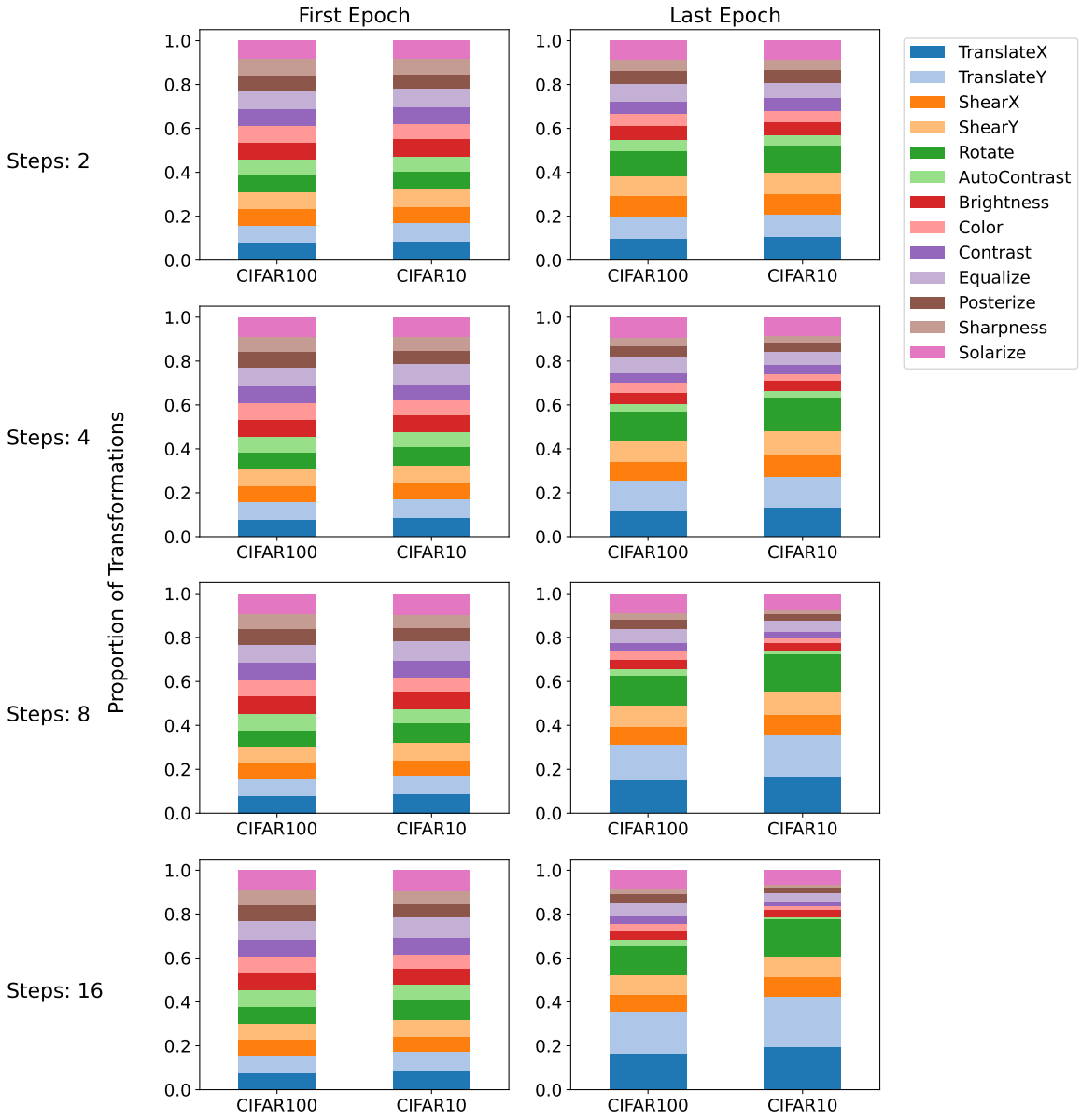

Figure 9. Frequency of sampled transformations for CIFAR datasets in the first and last epochs of training, for Wide-Resnet-40-2. As the number of steps increases, the the entropy of sampled transformations decreases, i.e., observed frequencies get further from uniformity.

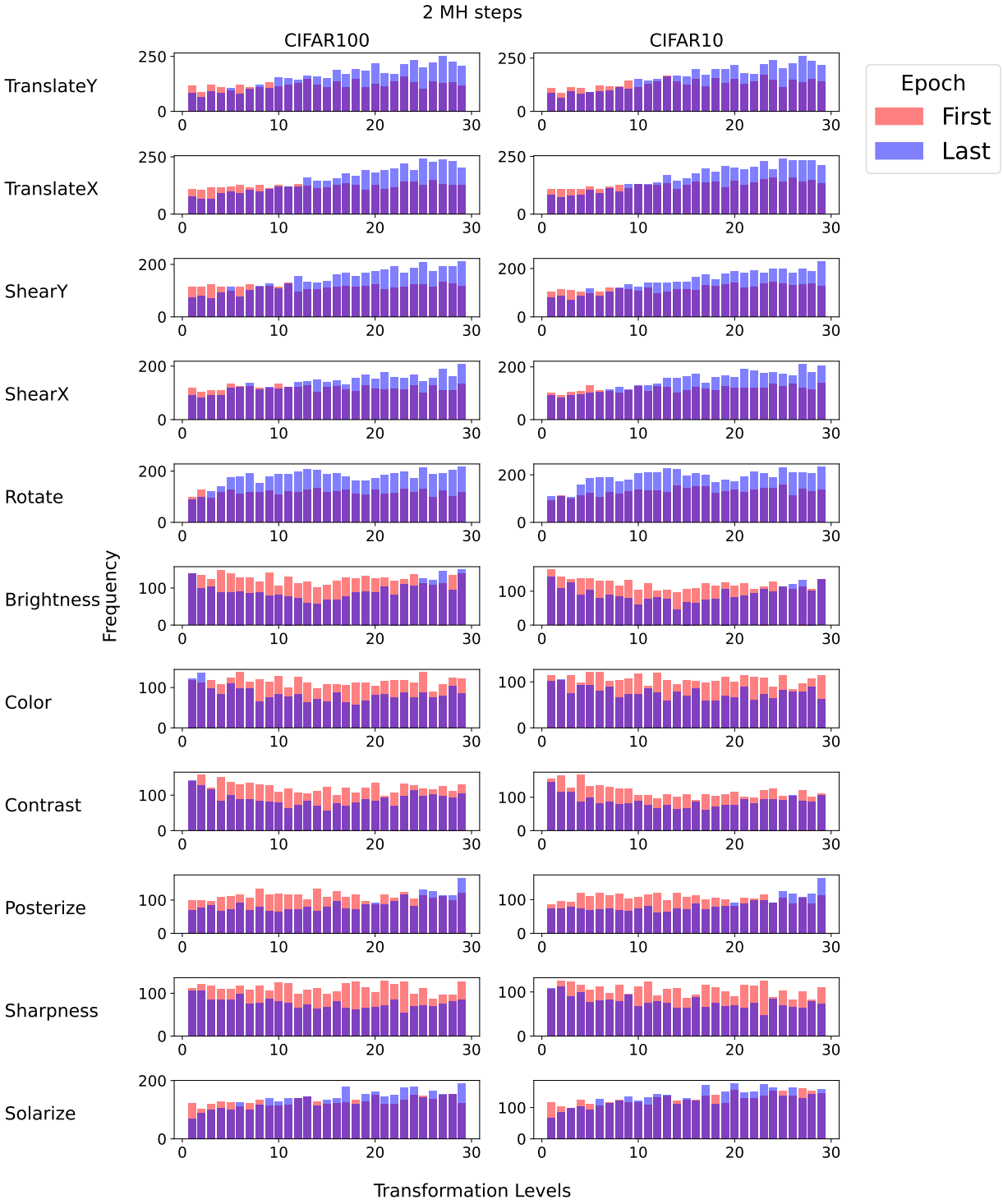

Figure 10. Frequency of sampled transformation levels across epochs, for different transformations, using two MH steps. Extreme levels are sampled more frequently, but some transformations deviate further from uniformity.

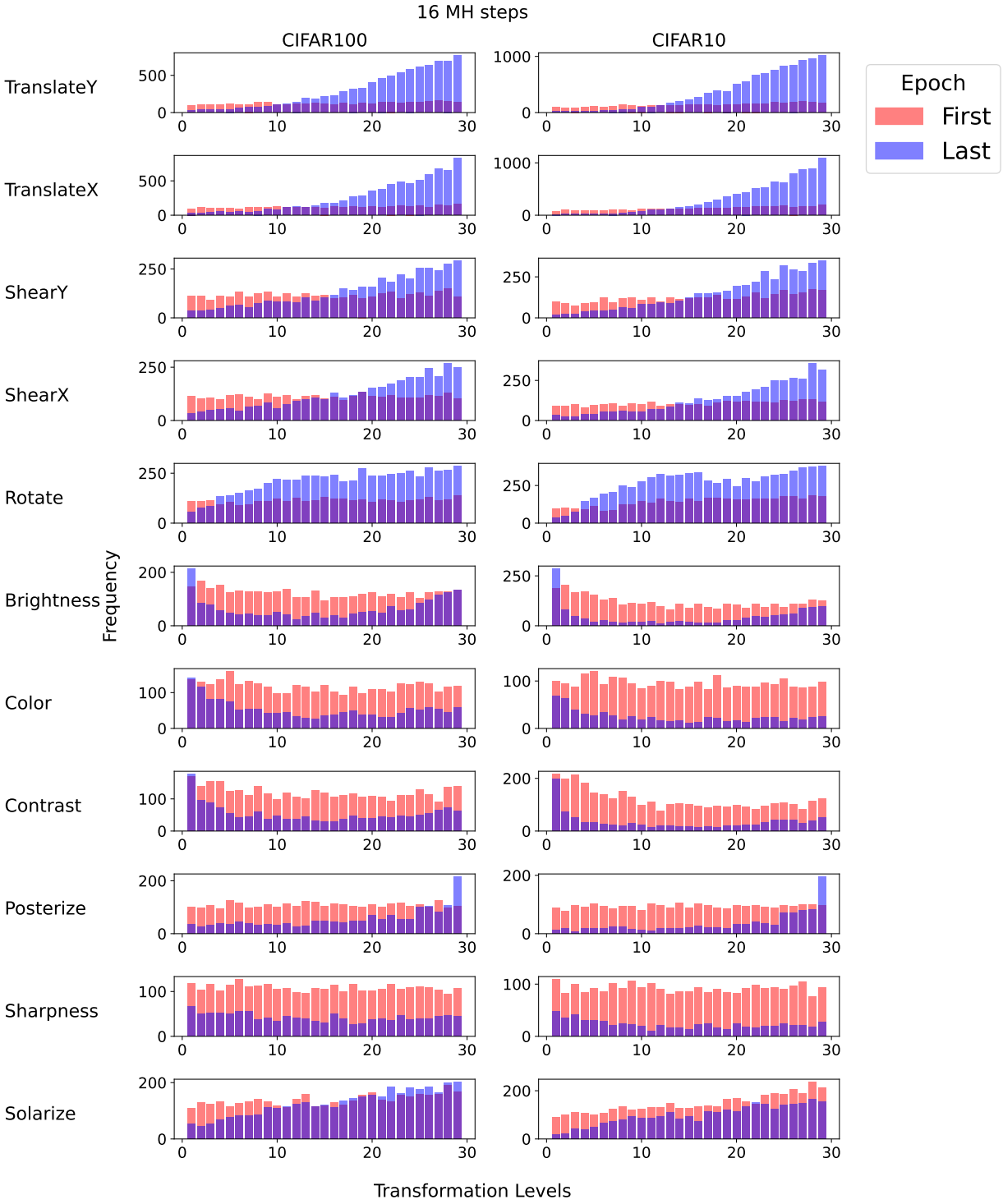

Figure 11. Frequency of sampled transformation levels for the first and last epochs, for different transformations, using sixteen MH steps. The frequencies concentrate in extreme values for some transformations.