DAMix: Exploiting Deep Autoregressive Model Zoo for Improving Lossless Compression Generalization

Qishi Dong[2,1*], Fengwei Zhou[1*], Ning Kang[1*], Chuanlong Xie[3,1*], Shifeng Zhang[1], Jiawei Li[1], Heng Peng[2†], Zhenguo Li[1†]

[1] Huawei Noah's Ark Lab,
[2] Hong Kong Baptist Uiversity,
[3] Beijing Normal University
19481284@life.hkbu.edu.hk, fzhou@connect.ust.hk, {kang.ning2, zhangshifeng4, li.jiawei}@huawei.com, clxie@bnu.edu.cn, hpeng@hkbu.edu.hk, li.zhenguo@huawei.com

Abstract

Deep generative models have demonstrated superior performance in lossless compression on identically distributed data. However, in real-world scenarios, data to be compressed are of various distributions and usually cannot be known in advance. Thus, commercially expected neural compression must have strong Out-of-Distribution (OoD) generalization capabilities. Compared with traditional compression methods, deep learning methods have intrinsic flaws for OoD generalization. In this work, we make the attempt to tackle this challenge by exploiting a zoo of Deep Autoregressive models (DAMix). We build a model zoo consisting of autoregressive models trained on data from diverse distributions. In the test phase, we select useful expert models by a simple model evaluation score and adaptively aggregate the predictions of selected models. By assuming the outputs from each expert models are biased in favor of their training distributions, a von Mises-Fisher based filter is proposed to recover the value of unbiased predictions that provides more accurate density estimations than a single model. We derive the posterior of unbiased predictions as well as concentration parameters in the filter, and a novel temporal Stein variational gradient descent for sequential data is proposed to adaptively update the posterior distributions. We evaluate DAMix on 22 image datasets, including in-distribution and OoD data, and demonstrate that making use of unbiased predictions has up to 45.6% improvement over the single model trained on ImageNet.

Introduction

The big data era, with the huge amount of data being generated each year, inspires new business lines including cloud service and streaming platforms. This motivates the industry to develop more efficient and effective lossless compression methods (Alakuijala et al. 2019; Sneyers and Wuille 2016; Collet and Turner 2016; Ahmed, Islam, and Uddin 2018). According to Shannon's source coding theorem, the more accurately the distribution of the data can be estimated, the better the limits of compression can be reached (MacKay 2003). Hence, deep generative models, such as VAEs (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014; Ho, Lohn, and Abbeel 2019), normalizing flows (Rezende and Mohamed 2015; Tran et al. 2019), and autoregressive models (Uria, Murray, and Larochelle 2013; Van den Oord et al. 2016; Salimans et al. 2017), have shown great potential in improving lossless compression ratio due to their powerful ability in modeling the distribution of various types of data, and various lossless compression algorithms (Zhang et al. 2021c,b,a; Kang et al. 2022) have been proposed based on deep generative models.

One primary assumption, ensuring these models are effective, is training and test data being Independent and Identically Distributed (IID). However, data to be compressed in real-world scenarios follow very different distributions and are usually OoD samples that cannot be known in advance. To cope with it, previous works in context mixing attempt to adaptively mix different compression algorithms. This idea has been widely used in non-AI compression algorithms to combine multiple statistical models to yield a prediction that is often more accurate than any of the individual predictions. PNG (Boutell 1997) makes use of 5 models to predict each pixel and mix them simply by selecting one of them for each line. WebP increases the number of models to 13, one of which is chosen for each block. For more advanced techniques, linear mixing and logistic mixing are applied in PAQ to mix 1000+ models. CMIX (Knoll 2007) further improves by introducing an LSTM Mixer with a gated linear network (Veness et al. 2017). However, none of them can be naturally applied to AI lossless compression.

On the other hand, recent works (Hendrycks et al. 2020; Albuquerque et al. 2020; Yi et al. 2021; Radford et al. 2021) have shown the advantages of pre-training for improving OoD generalization, i.e., learning from multiple training domains and being well applied to an unseen domain. Yi et al. (2021) prove that adversarially pre-trained models perform better for OoD generalization. Yu et al. (2021) show that the right choice of pre-trained models can achieve SOTA OoD results. Radford et al. (2021) demonstrate that large-scale pre-training on a dataset of image-text pairs results in much more robust models for downstream tasks with various distribution shifts. For data compression, Zhang et al. (2021c) and Zhang et al. (2021b) also show that large-scale pre-training can alleviate the performance degradation on OoD data. Naturally, we

[*] Equal Contribution. This work was carried out at Huawei Noah's Ark Lab.
[†] Corresponding Author.

expect a model zoo containing a large number of pre-trained models can further improve OoD generalization.

In this paper, we design a model zoo with large-scale pre-trained models covering possible distribution shifts to improve OoD generalization for lossless image compression. To maximally exploit our zoo of deep generative models, two important issues need to be addressed. First, given an image to be compressed, we need to quickly select a small subset of suitable models, since we do not want "wrong" expert models to be involved in subsequent aggregation, especially when the size of the model zoo is large. Second, we need to make adaptive adjustments to multiple models when dealing with the sequential pixels, since different local parts may follow different distributions (Fang et al. 2021; Zhang, Zhang, and McDonagh 2021). For example, an expert model trained on a vehicle dataset may predict unbiasedly on the vehicle part of an OoD image containing a variety of objects, but biasedly on the rest parts. If we can train different deep expert models on purpose to deal with different distributions, and dynamically select the most appropriate models for specific local image areas, we may be able to deal with image data with diverse distributions.

However, several challenges raise as many context mixing algorithms are based on simple weighted averaging and do not lead to unbiased predictions. In addition, it is technically difficult to build a sophisticated meta-probabilistic model to aggregate the model zoo, since the data we are dealing with are constrained in a simplex, e.g. the outputs from expert models are Multinomial distributions, which makes modeling using common distributions like Gaussian invalid. On the other hand, we build a model zoo with the spatially autoregressive model like PixelCNN++ (Salimans et al. 2017) since it is powerful in modeling image density and leads to outstanding performance in lossless compression. However, such models are usually blamed for slow inference. Thus, the proposed algorithm must be efficient enough to prevent additional computational burden.

To tackle the aforementioned problems and challenges, we propose DAMix, a **D**eep **A**utoregressive model zoo with quick model selection and model **Mix**ing for lossless compression. We pre-train multiple PixelCNN++ based on diverse datasets, one model corresponding to one dataset. In the model selection phase, the model score is evaluated based on the log-likelihood of a patch sampled from a given test image. A few models with higher scores are then selected for subsequent aggregation. Then, we treat the outputs of a PixelCNN++ as sequential data and obtain locally unbiased predictions for OoD images via a von Mises-Fisher (vMF) filter, whose **concentration parameters in vMF act as the mixing weights of pre-trained models' outputs.** Finally, to infer the posterior distribution of mixing weights, we propose a novel Temporal Stein Variational Gradient Descent (TSVGD) algorithm for online Bayesian inference. We give theoretical guarantees that the empirical distribution of the concentration parameters approaches the true posterior as TSVGD iterations progress, which implies that **our algorithm converges to an optimal mixing scheme.**

Our main contributions can be summarized as follows:

1) We build a zoo of deep autoregressive models trained by different datasets to improve compression on OoD data. Our model zoo can cover diverse distributions and empirically outperform single-model methods.

2) We propose a novel implicit mixing scheme to discover the unbiased density of local areas. Our method inspires a new probabilistic view for model ensemble that we prove inferring the posterior distribution of unbiased predictions in our vMF filter is equivalent to adaptively assembling models with weighted averaging.

3) We propose TSVGD, a general Bayesian inference method for sequential data. We provide the theoretical guarantee that empirical distributions of latent concentration parameters will converge to the true posterior leading to the optimal mixing scheme. We also analyze the complexity of TSVGD and show it does not increase the computational burden of PixelCNN++.

4) Extensive experiments on 22 datasets show DAMix effectively utilizes the model zoo to improve the OoD generalization of neural compression. The compression benchmark of 22 datasets is helpful for future research.

Preliminaries

Von Mises-Fisher Distribution. We denote the unit sphere by $\mathcal{S}^{d-1} = \left\{\mathbf{x}\in\mathbb{R}^d : \|\mathbf{x}\|=1\right\}$. We say a random variable $\mathbf{x}\in\mathcal{S}^{d-1}$ follows a von Mises-Fisher (vMF) distribution if its density function is

$$\mathrm{vMF}(\mathbf{x}\mid\boldsymbol{\mu},\kappa)=\mathcal{C}_d(\kappa)\exp\left(\kappa\boldsymbol{\mu}^T\mathbf{x}\right),$$

where $\boldsymbol{\mu}\in\mathcal{S}^{d-1}$, $\kappa\geq 0$ and $\mathcal{C}_d(\kappa)$ denotes the normalization constant. The parameters $\boldsymbol{\mu}$ and κ are called the mean direction and the concentration parameter, respectively. The greater the value of κ, the higher the concentration of the distribution around the mean direction $\boldsymbol{\mu}$.

Kalman Filter. Given a series of observed measurements $\{\mathbf{x}_t\}_{t=1}^T$ and the corresponding unknown values $\{\mathbf{z}_t\}_{t=1}^T$ with $\mathbf{x}_t,\mathbf{z}_t\in\mathbb{R}^d$, Kalman filter assumes

$$\mathbf{z}_t=\mathbf{A}\mathbf{z}_{t-1}+\mathbf{w}_t \quad \mathbf{x}_t=\mathbf{C}\mathbf{z}_t+\mathbf{v}_t,$$

with white noise: $\mathbf{w}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Gamma})$ and $\mathbf{v}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})$. Noted that Kalman filter defines a linear dynamic system by the Gaussian-linear model. Thus the inference problem is tractably solved.

Variational Inference. Variational Inference (VI) (Blei, Kucukelbir, and McAuliffe 2017) works as a faster alternative to MCMC (Gelfand and Smith 1990) for Bayesian inference. Recent developments in VI have tried to combine classic variational inference with MCMC (Liu et al. 2019; Liu and Wang 2016; Saeedi et al. 2017), leading to Particle Variational Inference (PVI), among which, one interesting method is Stein Variational Gradient Descent (SVGD) (Liu and Wang 2016). Given an arbitrary empirical distribution $\{\theta_i^0\}_{i=1}^n$ and target distribution $p(\theta)$, the particles from empirical distribution $\{\theta_i^\ell\}_{i=1}^n$ will converge towards target distribution in a gradient descent manner: $\theta_i^{\ell+1}\leftarrow\theta_i^\ell+\epsilon_\ell\hat{\phi}^*\left(\theta_i^\ell\right)$, where

$$\hat{\phi}^*(\theta)=\frac{1}{n}\sum_{j=1}^n\left[k\left(\theta_j^\ell,\theta\right)\nabla_{\theta_j^\ell}\log p\left(\theta_j^\ell\right)+\nabla_{\theta_j^\ell}k\left(\theta_j^\ell,\theta\right)\right]$$

and $k(\cdot,\cdot)$ is a kernel function.

Methodology

Model Zoo of PixelCNN++

For input data $\mathbf{x} = [x_1,...,x_T]$, neural compression methods involve a deep generative model to estimate the density $p(\mathbf{x})$, and the data is encoded with codelength $-\log_2 p(\mathbf{x})$. Therefore, the quality of density estimation determines how well the data is compressed. Based on the success of previous AI compression, building an AI compression model zoo, i.e. DAMix, is a promising attempt for practical lossless compression. We use PixelCNN++ (Salimans et al. 2017) as the base model, which is a deep autoregressive model that outputs the distribution $p(x_t|x_1,...,x_{t-1})$ of each discrete variable x_t in a recursive manner. Thus we can apply PixelCNN++ to data compression by encoding using predicted distribution with codelength $-\log_2 p(x_t|x_1,...,x_{t-1})$ with Arithemic Coder (Rissanen and Langdon 1979).

Model Selection by Log-Likelihood

Given an image, we want to remove inappropriate experts. Here we propose a simple evaluation score for quick model selection. He et al. (2021) illustrate the global semantic can be inferred using only a small set of pixels from the local area. Thus, we evaluate each expert model by sampling a small patch from the image. The performance of an expert model over this patch should be similar to that over the entire image. We estimate the log-likelihood of this patch as an evaluation score. The models with high scores are then selected for the next phase. In the experiment, we find model selection by log-likelihood is highly consistent and much faster than evaluating each model over the entire image.

Model Mixing via vMF Filter

Given an image with 256-level gray, PixelCNN++ predicts a Multinomial distribution with class 256 for each pixel. Thus each observed distribution is constrained in a 255-simplex:

$$\begin{aligned}\Delta^{255} &= \Big\{\mathbf{p} = [p_1,...,p_{256}] : \mathbf{p} \in \mathbb{R}_+^{256}, \sum_{i=1}^{256} p_i = 1\Big\} \\ &= \Big\{\mathbf{p}' = [\sqrt{p}_1,...,\sqrt{p}_{256}] : \mathbf{p}' \in \mathbb{R}_+^{256}, \|\mathbf{p}'\|_2 = 1\Big\}.\end{aligned}$$

By simply taking the square root, we convert the summation constrain of a simplex to the constrain of ℓ^2-norm. This transformation allows us to use polar coordinates to model the randomness of distributions (Davidson et al. 2018). Following this idea, we assume the outputs of PixelCNN++ follow a von Mises-Fisher (vMF) distribution.

Let $\mathbf{x} = [x_1,...,x_T]$ be an image with 256-level gray and $x_t \in \{0,1,...,255\}$ is the value of the t-th pixel.[1] We denote $\mathbf{p}_t^m \in \Delta^{255}$ be the output of the m-th model on the t-th pixel and write the results for the t-th pixel and the whole image as

$$\mathbf{p}_t = [\mathbf{p}_t^1,...,\mathbf{p}_t^M] \in \mathbb{R}_+^{M\times D}, \quad \boldsymbol{P} = [\mathbf{p}_1,...,\mathbf{p}_T] \in \mathbb{R}_+^{T\times M\times D}$$

respectively. Here $D = 256$ and M is the size of the model zoo. We denote $\boldsymbol{\mu}_t \in \Delta^{255}$ as the unbiased (ground-truth) density for the t-th pixel. Here $\boldsymbol{\mu}_t$ is a latent variable behind the observed outputs $\mathbf{p}_t$. Motivated by Kalman Filter, we approximate the generation of $\mathbf{p}_t^m$ with the following dynamic mechanism (vMF filter):

$$\boldsymbol{\mu}_t \sim \mathrm{vMF}\big(\boldsymbol{\mu}_{t-1}, \kappa_t^0\big), \quad \mathbf{p}_t^m \sim \mathrm{vMF}(\boldsymbol{\mu}_t, \kappa_t^m), \qquad (1)$$

where $\kappa_t^0,...,\kappa_t^M$ are concentration parameters. It is easy to see the likelihood of $\mathbf{p}_t$ given $\boldsymbol{\mu}_t$ is $\prod_{m=1}^M \mathrm{vMF}(\mathbf{p}_t^m \mid \boldsymbol{\mu}_t, \kappa_t^m)$. In this work, we propose an algorithm to infer the ground-truth distributions $\boldsymbol{\mu}_t, t = 1,...,T$ from the observed outputs $\boldsymbol{P}$ and the vMF filter.

Using vMF distribution. Gaussian distribution fails to model the outputs of PixelCNN++ due to the norm constrain. Although studies in compositional data analysis (Aitchison 1982) show alternatives like Dirichlet and log-normal distribution are well-defined on the simplex, those distributions lose the analytical properties of Gaussian that given a joint Gaussian distribution, its conditional and marginal distribution are also Gaussian. Meanwhile, tractable inference of Kalman filter relies heavily on this property to impose. Strictly speaking, the truncated vMF distribution is the best to match the data, since the square root transformation restricts its support to the non-negative region. However, the truncated posterior distribution is not tractable. We thus relax it to original vMF distribution to enjoy analytical properties and computational efficiency. Such relaxation has little influence on the performance of DAMix, because the main vMF posterior density still concentrates on the non-negative region.

Relationship to Model Mixing. In the next section, we propose a novel Stein variational gradient descent for online Bayesian inference by introducing the prior distribution of $\boldsymbol{\kappa}_t = [\kappa_t^0, \kappa_t^1, ..., \kappa_t^M]$. One can estimate the posterior density of $\boldsymbol{\mu}_t$ concentrates in direction of:

$$\kappa_t^0 \mathbb{E}[\boldsymbol{\mu}_{t-1} \mid \boldsymbol{P}_{t-1}, \mathbf{x}_{t-2}] + \sum_{m=1}^M \kappa_t^m \mathbf{p}_t^m.$$

Therefore, the concentration parameters $\kappa_t^1,...,\kappa_t^m$ plays a role as "mixing weights" for expert models at pixel t and κ_t^0 controls inheritance from the previous pixel.

Posterior Inference

Prior Distributions of $\boldsymbol{\kappa}$

We complete the specification of the proposed filter by introducing the prior distributions for concentration parameters $\boldsymbol{\kappa}_t$. In the first level of the prior distribution, we introduce a similar temporal transition $\boldsymbol{\kappa}_t \sim p(\boldsymbol{\kappa}_t \mid \boldsymbol{\kappa}_{t-1})$ consistent with the dynamic mechanism in vMF filter. In the second level, a hierarchical Dirichlet prior is adopted to leverage the information of the current true pixel value x_t:

$$\mathrm{Dir}(\boldsymbol{\kappa}_{t+1}) = \frac{1}{\mathrm{B}(\boldsymbol{\kappa}_{t+1})} \prod_{m=1}^M \tilde{\mathbf{p}}_t^m(x_t)^{\kappa_{t+1}^m} \cdot \tilde{\boldsymbol{\mu}}_t(x_t)^{\kappa_{t+1}^0}, \qquad (2)$$

where $\mathrm{B}(\cdot)$ is a multivariate beta function.

Notice that prior knowledge helps adaptively adjust the posterior distributions of $\boldsymbol{\kappa}_t$ for different local image areas, such that higher weights will be assigned to the expert models that predict unbiasedly on the current image patch.

[1] Although the proposed method is illustrated with the image example, it can fit easily to any types of sequential data with deep autoregressive models, e.g., text or video.

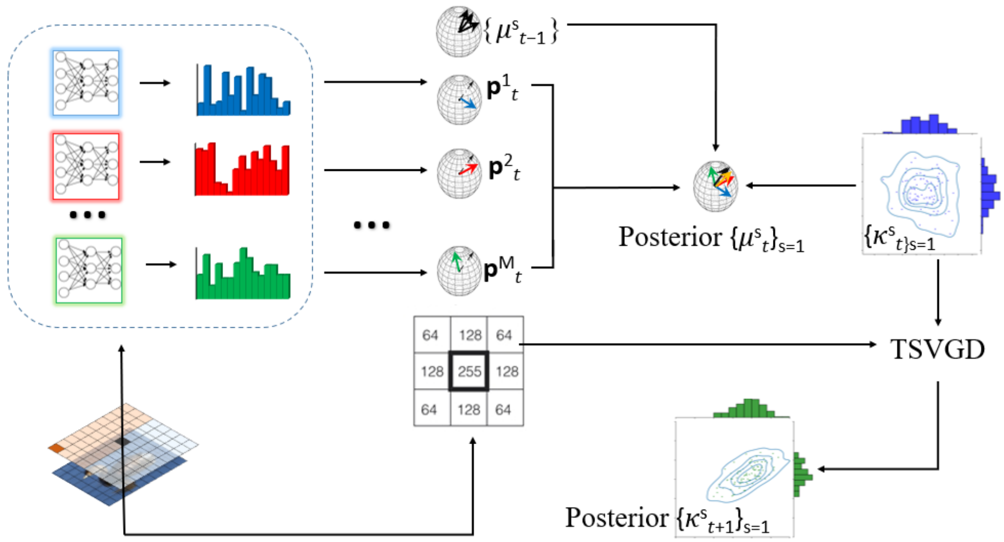

Figure 1: An overview of the proposed filter. DAMix forward propagates current pixel and projects the predicted distributions to a hyperball. The pixel t is compressed using the unbiased predictions $\boldsymbol{\mu}_t$ which is inferred by the observed projected directions and prior directions $\boldsymbol{\mu}_{t-1}$ with concentration parameters $\boldsymbol{\kappa}_t$. We update $\boldsymbol{\kappa}_{t+1}$ by prior information x_t for the next iteration using TSVGD.

In Eq. (2), $[\tilde{\mathbf{p}}_t^1(x_t),...,\tilde{\mathbf{p}}_t^M(x_t),\tilde{\boldsymbol{\mu}}_t(x_t)]$ is a normalized vector consisting of x_t-th element of observed likelihood $\mathbf{p}_t^m$ and previous $\mathbb{E}[\boldsymbol{\mu}_t \,|\, \boldsymbol{P}_t,\mathbf{x}_{t-1}]$. It's easy to see with high posterior probability, the estimated κ_{t+1}^m is larger when the model m has larger likelihood $\tilde{p}_{x_t}^m$. Therefore, the posterior distribution of $\boldsymbol{\kappa}_{t+1}$ can be adaptively updated in favor of previously better-performed models.

Unbiased Density Estimation

Since the unbiased density $\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_T$ is what we need but unknown for compression, we present an inference algorithm for joint posterior $p(\boldsymbol{\mu}_1,...,\boldsymbol{\mu}_T,\boldsymbol{\kappa}_1,...,\boldsymbol{\kappa}_T \,|\, \boldsymbol{P},\boldsymbol{x})$. An overview of the posterior inference is illustrated in Figure 1. [2] We start by inferring the posterior distribution of $\boldsymbol{\mu}_t$. To proceed further, we need more notations:

$$\mathbf{x}_t = [x_1,...,x_t] \quad \text{and} \quad \boldsymbol{P}_t = [\mathbf{p}_1,...,\mathbf{p}_t], \tag{3}$$

which are the previous pixels and outputs before the $(t+1)$-th pixel. The following Theorem shows the connection with the Kalman filter.

Theorem 1. *Given all observations $\{\boldsymbol{P}_t,\, \mathbf{x}_{t-1}\}$ at the t-th pixel, the joint posterior distributions of $\{\boldsymbol{\mu}_i\}_{i=1}^t$ are proportional to the following joint distribution:*

$$p(\boldsymbol{\mu}_1,...,\boldsymbol{\mu}_t \,|\, \boldsymbol{P}_t,\mathbf{x}_{t-1})$$
$$\propto \prod_{i=1}^{t} p(\mathbf{p}_i \,|\, \boldsymbol{\mu}_i,\mathbb{E}[\boldsymbol{\kappa}_i|\mathbf{x}_{i-1}]) \cdot p(\boldsymbol{\mu}_i \,|\, \boldsymbol{\mu}_{i-1},\mathbb{E}[\boldsymbol{\kappa}_i|\mathbf{x}_{i-1}]),$$

where $\mathbb{E}[\boldsymbol{\kappa}_i|\mathbf{x}_{i-1}]$ denotes the posterior expectation of $\boldsymbol{\kappa}_i$ given $\mathbf{x}_{i-1}$.

This shows the posterior of $\boldsymbol{\mu}_t$ has the same joint posterior distribution structure as Kalman filter (Welch, Bishop et al. 1995). Since the Kalman filter simply replaces all above emission distributions $p(\cdot)$ with Gaussian, its posterior distributions have analytical Gaussian forms. Naturally, we expect the same nice property from the von-Mises Fisher filter. Fortunately, the likelihood and conjugate prior for vMF distribution are of the same form, which leads to a tractable posterior. We consider the forward message passing from the initial state $\boldsymbol{\mu}_0$. According to **Theorem** 1, the unnormalized marginal distributions $p(\boldsymbol{\mu}_t \,|\, \boldsymbol{P}_t,\mathbf{x}_{t-1})$ are given by:

$$p(\boldsymbol{\mu}_t \,|\, \boldsymbol{P}_t,\mathbf{x}_{t-1}) \propto p(\mathbf{p}_t \,|\, \boldsymbol{\mu}_t,\mathbb{E}[\boldsymbol{\kappa}_t|\mathbf{x}_{t-1}])$$
$$\times \int p(\boldsymbol{\mu}_t \,|\, \boldsymbol{\mu}_{t-1},\mathbb{E}[\boldsymbol{\kappa}_t|\mathbf{x}_{t-1}]) \cdot p\Big(\boldsymbol{\mu}_{t-1} \,|\, \boldsymbol{P}_{t-1},\mathbf{x}_{t-2}\Big) \mathrm{d}\boldsymbol{\mu}_{t-1}.$$

However, unlike Kalman filter, the above integral is intractable for vMF distribution with $t>2$. We can avoid computing the integral by sampling from its predecessor $p(\boldsymbol{\mu}_{t-1} \,|\, \boldsymbol{P}_{t-1},\mathbf{x}_{t-2})$. The technique of sampling from vMF is referred to (Davidson et al. 2018). Using the conjugate property, the marginal posterior distribution now has the analytical mixture of vMF form using S previous posterior samples $\{\boldsymbol{\mu}_{t-1}^s\}_{s=1}^S$:

$$\begin{aligned} p(\boldsymbol{\mu}_t \,|\, \boldsymbol{P}_t,\mathbf{x}_{t-1}) &\approx \frac{1}{S}\sum_{s=1}^{S}\frac{1}{C_t^s}\mathrm{vMF}(\boldsymbol{\mu}_t \,|\, \boldsymbol{\mu}_{t-1}^s,\mathbb{E}[\boldsymbol{\kappa}_t|\mathbf{x}_{t-1}]) \\ &\quad \times \mathrm{vMF}(\mathbf{p}_t \,|\, \boldsymbol{\mu}_t,\mathbb{E}[\boldsymbol{\kappa}_t|\mathbf{x}_{t-1}]) \\ &= \frac{1}{S}\sum_{s=1}^{S}\mathrm{vMF}\left(\boldsymbol{\mu}_t \,|\, \frac{\boldsymbol{\lambda}_t^s}{\|\boldsymbol{\lambda}_t^s\|_2},\|\boldsymbol{\lambda}_t^s\|_2\right), \end{aligned} \tag{4}$$

where $\boldsymbol{\lambda}_t^s = \kappa_t^0 \cdot \boldsymbol{\mu}_{t-1}^s + \sum_{m=1}^{M} \kappa_t^m \cdot \mathbf{p}_t^m$ and $C_t^s = \mathcal{C}_D(\boldsymbol{\lambda}_t^s)$ is the normalization constant. The last equality comes from the tractability of vMF distribution. The term $\boldsymbol{\lambda}_t^s$ corresponds to the direction of the main posterior mass for each mixture component of $\boldsymbol{\mu}_t$ and is determined by weighted averaging of each prediction $\mathbf{p}_t^m$ and $\boldsymbol{\mu}_{t-1}$. This shows our filter can be viewed as a recursive model ensemble procedure (Lakshminarayanan, Pritzel, and Blundell 2017) with adaptive mixing weights $\boldsymbol{\kappa}_t$.

[2] Note that the same algorithm can also be derived from the perspective of mean-field variational inference(Blei, Kucukelbir, and McAuliffe 2017). We leave it to the Appendix.

Temporal Stein Variational Gradient Descent

So far, the marginal distribution of $\boldsymbol{\mu}_t$ depends on the posterior expectation $\mathbb{E}[\boldsymbol{\kappa}_t|\mathbf{x}_{t-1}]$. Since the choice of prior in our method is flexible, we seek the non-parametric inference method for the posterior of $\boldsymbol{\kappa}_t$ (Gershman, Hoffman, and Blei 2012; Liu and Wang 2016; Ranganath, Gerrish, and Blei 2014), which generalizes the inference procedure instead of derivation on a model-by-model basis. Furthermore, the inference method should support online updating, while traditional methods (Andrieu et al. 2003; Blei, Kucukelbir, and McAuliffe 2017) involve full-data iterations, and thus they can not be adapted to sequential data.

We propose Temporal Stein variational gradient descent (TSVGD) to allow online approximate Bayesian inference. The proposed method is a general algorithm that can be applied to any sequential data. Let

$$f(\mathbf{x}_n\,|\,\theta)=f(x_1\,|\,\theta)f(x_2\,|\,x_1,\theta)\cdots f(x_n\,|\,\mathbf{x}_{n-1},\theta)$$

be the likelihood function of unknown parameter θ on sequential data $\mathbf{x}_n=[x_1,...,x_n]$. The posterior of θ at time n is

$$\pi_n(\theta)=f(\theta\,|\,\mathbf{x_n})\propto\pi(\theta)f(\mathbf{x_n}\,|\,\theta).$$

The following results form the main idea of TSVGD.

Theorem 2. *Let $\boldsymbol{T}(\theta)=\theta+\epsilon\boldsymbol{\phi}(\theta)$ be the update operator of θ with the direction $\phi(\theta)$. We write $\Pi_{[\boldsymbol{T}]}(\theta)$ as the distribution of $\boldsymbol{T}(\theta)$. Give $\mathbf{x}_n$ and the variational posterior $\Pi_{n-1}(\theta)$ given $\mathbf{x}_{n-1}$, the direction of steepest descent that maximizes the negative gradient $\nabla_\epsilon \mathrm{KL}\big(\Pi_{[\boldsymbol{T}]}\|\pi_n\big)\big|_{\epsilon=0}$ is given by*

$$\boldsymbol{\phi}^*_{\Pi,\pi_n}(\theta)=\mathbb{E}_{\theta'\sim\Pi_{n-1}}\Big[k(\theta,\theta')\nabla_{\theta'}\log f(x_n\,|\,\mathbf{x}_{n-1},\theta') + \frac{1}{n}\nabla_\theta k(\theta,\theta')\Big],$$

where $\boldsymbol{\phi}^*\in\big\{\phi\in\mathcal{H}^d:\|\phi\|^2_{\mathcal{H}^d}\le r_n\big\}$ is in the zero-centered unit ball of vector-valued Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}^d$ with unique kernel $k(\theta,\theta')$.

Theorem 3. *For $\theta_0\sim p_0(\theta)$, each time a new x_n is observed, let $\theta_n=\boldsymbol{T}^n(\theta_{n-1})=\theta_{n-1}+\epsilon\boldsymbol{\phi}^*_{\Pi,\pi_n}(\theta_{n-1})$ be a sequential updating procedure using steepest descent direction $\boldsymbol{\phi}^*_{\Pi,\pi_n}(\cdot)$. Then $\mathrm{KL}\big(\Pi_{[\boldsymbol{T}^{n+1}]}\|\pi_{n+1}\big)\le\mathrm{KL}\big(\Pi_{[\boldsymbol{T}^n]}\|\pi_n\big)$.*

We apply TSVGD for the inference of the posterior of $\boldsymbol{\kappa}_t$ in our vMF filter (1). **Theorem 3 shows the empirical distribution $\{\boldsymbol{\kappa}^s_t\}^S_{s=1}$ will gradually approach the true posterior** $p(\boldsymbol{\kappa}_t\mid\mathbf{x}_{t-1})$**.** Using the above results, we obtain the progressive updating formula for $\boldsymbol{\kappa}_t$ as follows.

Corollary 1. *Given $\{\boldsymbol{\kappa}^s_t\}^S_{s=1}$ from the variational posterior of $\boldsymbol{\kappa}_t$ at the pixel x_t, the update of TSVGD at the pixel x_{t+1} is given by $\boldsymbol{\kappa}^s_{t+1}\leftarrow\boldsymbol{\kappa}^s_t+\tau\hat{\phi}^*(\boldsymbol{\kappa}^s_t)$, where τ is the pre-defined stepsize and*

$$\hat{\phi}^*(\boldsymbol{\kappa})=\frac{1}{S}\sum_{s=1}^{S}\Bigg[k(\boldsymbol{\kappa}^s_t,\boldsymbol{\kappa})\nabla_{\boldsymbol{\kappa}^s_t}\log\Big[q(x_t\,|\,\boldsymbol{\kappa}^s_t,\boldsymbol{P}_t)\cdot\mathop{\mathbb{E}}_{\boldsymbol{\kappa}_t}p(\boldsymbol{\kappa}^s_t\,|\,\boldsymbol{\kappa}_t)\Big] + \frac{1}{T}\nabla_{\boldsymbol{\kappa}^s_t}k(\boldsymbol{\kappa}^s_t,\boldsymbol{\kappa})\Bigg]. \tag{5}$$

Algorithm 1: Unbiased Prediction using DAMix

Input: Maximum sequence length T; Number of selected models M; Measurement dimension D; The observed measurements tensor $\mathbf{p}_1\in\mathbb{R}^{M\times D}$ for the first pixel; Step size τ; Number of posterior samples S;

Output: Posterior sample means $\{\bar{\boldsymbol{\mu}}_t\}^T_{t=1}=\{\frac{1}{S}\sum_s\boldsymbol{\mu}^s_t\}^T_{t=1}$ and $\{\bar{\boldsymbol{\kappa}}_t\}^T_{t=1}=\{\frac{1}{S}\sum_s\boldsymbol{\kappa}^s_t\}^T_{t=1}$;

1: Initialization of $\{\boldsymbol{\mu}^s_0\}^S_{s=1}$ as $\boldsymbol{\mu}^s_0=\frac{1}{M}\sum_{m=1}^M\mathbf{p}^m_1$; Initialization of $\{\boldsymbol{\kappa}^s_1\}^S_{s=1}$;
2: **for** $1\le t\le T$ **do**
3: Update $\boldsymbol{\lambda}^s_t=\bar{\kappa}^0_t\cdot\boldsymbol{\mu}^s_{t-1}+\sum_{m=1}^M\bar{\kappa}^m_t\cdot\mathbf{p}^m_t$;
4: Sampling $\{\boldsymbol{\mu}^s_t\}^S_{s=1}$ from the mixture distribution of Equation (4) based on $\{\boldsymbol{\mu}^s_{t-1}\}^S_{s=1}$;
5: Observe the true pixel value x_t at time t and predict $\mathbf{p}_{t+1}$;
6: Updating variational samples $\{\boldsymbol{\kappa}^s_t\}^S_{s=1}$ to new $\{\boldsymbol{\kappa}^s_{t+1}\}^S_{s=1}$ using TSVGD according to Equation (5);
7: Normalize $\{\boldsymbol{\kappa}^s_{t+1}\}^S_{s=1}$ such that $\sum_{s=1}^S\boldsymbol{\kappa}^s_{t+1}=\sum_{s=1}^S\boldsymbol{\kappa}^s_t$;
8: **end for**

The complete algorithm for model aggregation is given in Algorithm 1. Although Bayesian methods usually inherit high computational complexity, Algorithm 1 does not have this problem and enjoys a solid theoretical foundation. Theorem 2 indicates each updating step only involves current values of $\mathbf{p}_t$ and x_t. Thus, **the complexity of TSVGD is much smaller than traditional Bayesian methods** (Andrieu et al. 2003; Liu and Wang 2016) which involve full data iterations. The direct implementation of Algorithm 1 has a complexity per-step of $\mathcal{O}\left((M+1)DS+S^2\right)$ and overall complexity $\mathcal{O}\left((M+1)TDS+TS^2\right)$. Typically for a grayscale image patch with $T=1024$, $M=5$, $D=256$ and $S=3$, the complexity is around 10^7. Notice that we can further reduce the computational cost of T times for-loop (Step 2-8 in Algorithm 1). Since nearby pixel values are basically the same, it is unnecessary to update the posterior for every pixel. Then the for-loop can only update over $\mathbf{p}_t$ and x_t at the fixed interval. In our experiments, the time-consuming Bayesian approach is fast enough, decreasing the processing time for each image patch to around 3 seconds.

Experiments

In this section, we evaluate DAMix on 22 datasets, including both low-resolution and high-resolution images, to demonstrate the OoD generalization performance of DAMix. We first compare the compression performance of DAMix with that of other neural compression methods and that of a single PixelCNN++ (Salimans et al. 2017) pre-trained by ourselves. We show the effectiveness and robustness of DAMix in terms of compression ratio on in-distribution and OoD datasets. And then we conduct ablation studies to examine the contributions of the proposed model selection and vMF filter.

Implementation

Datasets. Following the previous lossless compression works (Ho et al. 2019; van den Berg et al. 2021; Zhang et al. 2021b; Kang et al. 2022), we conduct experiments on CIFAR10 (Krizhevsky and Hinton 2009),

	Dataset	PNG	FLIF	JPEG2000	L3C	iVPF	iFlow	PILC	Pixel CNN++ [1]	Pixel CNN++ [2]	Pixel CNN++ [3]	DAMix
In-Distribution	ImageNet32	6.41	5.06	7.50	5.19	4.03	3.88	5.10	4.32	4.31	3.97	3.98
	CIFAR10	5.91	4.27	6.75	4.55	3.49	3.36	4.23	2.95	2.96	3.29	2.92
	CIFAR100	5.82	4.41	5.43	4.26	3.51	3.36	4.23	2.99	2.98	3.30	2.95
	GTA5	3.17	2.26	2.57	2.87	2.10	1.86	2.91	2.06	2.05	1.86	1.80
	Camelyon17D1	5.27	4.85	5.03	5.19	4.45	4.34	5.09	4.44	4.42	4.35	3.25
	RxRx1D0	2.20	1.66	2.22	2.10	1.49	1.23	2.29	1.40	1.34	1.22	1.10
	DIV2K	4.23	3.24	4.11	3.13	2.60	2.77	3.41	2.61	2.60	2.55	2.51
Out-of-Distribution	CLIC.mobile	3.80	2.82	3.94	2.65	2.47	2.26	3.00	2.23	2.22	2.17	2.12
	CLIC.pro	3.90	3.03	3.79	2.96	2.63	2.45	3.23	2.46	2.45	2.39	2.34
	SYNTHIA	4.11	2.71	3.12	3.57	2.49	2.28	3.37	2.54	2.51	2.30	2.29
	Camelyon17D0	4.73	4.48	4.75	4.67	4.03	3.92	4.70	3.93	3.92	3.89	2.74
	Camelyon17D2	5.43	5.23	5.43	5.48	4.81	4.73	5.42	4.80	4.79	4.73	3.49
	Camelyon17D3	5.03	5.30	5.61	6.00	4.83	4.75	5.74	4.90	4.87	4.74	3.54
	Camelyon17D4	4.82	4.67	4.85	4.92	4.27	4.16	4.92	4.17	4.16	4.12	2.83
	Cityscapes	2.99	2.19	2.38	2.42	2.13	1.93	2.64	1.95	1.94	1.89	1.88
	Glomeruli	2.55	2.14	2.42	2.68	2.16	1.92	2.91	1.89	1.85	1.76	1.43
	RxRx1D1	2.45	1.82	2.52	2.25	1.65	1.37	2.50	1.63	1.55	1.38	1.26
	RxRx1D2	2.27	1.76	2.27	2.07	1.54	1.26	2.34	1.47	1.40	1.25	1.15
	RxRx1D3	2.06	1.54	1.96	1.94	1.44	1.16	2.19	1.33	1.27	1.15	1.04
	GlobalWheat	3.32	3.30	3.71	3.62	3.35	3.20	3.89	3.27	3.25	3.15	3.12
	Manga109	3.84	2.71	3.10	3.39	2.67	2.43	3.46	2.37	2.36	2.25	2.22
	Urban100	4.49	3.02	3.40	3.53	2.98	2.78	3.74	2.84	2.82	2.69	2.66

[1] Trained on CIFAR10; [2] Trained on CIFAR100; [3] Trained on ImageNet32.

Table 1: Compression performance in BPD on 22 datasets.

ImageNet32 (Chrabaszcz, Loshchilov, and Hutter 2017), CLIC.mobile, CLIC.pro, and DIV2K (Agustsson and Timofte 2017). In addition to these commonly used datasets, we have collected datasets with diverse distributions to better examine the proposed method. Specifically, we use CIFAR100 (Krizhevsky and Hinton 2009), GTA5 (Richter et al. 2016) (car perspective in the streets of virtual cities), Camelyon17 (Koh et al. 2021) (regions of tissues), RxRx1 (Koh et al. 2021) (cells obtained by fluorescent microscopy), SYNTHIA (Ros et al. 2016) (multi-viewpoint of a virtual city), Cityscapes (Cordts et al. 2016) (urban street scenes), Glomeruli (Bueno et al. 2020) (regions of tissues), GlobalWheat (Koh et al. 2021) (wheat fields), Manga109 (Matsui et al. 2017) (manga volumes), Urban100 (Huang, Singh, and Ahuja 2015) (urban scenes). Camelyon17 and RxRx1 are used to evaluate the domain generalization performance of deep models in (Koh et al. 2021), and we split them into five (Camelyon17D0 - Camelyon17D4) and four (RxRx1D0 - RxRx1D3) subsets according to the domains proposed in (Koh et al. 2021), respectively. For datasets that have already been divided into training and test sets, we divide them in the original way, while for other datasets, we split them into 80% training and 20% test. We sample 100 images from the test set for each dataset (21 for Manga109 and 20 for Urban100 because of the limited number of total images) to form the final test data.

Model zoo. DAMix is a general framework that can be used for various deep autoregressive models. We choose the most representative one, PixelCNN++ (Salimans et al. 2017), as the basic model to form the model zoo. For each dataset, we pre-train a PixelCNN++ on 32×32 patches sampled from the training images. We follow the original settings in PixelCNN++ (Salimans et al. 2017) to train the models. In order to expand diversity, we select 7 models trained on ImageNet32, CIFAR10, CIFAR100, GTA5, Camelyon17D1, RxRx1D0, and DIV2K to form the model zoo and test on 22 datasets. We regard those test data from the datasets involved in training the models in the zoo as in-distribution data and the other test data as OoD data.

Test phase. We first select suitable expert models using the evaluation score in the test phase. Given a test image, a 32×32 patch is sampled from the center of the image. Evaluated on this patch, the pre-trained models can get the estimated negative log-likelihood $\{l_i\}_{i=1}^{7}$. The evaluation score $s_i = p_i / \max\{p_i\}$, where $p_i = -\log(l'_i / \sum_i l'_i)$ and $l'_i = l_i - \min\{l_i\}$. The model with $s_i > 0.25$ is selected. Then test image is divided into 32×32 patches. For each patch, we use the same initialization of $\boldsymbol{\kappa}_1$, i.e., $\kappa_1^0 = 1 \times 10^4$ and $\kappa_1^i = s_i \cdot 6 \times 10^4$. After obtaining the values of $\boldsymbol{\kappa}_1$, the sample set $\{\boldsymbol{\kappa}_1^s\}_{s=1}^{S}$ is initialized for TSVGD by adding Gaussian noise for each $\boldsymbol{\kappa}_1^s = \boldsymbol{\kappa}_1 + \boldsymbol{\epsilon}^s$, where S is set to 3. For $\boldsymbol{\mu}_0$, we simply use the mean value of predictions $\boldsymbol{P}_1$ of the selected models on each patch as a non-informative start. The measurement dimension D is 256 and step size τ is set to 5×10^2. The experiment is conducted with PyTorch framework using one Tesla V100 GPU.

Main Results

To demonstrate the effectiveness and robustness of DAMix, we compare DAMix with a variety of conventional methods, including PNG (Boutell 1997), FLIF (Sneyers and Wuille 2016), and JPEG2000 (Taubman and Marcellin 2002), and neural compression methods, including L3C (Mentzer et al. 2019), iVPF (Zhang et al. 2021c), iFlow (Zhang et al. 2021b), PILC (Kang et al. 2022), and single PixelCNN++ (Salimans et al. 2017) model. For iVPF and iFlow, we adopt the models trained on ImageNet32 while for PILC, we adopt the model trained on Open Image (Kuznetsova et al. 2020).

The compression results in terms of average Bit Per Dimension (BPD) are reported in Table 1. It can be seen that DAMix outperforms other methods on all datasets except on Ima-

PixelCNN++ Pre-trained on	Improvement
ImageNet32	13.3%
CIFAR10	19.0%
CIFAR100	17.6%
GTA5	20.2%
Camelyon17D1	81.0%
RxRx1D0	71.4%
DIV2K	17.9%

Table 2: Relative improvement of DAMix in terms of compression ratio averaged on 22 datasets.

geNet32. On CIFAR10 and CIFAR100, compared with other methods, PixelCNN++ achieves very good results. On this basis, our method achieves even better results by adaptively aggregating the predictions of multiple models. This shows that although the other pre-trained models perform worse than PixelCNN++ trained on CIFAR10/100 on average, they can also contribute to the final results when adaptively aggregating them to predict pixel-by-pixel. ImageNet32 pre-trained models are often selected to test generalizability (Zhang et al. 2021c,b). Compared with PixelCNN++ trained on ImageNet32, DAMix achieves superior performance on all OoD data and has at most 45.6% relative improvement in terms of compression ratio on Camelyon17D4. Table 2 summarizes the relative improvement of DAMix averaged on 22 datasets compared with the single models in the zoo. Moreover, DAMix is more stable and consistent on all datasets. PixelCNN++ trained on ImageNet32 performs poorly on Camelyon17 and RxRx1, since the distributions of these two datasets are quite different from that of ImageNet32. By leveraging a zoo of pre-trained models, neural compression is likely to deal with data of diverse distributions.

Ablation Studies

Model evaluation score. Our method selects expert models by evaluating them on a sampled 32×32 patch for every image. To examine the effectiveness of this evaluating method, we first compare it with selecting expert models for every 32×32 patch (**Fine-selection**). Then we use all the models in the zoo and 1) assign them the same initial score for every patch (**Uniform**); 2) assign them the same initial score for the first patch of every image and update the weights across the entire image (**Uniform-trans**). The results are illustrated in Table 3. Fine selection provides a more granular evaluation of the model, but increases the computing cost, especially when the number of pre-trained models is large. Compared with it, our simple evaluation approach achieves almost the same results. Due to the effectiveness of the following adaptive aggregation of the expert models, DAMix performs well even in the case of a biased initial evaluation score. We observe the performance degradation when using Uniform, which shows model selection is necessary. Uniform-trans improves the performance of Uniform on most of the datasets. However, when using Uniform-trans, the computations between different patches can only be serial but not parallel.

Model mixing. In the framework of DAMix, the proposed vMF filter can be regarded as model ensemble. We conduct contrast experiments by using linear mixing and adapting the mixing weights online to the most accurate models(**Linear-mixing**). We also compare our method with selecting the best single model in the zoo for every patch of the image (**Best-single**). The results are reported in Table 3. We observe that DAMix performs consistently better than Linear-mixing. However, Linear-mixing still outperforms single PixelCNN++ trained on ImageNet32 on most of the datasets, which shows that even with a simple model mixing approach, using a model zoo can still provide generalization gains over a single model. Even when compared with Best-single, our method achieves performance gains on part of the datasets. Best-single computes all the models in the zoo on every patch while our method involves a small subset of the zoo in the subsequent aggregation. Note that the results of our method only leverage the information of G_t, while more prior information can be used in mixing weight updating like image type and pixel locations, which will surely improve the performance reported here.

	Dataset	Fine-selection	Uniform	Uniform-trans	Linear-mixing	Best-single	DAMix
In-Distribution	ImageNet32	3.98	4.27	4.27	4.10	3.97	3.98
	CIFAR10	2.92	3.33	3.33	3.00	2.94	2.92
	CIFAR100	2.95	3.34	3.34	3.02	2.97	2.95
	GTA5	1.80	1.99	1.82	1.85	1.80	1.80
	Camelyon17D1	3.25	3.61	3.28	3.27	3.25	3.25
	RxRx1D0	1.09	1.23	1.08	1.13	1.08	1.10
	DIV2K	2.50	2.64	2.53	2.54	2.52	2.51
Out-of-Distribution	CLIC.mobile	2.11	2.27	2.18	2.15	2.13	2.12
	CLIC.pro	2.33	2.46	2.47	2.37	2.35	2.34
	SYNTHIA	2.27	2.47	2.33	2.33	2.29	2.29
	Camelyon17D0	2.74	3.08	2.77	2.75	2.74	2.74
	Cityscapes	1.88	1.99	1.92	1.91	1.89	1.88
	Glomeruli	1.42	1.56	1.69	1.48	1.42	1.43
	RxRx1D1	1.26	1.40	1.25	1.32	1.25	1.26
	GlobalWheat	3.11	3.25	3.27	3.14	3.13	3.12
	Manga109	2.20	2.35	2.56	2.26	2.24	2.22
	Urban100	2.65	2.80	2.90	2.70	2.67	2.66

Table 3: Ablation studies on model evaluation and vMF filter.

Conclusion

In this work, we propose DAMix, a zoo of Deep Autoregressive models associated with expert models aggregation for lossless compression. Through extensive experiments, we show the potential of building a collection of expert models trained on local image patches for handling OoD data. A simple log-likelihood score is proposed to evaluate each expert model and a subset of promising experts is selected to recover unbiased density using a vMF filter. To adaptively adjust the mixing weights, a novel TSVGD is proposed as a general Bayesian inference method with theoretical guarantees. Experimental results show DAMix achieves a much higher compression ratio on both low and high-resolution images than previous neural compression models trained on a large-scale dataset. One limitation of the current work is that the proposed model aggregation is only accessible to autoregressive models. However, recent advances in generative models, such as VAEs (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014; Ho, Lohn, and Abbeel 2019) and flow models (Rezende and Mohamed 2015; Tran et al. 2019), show great potential for data compression. We will consider the aggregation of different types of generative models for future work.

References

Agustsson, E.; and Timofte, R. 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 126–135.

Ahmed, R.; Islam, M. S.; and Uddin, J. 2018. Optimizing Apple lossless audio codec algorithm using NVIDIA CUDA architecture. *International Journal of Electrical and Computer Engineering*, 8(1): 70.

Aitchison, J. 1982. The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2): 139–160.

Alakuijala, J.; Van Asseldonk, R.; Boukortt, S.; Bruse, M.; Comșa, I.-M.; Firsching, M.; Fischbacher, T.; Kliuchnikov, E.; Gomez, S.; Obryk, R.; et al. 2019. JPEG XL next-generation image compression architecture and coding tools. In *Applications of Digital Image Processing XLII*, volume 11137, 111370K. International Society for Optics and Photonics.

Albuquerque, I.; Naik, N.; Li, J.; Keskar, N. S.; and Socher, R. 2020. Improving out-of-distribution generalization via multi-task self-supervised pretraining. *ArXiv*, abs/2003.13525.

Andrieu, C.; De Freitas, N.; Doucet, A.; and Jordan, M. I. 2003. An introduction to MCMC for machine learning. *Machine learning*, 50(1): 5–43.

Blei, D. M.; Kucukelbir, A.; and McAuliffe, J. D. 2017. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518): 859–877.

Boutell, T. 1997. PNG (Portable Network Graphics) Specification Version 1.0. *RFC*, 2083: 1–102.

Bueno, G.; Gonzalez-Lopez, L.; Garcia-Rojo, M.; Laurinavicius, A.; and Deniz, O. 2020. Data for glomeruli characterization in histopathological images. *Data in brief*, 29: 105314.

Chrabaszcz, P.; Loshchilov, I.; and Hutter, F. 2017. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*.

Collet, Y.; and Turner, C. 2016. Smaller and faster data compression with Zstandard. *Facebook Code [online]*, 1.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Davidson, T. R.; Falorsi, L.; De Cao, N.; Kipf, T.; and Tomczak, J. M. 2018. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*.

Fang, G.; Bao, Y.; Song, J.; Wang, X.; Xie, D.; Shen, C.; and Song, M. 2021. Mosaicking to Distill: Knowledge Distillation from Out-of-Domain Data. *Advances in Neural Information Processing Systems*, 34.

Gelfand, A. E.; and Smith, A. F. 1990. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410): 398–409.

Gershman, S.; Hoffman, M.; and Blei, D. 2012. Nonparametric variational inference. *arXiv preprint arXiv:1206.4665*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2021. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*.

Hendrycks, D.; Liu, X.; Wallace, E.; Dziedzic, A.; Krishnan, R.; and Song, D. X. 2020. Pretrained Transformers Improve Out-of-Distribution Robustness. In *ACL*.

Ho, J.; Chen, X.; Srinivas, A.; Duan, Y.; and Abbeel, P. 2019. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, 2722–2730. PMLR.

Ho, J.; Lohn, E.; and Abbeel, P. 2019. Compression with Flows via Local Bits-Back Coding. In *NeurIPS*, 3874–3883.

Huang, J.-B.; Singh, A.; and Ahuja, N. 2015. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5197–5206.

Kang, N.; Qiu, S.; Zhang, S.; Li, Z.; and Xia, S.-T. 2022. PILC: Practical Image Lossless Compression with an End-to-end GPU Oriented Neural Framework. *2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Knoll, B. 2007. CMIX.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Gao, I.; Lee, T.; David, E.; Stavness, I.; Guo, W.; Earnshaw, B. A.; Haque, I. S.; Beery, S.; Leskovec, J.; Kundaje, A.; Pierson, E.; Levine, S.; Finn, C.; and Liang, P. 2021. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *International Conference on Machine Learning (ICML)*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; Duerig, T.; and Ferrari, V. 2020. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Liu, C.; Zhuo, J.; Cheng, P.; Zhang, R.; and Zhu, J. 2019. Understanding and accelerating particle-based variational inference. In *International Conference on Machine Learning*, 4082–4092. PMLR.

Liu, Q.; and Wang, D. 2016. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29.

MacKay, D. J. 2003. *Information theory, inference and learning algorithms*. Cambridge university press.

Matsui, Y.; Ito, K.; Aramaki, Y.; Fujimoto, A.; Ogawa, T.; Yamasaki, T.; and Aizawa, K. 2017. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20): 21811–21838.

Mentzer, F.; Agustsson, E.; Tschannen, M.; Timofte, R.; and Gool, L. V. 2019. Practical Full Resolution Learned Lossless Image Compression. In *CVPR*, 10629–10638. Computer Vision Foundation / IEEE.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*.

Ranganath, R.; Gerrish, S.; and Blei, D. 2014. Black box variational inference. In *Artificial intelligence and statistics*, 814–822. PMLR.

Rezende, D.; and Mohamed, S. 2015. Variational inference with normalizing flows. In *International conference on machine learning*, 1530–1538. PMLR.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, 1278–1286. PMLR.

Richter, S. R.; Vineet, V.; Roth, S.; and Koltun, V. 2016. Playing for data: Ground truth from computer games. In *European conference on computer vision*, 102–118. Springer.

Rissanen, J.; and Langdon, G. G. 1979. Arithmetic Coding. *IBM J. Res. Dev.*, 23: 149–162.

Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; and Lopez, A. M. 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3234–3243.

Saeedi, A.; Kulkarni, T. D.; Mansinghka, V. K.; and Gershman, S. J. 2017. Variational particle approximations. *The Journal of Machine Learning Research*, 18(1): 2328–2356.

Salimans, T.; Karpathy, A.; Chen, X.; and Kingma, D. P. 2017. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. In *ICLR (Poster)*. OpenReview.net.

Sneyers, J.; and Wuille, P. 2016. FLIF: Free lossless image format based on MANIAC compression. In *ICIP*, 66–70. IEEE.

Taubman, D. S.; and Marcellin, M. W. 2002. *JPEG2000 - image compression fundamentals, standards and practice*, volume 642 of *The Kluwer international series in engineering and computer science*. Kluwer.

Tran, D.; Vafa, K.; Agrawal, K.; Dinh, L.; and Poole, B. 2019. Discrete flows: Invertible generative models of discrete data. *Advances in Neural Information Processing Systems*, 32.

Uria, B.; Murray, I.; and Larochelle, H. 2013. RNADE: The real-valued neural autoregressive density-estimator. *Advances in Neural Information Processing Systems*, 26.

van den Berg, R.; Gritsenko, A. A.; Dehghani, M.; Sønderby, C. K.; and Salimans, T. 2021. IDF++: Analyzing and Improving Integer Discrete Flows for Lossless Compression. In *ICLR*. OpenReview.net.

Van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; Graves, A.; et al. 2016. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29.

Veness, J.; Lattimore, T.; Bhoopchand, A.; Grabska-Barwinska, A.; Mattern, C.; and Toth, P. 2017. Online Learning with Gated Linear Networks. *CoRR*, abs/1712.01897.

Welch, G.; Bishop, G.; et al. 1995. An introduction to the Kalman filter.

Yi, M.; Hou, L.; Sun, J.; Shang, L.; Jiang, X.; Liu, Q.; and Ma, Z.-M. 2021. Improved OOD Generalization via Adversarial Training and Pre-training. In *ICML*.

Yu, Y.; Jiang, H.; Bahri, D.; Mobahi, H.; Kim, S.; Rawat, A. S.; Veit, A.; and Ma, Y. 2021. An Empirical Study of Pre-trained Vision Models on Out-of-distribution Generalization. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*.

Zhang, C.; Zhang, S.; Carlucci, F. M.; and Li, Z. 2021a. OSOA: One-Shot Online Adaptation of Deep Generative Models for Lossless Compression. *Advances in Neural Information Processing Systems*, 34.

Zhang, M.; Zhang, A.; and McDonagh, S. 2021. On the out-of-distribution generalization of probabilistic image modelling. *Advances in Neural Information Processing Systems*, 34.

Zhang, S.; Kang, N.; Ryder, T.; and Li, Z. 2021b. iFlow: Numerically Invertible Flows for Efficient Lossless Compression via a Uniform Coder. *Advances in Neural Information Processing Systems*, 34.

Zhang, S.; Zhang, C.; Kang, N.; and Li, Z. 2021c. iVPF: Numerical invertible volume preserving flow for efficient lossless compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 620–629.