CHANNEL-WISE AUTOREGRESSIVE ENTROPY MODELS FOR LEARNED IMAGE COMPRESSION

David Minnen & Saurabh Singh

Google Research, Mountain View, CA 94043, USA

ABSTRACT

In learning-based approaches to image compression, codecs are developed by optimizing a computational model to minimize a rate-distortion objective. Currently, the most effective learned image codecs take the form of an entropy-constrained autoencoder with an entropy model that uses both forward and backward adaptation. Forward adaptation makes use of side information and can be efficiently integrated into a deep neural network. In contrast, backward adaptation typically makes predictions based on the causal context of each symbol, which requires serial processing that prevents efficient GPU / TPU utilization. We introduce two enhancements, channel-conditioning and latent residual prediction, that lead to network architectures with better rate-distortion performance than existing context-adaptive models while minimizing serial processing. Empirically, we see an average rate savings of 6.7% on the Kodak image set and 11.4% on the Tecnick image set compared to a context-adaptive baseline model. At low bit rates, where the improvements are most effective, our model saves up to 18% over the baseline and outperforms hand-engineered codecs like BPG by up to 25%.

Index Terms**—** Image Compression, Neural Networks, Adaptive Entropy Modeling

1. INTRODUCTION

Most recent research in learned image compression uses deep neural networks, and a wide range of model architectures have been explored including recurrent networks [1]–[4] and autoencoders with an entropy-constrained bottleneck [5]–[16]. In models that use an autoencoder, an *analysis* network transforms pixels into a quantized latent representation suitable for compression by standard entropy coding algorithms, while a *synthesis* network is jointly optimized to transform the latent representation back into pixels.

To date, the most effective models make use of both *forward* and *backward-adaptive* components to improve the predictive power of the entropy model, which leads to higher compression rates without increasing distortion. Forward-adaption typically makes use of side information, for example in the form of local histograms over the quantized latent representation [9] or a learned hyperprior [10]. The hyperprior approach is particularly popular since it can easily be integrated into an end-to-end optimized network and allows for efficient encoding and decoding.

Backward-adaptation, on the other hand, typically incorporates predictions from the causal context of each symbol, *i.e.* neighboring symbols above and to the left of the current symbol as well as symbols in previously decoded channels [11]–[14]. In such context-adaptive models, encoding can still be performed efficiently using masked convolution, which will run in parallel across the entire latent tensor on a GPU or TPU [17]. Decoding, however, is inherently serial, and thus does not effectively utilize massively parallel hardware.

Our goal is to develop an image compression architecture capable of matching the rate-distortion (RD) performance of a context-adaptive model while minimizing serial processing that can lead to slow decoding times. Toward this goal, we explore two architectural enhancements: channel-conditioning (CC) and latent residual prediction (LRP). In addition, we show how training synthesis transforms with rounded latent values interacts positively with CC and LRP to further boost RD performance.

The combined effect of these improvements is a highly parallelizable architecture that outperforms recently proposed context-adaptive models [12]–[14] by 6.7% on Kodak [18] and 11.4% on the Tecnick image set [19]. We see even larger gains compared to standard codecs and learning-based models that do not use context (see Figures 2 and 3). The coding improvements provided by CC and LRP are most effective at low bit rates where our model saves more than 16% compared to the context-adaptive baseline and as much as 25% relative to BPG [20]. The following three sections describe channel-conditioning, latent residual prediction, and round-based training. A detailed analysis of the empirical results is presented in Section 5 and discussed in Section 6.

2. CHANNEL-CONDITIONAL ENTROPY MODELS

Our model builds on the hyperprior architecture introduced in [10]. This model learns to generate an image-dependent hyper-latent tensor that is compressed and transmitted as side information. It jointly learns to transform this tensor into the entropy parameters used to compress the symbols that repre-

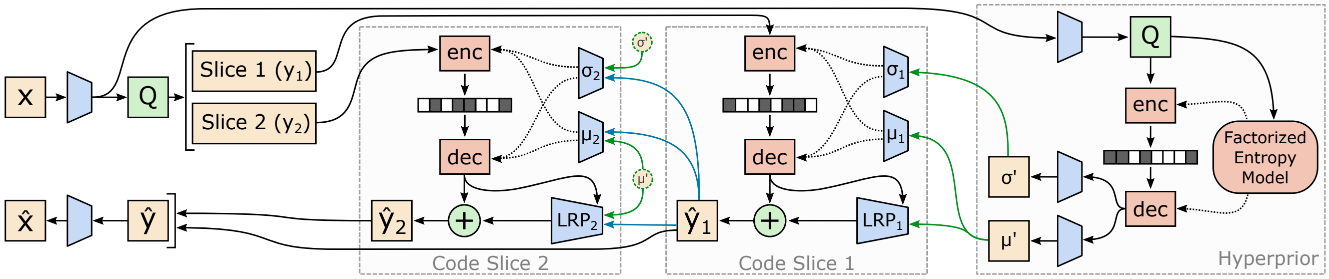

Fig. 1: This data-flow diagram shows the architecture of our compression model with latent residual prediction (LRP) and two slices for channel-conditioning (CC). Tan blocks represent data tensors, blue represents transforms comprised of convolutional layers, green is for basic arithmetic operations, and red represents entropy coding. In this model, an input image (x) is transformed and quantized before the latent representation (y) is split along the channel dimension. The first slice (y_1) is compressed using a Gaussian entropy model conditioned solely on the hyperprior (green arrows from μ' and σ'), while the entropy model for the second slice (y_2) is conditioned on both the hyperprior and the decoded symbols in the first slice (blue arrows from $\hat{y}_1$). After range coding (`enc` and `dec` blocks), quantization error is reduced by adding the predicted residual ($\texttt{LRP}_1$ and $\texttt{LRP}_2$), which is conditioned on the hyperprior via μ'. Finally, the decoded slices are concatenated to form $\hat{y}$ and transformed into the final reconstructed image ($\hat{x}$).

sents the input image (see the *Hyperprior* block at the right of Figure 1). Hyperprior models typically use a conditional Gaussian model parameterized by scale [10] or both scale and mean, and the most effective models combine information from the hyperprior (forward-adaptation) with a spatially autoregressive model (backward-adaptation) before predicting the entropy parameters μ and σ [12]–[14].

Conditioning on the causal context allows for better modeling of spatial correlation and is commonly used in standard image codecs [20]–[22] and for intra-frame prediction in video codecs [23]–[25]. In a learning-based codec, the model must estimate the parameters of a spatially autoregressive (AR) model. This approach is effective but requires running the AR model sequentially to decode each symbol, which can slow down decoding times on GPUs and TPUs compared to architectures that better utilize the massively parallel processing abilities of such hardware. For this reason, we explore channel-conditional (CC) models, which split the latent tensor along the channel dimension into N roughly equal-size slices, and conditions the entropy parameters for each slice on previously decoded slices.

Figure 1 provides a high-level overview of this architecture where the blue arrows show how y_2 (the second slice) is conditioned on $\hat{y}_1$ (the first slice). In a model with more splits, the third slice (y_3) would be conditioned on the hyperprior along with both $\hat{y}_1$ and $\hat{y}_2$, *etc*.

We can interpret CC models as autoregressive along the channel dimension rather than the spatial dimensions. Although this structure also introduces some serial processing (slice y_i can only be decoded after slices $[y_1 \ldots y_{i-1}]$), we typically use relatively few slices due to diminishing benefits to RD performance (see Figure 4). Note that in a model with N slices, each slice contains $W \times H \times \frac{C}{N}$ values that can be processed in parallel (where W, H, and C correspond to the width, height and number of channels, respectively). Contrast with a spatially autoregressive model where a naive implementation requires $W \times H$ sequential steps with only C values computed during each run. A more careful implementation using wavefront processing adds some parallelization [26] but still far less than channel-conditioning.

3. LATENT RESIDUAL PREDICTION

Autoencoder models learn to transform pixel values (x) into real-valued latents (y) that are quantized before they are losslessly compressed. This process inevitably leads to a residual error in the latent space ($r = y - Q[y]$) that manifests as extra distortion when $Q[y]$ is transformed back into the pixel domain ($\hat{x}$).

Latent residual prediction attempts to reduce this quantization error by predicting the residual based on the hyperprior and any previously decoded slices. The predicted residual is added to the quantized latents slice-by-slice, which allows LRP to improve results both by decreasing distortion and by decreasing entropy since the entropy parameters used to code later slices are conditioned on previous ones that include LRP.

Previous approaches for augmenting the input to the synthesis transform either re-used the mean prediction directly [15] or used dilated convolution to provide additional features based on a larger receptive field [16]. In both cases, however, the extra features were concatenated with the latent tensor, which increases computation, and neither used channel-conditioning, which means that potential improvements could only affect distortion.

4. TRAINING WITH ROUNDED LATENT VALUES

All compression models trained using gradient-based optimization are hindered by quantization, which yields gradi-

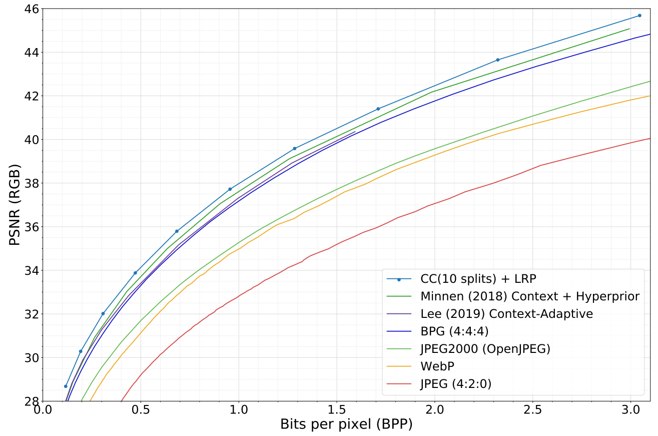

Fig. 2: Models using channel-conditioning and latent residual prediction outperform both the learning-based baselines and standard codecs on the Kodak image set.

ents that are either zero or infinite at all values. Typically, researchers avoid this problem by either training with uniform noise, which simulates "noisy quantization" without destroying the gradient [6], [10], [12]–[14], [27], [28], or they use straight-through gradients where rounding is applied but the true gradient function is replaced with the identity function [5].

Although space constraints preclude a full report on the effects of different training methods, we empirically found that a mixed approach improves RD performance. Our baseline models replace quantization with uniform noise during training: $Q[y] \doteq y + \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$. The mixed approach uses the same uniform noise for learning entropy models but replaces the noisy tensor with a rounded one whenever the quantized tensor is passed to a synthesis transform. Looking at Figure 1, the difference is essentially whether the quantized tensor is flowing to the right (add noise) or left (round with straight-through gradients). We experimented with using the rounding-based method everywhere, but this approach performed worse than the noise-based baseline.

5. EXPERIMENTAL RESULTS

In this section, we evaluate the effects of using CC, LRP, and round-based training in a learned image codec. Figure 2 compares RD curves averaged over the Kodak image set [18]. The graph shows that our full model (10 CC slices + LRP + round-based training) outperforms all of the standard codecs (BPG, JPEG2000, WebP, and JPEG) as well as learning-based codecs that combine spatial context with a hyperprior [13], [14]. To improve clarity, earlier learning-based methods, including [2]–[12], are not shown in Figure 2, but all of these methods have worse RD performance than both BPG and our CC + LRP model.

Additional results are shown in Figure 3, which plots the relative rate savings compared to BPG at different quality levels. Larger values correspond to larger relative rate savings and thus better compression. This graph generalizes a Bjøntegaard Delta (BD) chart [29] by plotting rate savings as a function of quality, rather than only presenting the average savings. Our largest model, which uses 10 CC slices, provides a significant rate savings over BPG, ranging from 10% at higher quality levels up to 25% at the lowest. This corresponds to an average BD rate savings of 13.9% over BPG and 6.7% over the context-adaptive baseline [13]. The following sections analyze how each proposed improvements contributes to the final result.

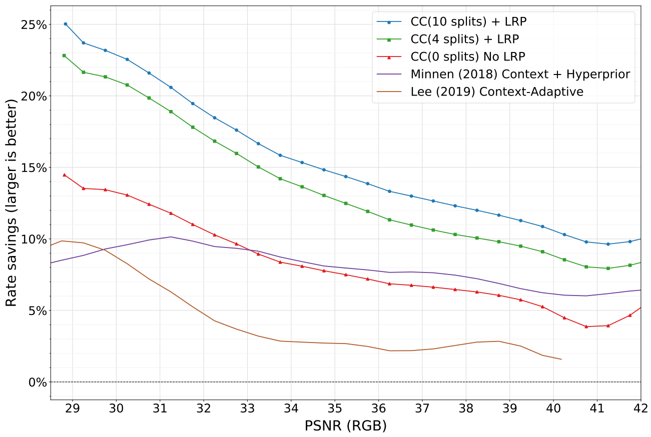

Fig. 3: Each curve shows the rate savings relative to BPG averaged over the Kodak image set. Our largest model (10 CC splits + LRP + round-based training) outperforms BPG by 10% at high bit rates and up to 25% at low bit rates.

5.1. Number of Channel-Conditional Slices

Figure 4 shows the average rate savings as the number of channel-conditioning slices increases. When we split the latent tensor into more slices, there are more opportunities to model the dependencies between channels, which reduces entropy. This benefit, however, comes at the cost of extra computation, and we also see diminishing returns as the number of slices increases.

5.2. Latent Residual Prediction

Figure 5 shows the effect of LRP for different numbers of channel-conditioning splits. Each curve compares a model trained with LRP to an identical model without LRP by plotting the relative rate savings when LRP is used.

The figure shows several effects. First, LRP has almost no benefit for models that do not use channel-conditioning, which we can see because the blue "CC(0 splits)" curve is always close to zero. Second, regardless of the number of CC splits, LRP slightly reduces RD performance at high bit rates.

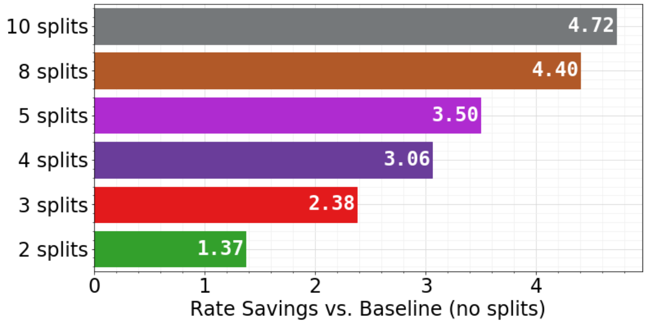

Fig. 4: RD performance increases with additional channel-conditional splits. The graph shows BD rate savings for models that are identical except for the number of CC splits. Note that these models were trained without LRP to isolate the effect of channel-conditioning.

At low bit rates, however, the benefit of LRP increases with the number of CC slices and improves compression by more than 6% for the model with 10 splits.

5.3. Rounding-based Optimization

Figure 6 shows the impact of mixed training with noise and round-based handling of quantized tensors as described in Section 4. The figure shows results for two CC models (zero and five splits) and plots both variants with and without LRP. Each curve shows the rate savings relative to an identical model optimized using uniform-noise everywhere, which means that the rate savings are due entirely to the change in how quantization is handled. We see the same trend in all cases: the benefit is minimal at higher quality levels but becomes significant at lower bit rates. For the "CC (5 splits) + LRP" model, the savings exceed 15% at the lowest bit rates.

6. DISCUSSION

From a theoretical perspective, the positive results from both CC and LRP are somewhat surprising. Ideally, the optimization process should expand the range of each channel to balance the rate-distortion trade-off, which means that using additional bits in the hyperprior to drive LRP would not be helpful. Essentially, channels that significantly reduce distortion would use more symbols, which can be interpreted as finer precision, *e.g*. consider a channel that uses values $[-1, 0, 1]$ vs. one that uses $[-100, -99, \ldots, 99, 100]$ and is scaled by $\frac{1}{100}$ in the next convolutional layer. Since the most useful channels should already have higher effective granularity, there is less opportunity for LRP to provide a benefit.

Similarly, the analysis transform would ideally learn to map pixels into a latent space such that each channel is conditionally independent given the hyperprior. If this is not the case, it means there is redundant information, which will increase entropy without reducing distortion.

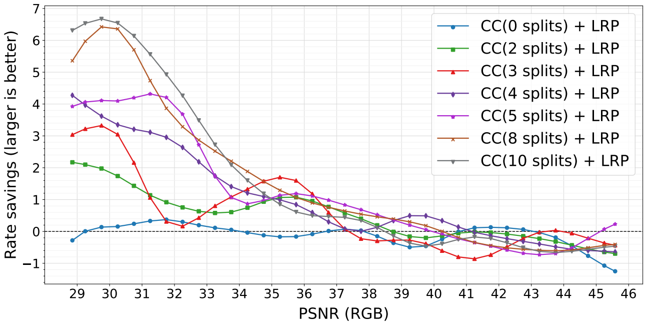

Fig. 5: Combined with channel-conditioning, latent residual prediction (LRP) helps significantly at lower bit rates but reduces performance slightly at the highest bit rates.

Empirically, we see significant improvements using both CC and LRP, which implies that existing models are far from ideal. Further research is needed to understand why the models are failing to reach an optimal state, but we can theorize that the relatively simple 4-layer convolutional networks that make up the analysis and synthesis transforms lack the capacity to generate/decode a latent representation with conditionally independent channels. Alternatively, the networks may have the necessary capacity, but our learning procedure, which uses the Adam optimizer [30], is unable to find a suitable minimum despite training for five million steps.

By combining channel-conditioning, latent residual prediction, and round-based training, we have developed a neural image compression architecture that outperforms a corresponding context-adaptive model while minimizing serial processing. In future research, we plan to investigate combining channel-conditioning with spatial context modeling to see if the two approaches are complementary.

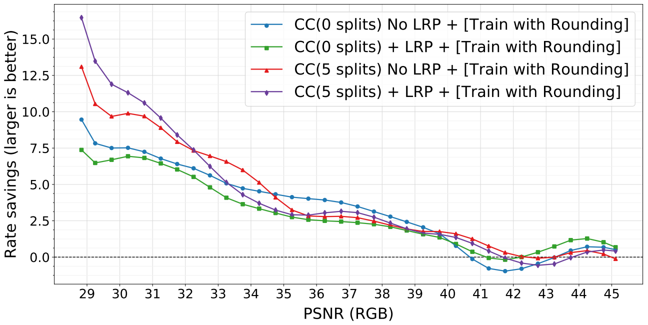

Fig. 6: Each curve shows the average rate savings on the Kodak image set when training part of the model with rounded values vs. using uniform noise everywhere (see Section 4 for details). At low and moderate bit rates, there is a significant benefit to round-based training.

References

[1] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] M. H. Baig, V. Koltun, and L. Torresani, "Learning to inpaint for image compression," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 1246–1255.

[3] D. Minnen, G. Toderici, M. Covell, T. Chinen, N. Johnston, J. Shor, S. J. Hwang, D. Vincent, and S. Singh, "Spatially adaptive image compression using a tiled deep network," *Int. Conf. on Image Processing*, 2017.

[4] N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S. J. Hwang, J. Shor, and G. Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in *2018 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," 2017, Int. Conf. on Learning Representations.

[6] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," 2017, Int. Conf. on Learning Representations.

[7] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proc. of Machine Learning Research*, 2017.

[8] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[9] D. Minnen, G. Toderici, S. Singh, S. J. Hwang, and M. Covell, "Image-dependent local entropy models for image compression with deep networks," *Int. Conf. on Image Processing*, 2018.

[10] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *6th Int. Conf. on Learning Representations*, 2018.

[11] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. V. Gool, "Conditional probability models for deep image compression," in *2018 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] J. P. Klopp, Y.-C. F. Wang, S.-Y. Chien, and L.-G. Chen, "Learning a code-space predictor by exploiting intra-image-dependencies," in *British Machine Vision Conf.*, 2018.

[13] D. Minnen, J. Ballé, and G. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems*, 2018.

[14] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," in *Int. Conf. on Learning Representations (ICLR)*, 2019.

[15] J. Zhou, "Multi-scale and context-adaptive entropy model for image compression," in *Workshop and Challenge on Learned Image Compression at CVPR*, Jun. 2019.

[16] S. Wen, "Variational autoencoder based image compression with pyramidal features and context entropy model," in *Workshop and Challenge on Learned Image Compression at CVPR*, Jun. 2019.

[17] A. van den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves, "Conditional image generation with PixelCNN decoders," in *Advances in Neural Information Processing Systems 29*, 2016.

[18] E. Kodak, *Kodak lossless true color image suite (PhotoCD PCD0992)*. [Online]. Available: `http://r0k.us/graphics/kodak/`.

[19] N. Asuni and A. Giachetti, "TESTIMAGES: A large-scale archive for testing visual devices and basic image processing algorithms (SAMPLING 1200 RGB set)," in *STAG: Smart Tools and Apps for Graphics*, 2014. [Online]. Available: `https://sourceforge.net/projects/testimages/files/OLD/OLD_SAMPLING/testimages.zip`.

[20] F. Bellard, *BPG image format*, Accessed: 2017-01-30. [Online]. Available: `http://bellard.org/bpg/`.

[21] Google, *WebP: Compression techniques*, Accessed: 2017-01-30. [Online]. Available: `http://developers.google.com/speed/webp/docs/compression`.

[22] "Information technology–JPEG 2000 image coding system," International Organization for Standardization, Geneva, CH, Standard, Dec. 2000.

[23] *ITU-R rec. H.265 & ISO/IEC 23008-2: High efficiency video coding*, 2013.

[24] A. Grange, A. Norkin, C. Chen, C.-H. Chiang, D. Mukherjee, H. Su, J. Bankoski, J.-M. Valin, J. Han, L. Trudeau, N. Egge, P. Wilkins, P. de Rivaz, S. Parker, S. Midtskogen, T. Davies, U. B. Joshi, Y. Xu, Y. Chen, Y. Wang, and Z. Liu, "An overview of core coding tools in the AV1 video codec," 2018.

[25] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd. Wiley Publishing, 2010, ISBN: 0470516925.

[26] M. Alvarez-Mesa, C. C. Chi, B. Juurlink, V. George, and T. Schierl, "Parallel video decoding in the emerging hevc standard," in *Proceedings of Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2012)*, Kyoto, Japan, 2012.

[27] R. Zamir, *Lattice Coding for Signals and Networks*. Cambridge University Press, 2014.

[28] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimization of nonlinear transform codes for perceptual quality," in *Picture Coding Symposium (PCS), 2016*, 2016. DOI: `10.1109/PCS.2016.7906310`.

[29] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," ITU-T SG16/Q6 VCEG, Austin, TX, USA, Doc. VCEG-M33, Apr. 2001.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. on Learning Representations, ICLR*, San Diego, CA, 2015.