# GENERALIZED NESTED LATENT VARIABLE MODELS FOR LOSSY CODING APPLIED TO WIND TURBINE SCENARIOS

*Raül Pérez-Gonzalo*[1,2]*, Andreas Espersen*[2] *and Antonio Agudo*[1]

[1]Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Spain
[2]Wind Power LAB, Copenhagen, Denmark

## ABSTRACT

Rate-distortion optimization through neural networks has accomplished competitive results in compression efficiency and image quality. This learning-based approach seeks to minimize the compromise between compression rate and reconstructed image quality by automatically extracting and retaining crucial information, while discarding less critical details. A successful technique consists in introducing a deep hyperprior that operates within a 2-level nested latent variable model, enhancing compression by capturing complex data dependencies. This paper extends this concept by designing a generalized $L$-level nested generative model with a Markov chain structure. We demonstrate as $L$ increases that a trainable prior is detrimental and explore a common dimensionality along the distinct latent variables to boost compression performance. As this structured framework can represent autoregressive coders, we outperform the hyperprior model and achieve state-of-the-art performance while reducing substantially the computational cost. Our experimental evaluation is performed on wind turbine scenarios to study its application on visual inspections.

***Index Terms*—** Image Compression, Rate-distortion Loss, Nested Models, Wind Turbine, Blade Inspections.

## 1. INTRODUCTION

Lossy image compression aims to convert an image to a compressed representation by capturing its spatial redundancies. It sacrifices the ability to perfectly reconstruct the original data to achieve a higher compression ratio. In response to the ever-growing demand for efficient image storage and transmission, distinct traditional codecs have emerged. Notably, these routines are grounded in manually tailored algorithms: JPEG2000 [1] uses a wavelet-based transform, WebP [2] combines predictive coding and discrete cosine transform, HEVC [3] integrates spatial prediction and transform coding, BPG [4] employs context modeling, and VTM [5] utilizes intra and inter prediction, and block-based transform coding.

This work has been supported by the Innovation Fund Denmark under 2021 ID1044-0044A and by the project MoHuCo PID2020-120049RB-I00 funded by MCIN/AEI/10.13039/501100011033.

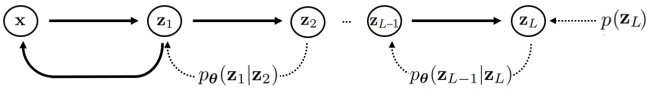



**Fig. 1**: **Generalized nested latent variable model for lossy compression.** Solid arrows signify direct calculations of the latent variables $\mathbf{z}_l$ from the encoder and the input image $\mathbf{x}$ from the decoder, while dashed arrows entail the estimation of likelihood and prior distributions.

Instead of relying on handcrafted algorithms, learning-based codecs formulate image compression as an optimization problem [6, 7]. By implementing a relaxed rate-distortion loss [8, 9], they can directly generate compressed representations of data and automatically learn to prioritize what information to retain for optimal compression performance.

A common powerful technique involves leveraging a latent variable, a concealed factor not inherently part of the input data, which strengthens the model's capacity to capture intricate data dependencies. Bls2017 [10] proposes an end-to-end optimized image compression framework that utilizes an autoencoder network and a factorized trainable prior. HP [11] extends this set-up by introducing a hyperprior, adding a second latent variable nested to the previous one. Recent advances in this field explore a coarse-to-fine architecture for enhancing conditional entropy modeling [12]. An extended coarse-to-fine approach with ResNet architecture is presented in [13], and QARV [14] further refines it with variable-rate compression by embedding the specific rate-distortion trade-off desired through adaptive normalization. Other approaches apply a context prediction module as JA [15] or others [16, 17, 18, 19], tailored non-linear transforms [20] or attention mechanisms [21] such as GMM-Anchor/-Attn [22, 23].

In this work, we extend the hyperprior concept by stacking $L$ distinct layers of latent variable [24, 25], as illustrated in Fig. 1. Therefore, we construct a generalized nested latent model with a more flexible conditional entropy model, which is easily parallelizable [26]. We demonstrate that the optimal level of nested layers depend on the rate-distortion trade-off desired. In addition to that, we showcase that for greater levels of $L$ a trainable prior is detrimental, thus, we explore a predefined logistic prior along with a common dimension for the latent variables to enhance generalized nested models.

Our approach is evaluated on a real-world industry problem. In particular, we study generalized nested models on wind turbine imagery captured during blade inspections [27, 28]. Reducing the image size without compromising its quality is crucial to properly assess wind turbines and arrange their repair [29, 30]. We showcase nested latent variables models can successfully approximate autoregressive models and, therefore, generalize them. Hence, our approach reaches competing performance compared to state-of-the-art coders on the presented challenging data, while substantially reducing its running cost, thus, it emerges as the optimal lossy coding solution for wind industry applications.

## 2. NESTED LATENT VARIABLE MODELS FOR LOSSY CODING

We aim to learn a probabilistic model $p_{\boldsymbol{\theta}}(\mathbf{x})$ of our observed data $\mathbf{x}$ to successfully apply an entropy coder capable of compressing them, where $\boldsymbol{\theta}$ denotes the model parameters. To address this learning problem, fully-observed models can be marginalized over a latent variable $\mathbf{z}_1$. Let us denote the likelihood distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1)$ and the prior distribution $p(\mathbf{z}_1)$, then, by the Bayes' rule, $p_{\boldsymbol{\theta}}(\mathbf{x})$ can be expressed as the joint distribution of the observed and latent variables as:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_1) d\mathbf{z}_1 = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1) p(\mathbf{z}_1) d\mathbf{z}_1. \quad (1)$$

This implicit representation can be recursively applied to each latent variable $\mathbf{z}_l$ to obtain an $L$-layer Markov model (see Fig. 1), where $l \in \{1, \dots, L\}$. In this way, we can sequentially gather the distinct latent dependencies to express the model evidence $p_{\boldsymbol{\theta}}(\mathbf{x})$ defined in Eq. (1) as:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1) p_{\boldsymbol{\theta}}(\mathbf{z}_{1:L-1}|\mathbf{z}_L) p(\mathbf{z}_L)\, d\mathbf{z}_{1:L}, \quad (2)$$

where we have defined for conciseness $p_{\boldsymbol{\theta}}(\mathbf{z}_{1:L-1}|\mathbf{z}_L) = \prod_{l=1}^{L-1} p_{\boldsymbol{\theta}}(\mathbf{z}_l|\mathbf{z}_{l+1})$ and $d\mathbf{z}_{1:L} = d\mathbf{z}_1 d\mathbf{z}_2 \dots d\mathbf{z}_{L-1} d\mathbf{z}_L$.

### 2.1. Generalized Relaxed Rate-distortion Loss

We seek to minimize the trade-off governed by $\lambda \in \mathbb{R}$ between the compression rate and the decompression error:

$$-\mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \left[\log p_{\boldsymbol{\theta}}(\mathbf{z}_{1:L-1}|\mathbf{z}_L) + \log p(\mathbf{z}_L)\right] + \lambda\, \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \left[d(\mathbf{x}, \hat{\mathbf{x}})\right],$$

where $d(\mathbf{x}, \hat{\mathbf{x}})$ indicates a distortion metric between the input image $\mathbf{x}$ and the reconstructed one $\hat{\mathbf{x}}$.

The presented loss operates in the continuous space, however, entropy coders can only operate with finite discrete alphabets, requiring a quantization step. In our case, we define the quantizer $Q$ as a rounding operation that is continuously approximated through uniform random noise [10]. Let $\tilde{\mathbf{z}}_l = Q(\mathbf{z}_l)$ denote the noisy approximation, then we end up with the following trainable rate-distortion loss:

$$-\mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \left[\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}_{1:L-1}|\tilde{\mathbf{z}}_L) + \log p(\tilde{\mathbf{z}}_L)\right] + \lambda\, \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \left[d(\mathbf{x}, \hat{\mathbf{x}})\right].$$

Note that the quantizer also compromises the distortion term, because $\hat{\mathbf{x}}$ relies on the latent variable $\tilde{\mathbf{z}}_1$.

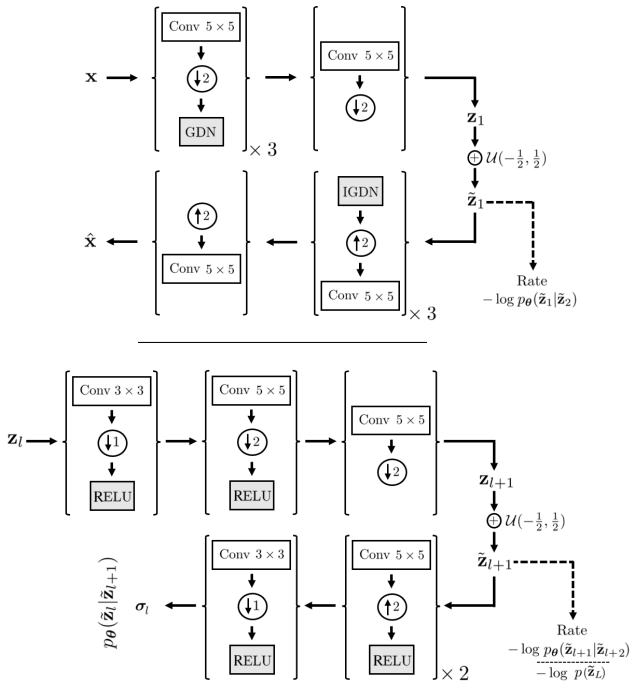



**Fig. 2**: **Proposed architecture for a generalized nested latent variable model**. In the first layer, the decoder reconstructs directly the input image $\mathbf{x}$, while it estimates the likelihood distributions in the rest of the layers. The network is built with building blocks that are composed of a convolution, a down/upsampling operation and a nonlinear function.

### 2.2. Model Architecture

We design our encoder and decoder with symmetrical structures, ensuring that the output dimension matches the input dimension. Both transforms employ a sequence of three blocks for each latent layer: a convolutional layer, a down/upsampling operation, and a non-linear activation function. For the initial layer, we utilize the generalized divisive normalization with adaptable parameters as the chosen non-linear function [10]. Subsequent layers employ ReLU.

Specifically, convolutions employ a 2-dimensional kernel size of $5 \times 5$, except those on top of the latent layers which operate with a 3-kernel. The padding and stride are meticulously selected to ensure congruence between input and output dimensions. To minimize computational time, down/upsampling operations are seamlessly integrated with linear convolutions through adjusting the convolution stride. These convolutions utilize 70 filters, except for those interacting with $\mathbf{z}_l$, which employ 150. A comprehensive illustration of the network architecture is presented in Fig. 2.

The standard logistic distribution with statistical independence across its components serves as the prior $p(\tilde{\mathbf{z}}_L)$. The likelihood $p_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}_l|\tilde{\mathbf{z}}_{l+1})$ for $l \in \{1, \dots, L-1\}$ are characterized as conditional independent zero-mean Gaussian distributions. Hence, these distributions are modelled by a convolutional network for every standard deviation $\{\boldsymbol{\sigma}_l\}_{l \in \{1,\dots,L-1\}}$.

**Fig. 3**: **Four instances of wind turbine blade images.** The pictures showcase distinct blade surfaces and locations with respect to the rotor of the turbine.

### 2.3. Compression Scheme

Each latent variable $\mathbf{z}_l = (z_{l,1}, \dots, z_{l,2^P})$ undergoes independent discretization into $2^P$ distinct bins, where $P = 10$ is a predefined precision. Within each latent layer and for every component $z_{l,j}$, the bins exhibit uniform and identical widths. Defining $B(z_{l,j})$ as the bin encompassing $z_{l,j}$, we establish the discretized value for $z_{l,j}$ as the midpoint of bin $B(z_{l,j})$. Consequently, the discretized distributions become highly smooth, particularly as the number of discretization bins increases. These distributions are combined with asymmetric numeral systems for entropy coding [31].

### 2.4. Autoregressive Universal Approximators

Let $x^{(t)}$ denote the $t$-th pixel component of $\mathbf{x}$, autoregressive models [32] leverage the previously decoded components to enhance the coding performance reconstruction of the next pixel component in order to learn $p_{\boldsymbol{\theta}}(\mathbf{x})$ from Sec. 2.

$$p_\theta(\mathbf{x}) = \prod p_\theta(x^{(t)}|x^{(t-1)}, \dots, x^{(1)}), \tag{3}$$

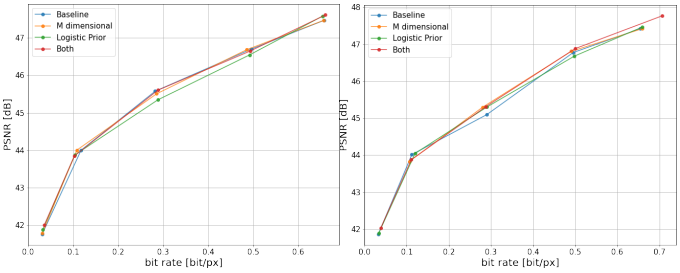
where $x^{(t-1)}, x^{(t-2)} \dots, x^{(1)}$ are the previously decoded pixel components from $\mathbf{x}$.

Without loss of generality, we assume each latent variable $z_l$ only has a single component. Nested latent variable models can approximate autoregressive models by simply decoding each pixel component through an additional latent variable. In this way, we can reproduce the sequential decoding order of an autoregressive model: the first pixel component $x_1$ is decoded by $z_L$ through $p(z_L)$, the second component $x_2$ is decoded by $z_{L-1}$ through $p(z_{L-1}|z_L)$, and so on. In general terms, the $t$-th pixel component $x^{(t)}$ is decoded by $z_{L-t+1}$ through $p(z_{L-t+1}|z_{L-t+2}, \dots, z_L)$. Hence, the model evidence $p_{\boldsymbol{\theta}}(\mathbf{x})$ in Eq. (3) can be expressed as:

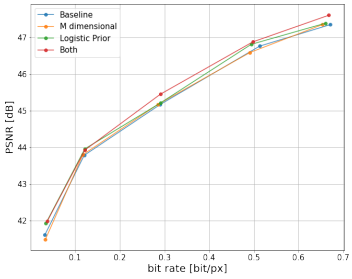
$$p_\theta(\mathbf{x}) = \prod p(z_{L-t+1}|z_{L-t+2}, \dots, z_L). \tag{4}$$

The provided rationale can be applied to approximate any autoregressive variation, including channel-wise [33] or checkerboard [34] autoregressive models. A thorough proof is detailed in [35].

Autoregressive models, while formally capable of conditioning predictions on all previous decoded components, commonly utilize a fixed context, such as $5 \times 5$ convolution kernels [15]. Thus, autoregressive lossy models can be approximated by incorporating a limited number of $z_l$ variables.

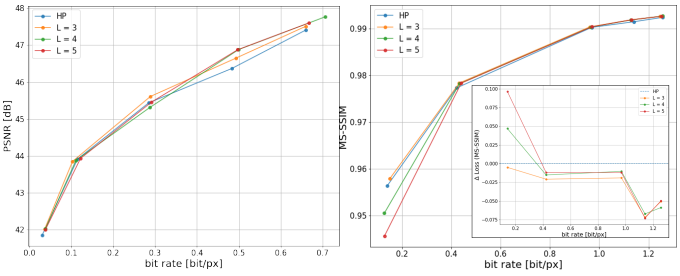



(a) $L = 3$ (b) $L = 4$




(c) $L = 5$

**Fig. 4**: **Ablation study results**. "Baseline" extends exactly the architecture from [11]; "M dimensional" denotes establishing all latents $\mathbf{z}_l$ as M-dimensional variables; "Logistic prior" indicates employing a standard logistic for $p(\mathbf{z}_L)$.




(a) PSNR quality measure (b) MS-SSIM quality measure

**Fig. 5**: **Quantitative performance comparison with respect to $L$ over the validation set**. HP [11] employs $L = 2$. Plots are divided based on the training's distortion metric. MS-SSIM graphic contains another plot comparing loss differences between each $L$ to HP [11].

## 3. EXPERIMENTAL RESULTS

### 3.1. Dataset

We possess a collection of 64,438 high-resolution blade images in raw format, each with an RGB resolution of 6,744×4,502 pixels (see examples in Fig. 3). These images are categorized into three sets: a training set encompassing ∼80% of the data, along with validation and test sets, each containing ∼10%. To ensure unbiased performance and effective generalization for new acquired data, images from the same inspection campaign are exclusively present in one set. During training, random crop selections are utilized.

### 3.2. Implementation Details

The model processes 256×256 pixel patches using an NVIDIA GeForce RTX 3080 Ti. Training employs the Adam opti-

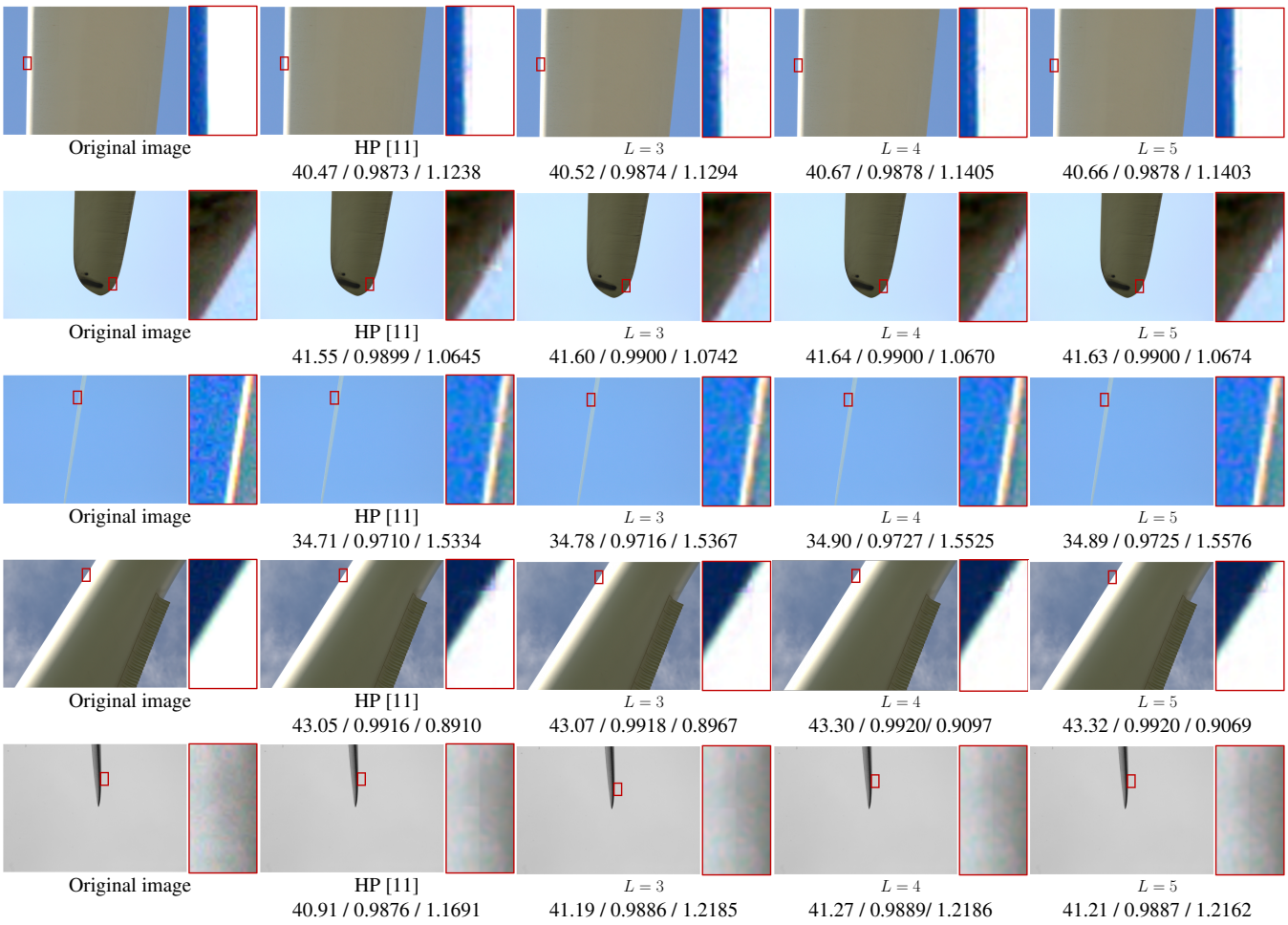



**Fig. 6**: **Visual comparison of distinct blade instances with respect to** $L$. Subcaptions denote the PSNR, MS-SSIM and bit/px. HP [11] employs $L = 2$. Zoomed region contrast is increased in the images on the first and fourth rows.

mizer [36] with an initial learning rate of $10^{-4}$. Two distortion metrics are used: Mean Squared Error (MSE) and negative Multi-Scale Structural SIMilarity (MS-SSIM) [37]. To ensure stable training and control the gradients, the rate terms have a minimum bound of $10^{-9}$. Overfitting prevention involves early-stopping [38] and diversified training data via random cropping from full-resolution images in each epoch.

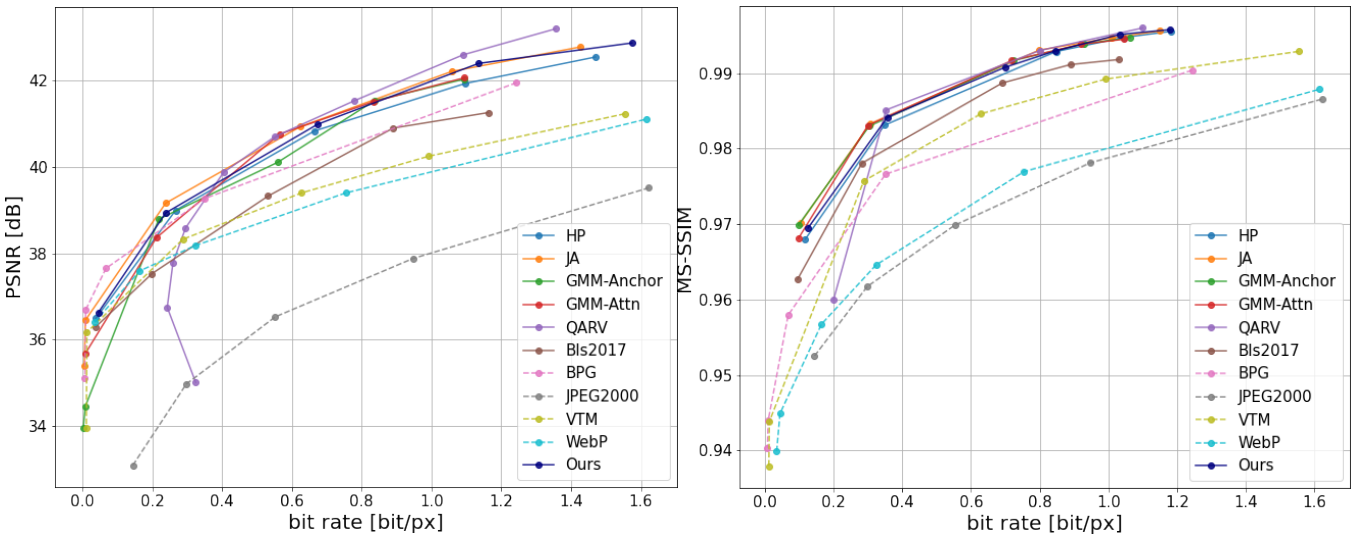
### 3.3. Ablation Study

We conduct an ablation study of the lossy neural architecture trained on MSE for $L = \{3, 4, 5\}$; presented in Section 2.2. In particular, Fig. 4 explores the performance of four distinct architectures on the validation set in terms of rate-distortion curves: the baseline extends HP [11] for any $L$, an architecture featuring equal dimensionality $M = 150$ across all latent variables, an architecture employing a standard logistic prior, and a hybrid architecture integrating the two previous ones.

Both architectural modifications significantly enhance compression performance, especially in scenarios with a large value of $L$, higher bit rates, or even more when both factors are combined. For $L = 3$, this hybrid architecture demonstrates that an equal compression performance compared to the baseline is achieved, thus, the prior entropy network [11] is no longer required to achieve top-performing results. When $L = 4$, we observe that for the highest bit rate model, the hybrid architecture surpasses the rest and, for $L = 5$, this outperforming behaviour extends to all the rate-distortion curve, showing that the prior entropy network is indeed detrimental.

The hybrid architectures of $L = \{3, 4, 5\}$ and HP [11], which corresponds to the baseline with $L = 2$, are compared in Fig. 5. The evaluation is made depending on the distortion metric used during the training: Peak Signal-to-Noise Ratio (PSNR) for $d =$MSE and MS-SSIM for $d = 1-$MS-SSIM.

Figure 5a illustrates that higher bit rates demand a larger $L$ for optimal results. Specifically, $L = \{4, 5\}$ excel on highest bit rate scenarios, $L = 3$ outperforms on intermediate bit rate cases, and comparable compression performance is exhibited for any $L$ at the lowest bit rate. This observed behavior can be attributed to the network's inherent struggle to minimize distortion versus reducing the bit rate. To elucidate, when increasing $\lambda$ and intensifying the requirement for image quality,




(a) PSNR quality measure

(b) MS-SSIM quality measure

**Fig. 7**: **Test rate-distortion curves for distinct lossy coders**. Plots are divided by the distortion metric. Best viewed in color.

the network necessitates more epochs to converge. Therefore, a higher image quality demand entails a more complex network, which translates in a larger $L$.

A similar trend is noticeable in Fig. 5b. Nevertheless, the distinctions between the architectures are less pronounced in MS-SSIM, and its graphical representation is somewhat impaired. Within this figure, we offer a more comprehensive visualization by directly comparing the losses of each $L$, revealing that the optimal $L$ increases when the bit rate does.

### 3.4. Visual Analysis

To shed light on how generalized nested models impact the visual perspective, Fig. 6 depicts blade picture instances compressed with a distinct number of latent variables $L = \{2, 3, 4, 5\}$. For $L = \{3, 4, 5\}$, the network incorporates the hybrid modifications discussed in Section 3.3. Because these pictures are high-resolution images, the figure focuses on the comparison of the distinct models on image patches with high frequency information, so we can facilitate its comparison. The contrast may be increased for the same reason.

To start with the first row, we can clearly see on the original image a perfect line boundary between the blade region and the blue sky. However, when the image is compressed using HP [11] ($L = 2$), much noisy distortion is introduced on the blade; and also on the background. When applying generalized networks, this distortion is iteratively reduced when we increase $L$. Within the same bit rate range, by utilizing $L = 3$, the distortion metrics are reduced and its perceptual view is enhanced. What is more, our 4-layer latent model substantially drops this distortion noise and even improves the quality measures. Note that for $L = 5$, we reach an equal level of compression performance as $L = 4$, showing the optimal layer depth for this rate-distortion trade-off is $L = 4$.

Another common distortion iteratively reduced when incorporating additional latent variables is blocking artifacts; caused due to compressing the images through individual image patches. As seen from the second to the last row, nested models can selectively enhance those regions. In all cases, we obtain again $L = 4$ as the optimal number of latent variables.

### 3.5. Quantitative Comparison

In order to contextualize our approach, we present a concise comparison with state-of-the-art lossy coders in Fig. 7. To this end, we selected the optimal layer depth $L$ over the validation set (Fig. 5) for each rate-distortion model to build our proposed lossy coder; which is now evaluated on the test set.

Learning-based coders highlight superior performance than traditional ones, except for BPG [4] evaluated on PSNR, which surpasses the rest on low bit rate scenarios. Our approach excels on compression performance in both quality measures, accomplishing comparable results to top-performing autoregressive coders, such as JA [15]. However, JA's autoregressive nature weakens its applicability due to its high computational cost, as we will see in Section 3.6. Despite outperforming our model in high bit rates, QARV's [14] embedding layer is not sufficient to generalize for distinct rate-distortion terms, therefore, it performs poorly on low bit rates. In addition to that, its computational cost is higher than a typical autoregressive model, as discussed later.

Focusing on the baseline codecs of our proposed generalized networks, we substantially outperform in compression performance Bls2017 [10], which corresponds to $L = 1$, and show significant improvement also compared to HP [11] (which is $L = 2$). As mentioned in Section 3.3, generalized nested models significantly enhance the compression performance on high bit rate scenarios, when the image quality requirements imply a large $L$ as optimal. In this range of image quality is relevant to accomplish a high compression performance, because they ensure an acceptable image quality to perform defect detection during blade assessments.

| Coding scheme | Compression Time | | | | Decompression Time | | | |
|---|---|---|---|---|---|---|---|---|
| | Minimum time [s] | Maximum time [s] | Mean time [s] | Slope growth [s/bit] | Minimum time [s] | Maximum time [s] | Mean time [s] | Slope growth [s/bit] |
| HP [11] | 5.76 | 6.17 | 5.96 | 0.25 | 6.51 | 15.60 | 12.83 | 5.57 |
| JA [15] | 200.68 | 204.19 | 203.01 | 1.96 | 442.74 | 458.26 | 451.08 | 8.69 |
| GMM-Anchor [22] | 227.91 | 235.96 | 233.04 | 5.51 | 480.62 | 495.27 | 487.40 | 10.02 |
| GMM-Attn [22] | 247.97 | 258.12 | 252.81 | 6.78 | 500.60 | 523.14 | 508.58 | 15.06 |
| Bls2017 [10] | 0.27 | 0.35 | 0.30 | 0.07 | 2.85 | 11.94 | 9.12 | 7.36 |
| QARV [14] | 371.08 | 374.68 | 373.28 | 2.21 | 238.08 | 240.99 | 239.70 | 1.79 |
| BPG [4] | 2.47 | 8.80 | 4.75 | 3.25 | 1.08 | 12.78 | 5.91 | 6.01 |
| JPEG2000 [1] | 7.02 | 7.03 | 7.02 | 0.01 | 24.40 | 58.48 | 38.92 | 16.66 |
| VTM [5] | 102.57 | 9943.02 | 4584.58 | 5309.94 | 1.42 | 4.32 | 2.73 | 1.57 |
| WebP [2] | 2.49 | 9.40 | 4.17 | 3.59 | 1.26 | 10.87 | 5.56 | 4.99 |
| Ours | 9.10 | 13.12 | 10.73 | 2.71 | 9.51 | 18.82 | 15.24 | 6.26 |

**Table 1**: **Running time comparison of a full-resolution image** for distinct state-of-the-art lossy coders.

### 3.6. Computational Cost

Table 1 shows the computational cost of coding a wind turbine blade image using an NVIDIA RTX 3080 Ti GPU and a 20-core Intel Core i9-10900 KF processor at 3.70 GHz, assuming pre-loaded model weights to simulate a drone inspection.

We analyzed the computational cost of adding an extra nested latent variable. Using HP [11] as a reference, our generalized hyperprior model shows a linear increase in cost with each new latent variable, as its parameter count grows linearly and the architecture structure is a pure extension. Notably, a new latent variable results in an additional demand of approximately 4s during the encoding and 2s during decoding.

In Table 1, we showcase the minimum, maximum, and average coding run times in seconds. Recognizing the typical trend of increased computational time with higher bit rates attributed to the entropy coder, we also present the growth slope observed as the bit rate increases. Our objective is to achieve a high-performing coder characterized by swift compression and decompression times, exhibiting minimal sensitivity to the desired bit rate; signified by a low slope growth.

Our nested latent variable model meets these criteria effectively, demonstrating outstanding compression performance, while not significantly increasing the coding time, except compared to Bls2017 [10]. It exhibits modest compression slope growth and maintains reasonably low average decompression time. In contrast, autoregressive models prove to be considerably slower due to their sequential contextual prediction process. Notably, QARV [14], despite not being autoregressive, fails to strike a favorable balance between coding performance and computational cost. Consequently, our approach matches autoregressive models in compression performance while reducing computational cost.

## 4. CONCLUSION

This paper extends the scope of the deep hyperprior for neural lossy coding, introducing a versatile $L$-level nested latent model. Our method captures the intricate dependencies among latent variables with greater fidelity and marked compression improvement. By carefully designing the architecture and selecting the optimal layer depth depending on the rate-distortion trade-off, these generalized models surpass the hyperprior performance without a trainable prior and successfully approximate autoregressive models, accomplishing state-of-the-art results while reducing substantially the computational cost. Our framework effectiveness is solidified through empirical evaluation in a real-world context. Specifically, we have demonstrated its applicability within visual wind turbine inspection data by yielding compelling results, serving as a testament to its robustness and practicality.

## 5. REFERENCES


[1] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *SPM*, vol. 18, no. 5, pp. 36–58, 2001.

[2] Google, "WebP: Compression techniques," `https://developers.google.com/speed/webp/docs/compression`, 2017.

[3] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *TCSVT*, vol. 22, no. 12, pp. 1649–1668, 2012.

[4] F. Bellard, "BPG image format," *SPIC*, 2015.

[5] Joint Video Exploration Team (JVET), "VVC VTM," Software available at `https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM`.

[6] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and video compression with neural networks: A review," *TCSVT*, vol. 30, no. 6, pp. 1683–1698, 2019.

[7] N. Anantrasirichai and D. Bull, "Artificial intelligence in the creative industries: a review," *AIR*, 2022.

[8] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," *ICLR*, 2016.

[9] Y. Yang, S. Mandt, L. Theis, et al., "An introduction to neural data compression," *FTCGV*, vol. 15, no. 2, pp. 113–200, 2023.

[10] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *ICLR*, 2017.

[11] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *ICLR*, 2018.

[12] Y. Hu, W. Yang, and J. Liu, "Coarse-to-fine hyper-prior modeling for learned image compression," in *AAAI*, 2020.

[13] Z. Duan, M. Lu, Z. Ma, and F. Zhu, "Lossy image compression with quantized hierarchical vaes," in *WACV*, 2023.

[14] Z. Duan, M. Lu, J. Ma, Y. Huang, Z. Ma, and F. Zhu, "QARV: Quantization-aware resnet VAE for lossy image compression," *TPAMI*, vol. 46, no. 1, pp. 436–450, 2024.

[15] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *NeurIPS*, 2018.

[16] J. Lee, S. Cho, and S. K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," *ICLR*, 2019.

[17] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, "Checkerboard context model for efficient learned image compression," in *CVPR*, 2021.

[18] A. Meyer and A. Kaup, "A novel cross-component context model for end-to-end wavelet image coding," in *ICASSP*, 2023.

[19] W. Jiang, J. Yang, Y. Zhai, P. Ning, F. Gao, and R. Wang, "Mlic: Multi-reference entropy model for learned image compression," in *ACM Multimedia*, 2023.

[20] C. Shin, H. Lee, H. Son, S. Lee, D. Lee, and S. Lee, "Expanded adaptive scaling normalization for end to end image compression," in *ECCV*, 2022.

[21] H. Liu, T. Chen, P. Guo, Q. Shen, X. Cao, Y. Wang, and Z. Ma, "Non-local attention optimized deep image compression," *TIP*, vol. 30, no. 2, pp. 3179–3191, 2021.

[22] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *CVPR*, 2020.

[23] J. Liu, H. Sun, and J. Katto, "Learned image compression with mixed transformer-cnn architectures," in *CVPR*, 2023.

[24] L. Maaløe, M. Fraccaro, V. Liévin, and O. Winther, "BIVA: A very deep hierarchy of latent variables for generative modeling," *NeurIPS*, 2019.

[25] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder variational autoencoders," *NeurIPS*, 2016.

[26] F. Pakdaman and M. Gabbouj, "Comprehensive complexity assessment of emerging learned image compression on CPU and GPU," in *ICASSP*, 2023.

[27] R. Pérez-Gonzalo, A. Espersen, and A. Agudo, "Robust wind turbine blade segmentation from RGB images in the wild," in *ICIP*, 2023.

[28] A. Shihavuddin, X. Chen, V. Fedorov, A. Nymark Christensen, N. Andre Brogaard Riis, K. Branner, A. Bjorholm Dahl, and R. Reinhold Paulsen, "Wind turbine surface damage detection by deep learning aided drone inspection analysis," *Energies*, vol. 12, no. 4, pp. 676, 2019.

[29] S. Nandipati, A. N. Nichenametla, and A. L. Waghmare, "Cost-effective maintenance plan for multiple defect types in wind turbine blades," in *RMS*, 2018.

[30] A. González-González, A. J. Cortadi, D. Galar, and L. Ciani, "Condition monitoring of wind turbine pitch controller: A maintenance approach," *Measurement*, vol. 123, pp. 80–93, 2018.

[31] J. Duda, K. Tahboub, N. J. Gadgil, and E. J. Delp, "The use of asymmetric numeral systems as an accurate replacement for huffman coding," in *PCS*, 2015.

[32] A. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelcnn decoders," in *NeurIPS*, 2016.

[33] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in *ICIP*, 2020.

[34] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, "Checkerboard context model for efficient learned image compression," in *CVPR*, 2021.

[35] R. Child, "Very deep vaes generalize autoregressive models and can outperform them on images," in *ICLR*, 2021.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[37] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *ACSSC*, 2003.

[38] L. Prechelt, "Early stopping-but when?," in *Neural Networks: Tricks of the trade*. 1998.