Fitted Neural Lossless Image Compression

Zhe Zhang[1] Zhenzhong Chen[1*] Shan Liu[2]
[1]Wuhan University, China [2]Tencent Media Lab, USA
{zhe_zhang, zzchen}@whu.edu.cn

Abstract

Neural lossless image compression methods have recently achieved impressive compression ratios by fitting neural networks to represent data distributions of large datasets. However, these methods often require complex networks to capture intricate data distributions effectively, resulting in high decoding complexity. In this paper, we present a novel approach named Fitted Neural Lossless Image Compression (FNLIC) that enhances efficiency through a two-phase fitting process. For each image, a latent variable model is overfitted to optimize the representation of the individual image's probability distribution, which is inherently simpler than the distribution of an entire dataset and requires less complex neural networks. Additionally, we pre-fit a lightweight autoregressive model on a comprehensive dataset to learn a beneficial prior for overfitted models. To improve coordination between the pre-fitting and overfitting phases, we introduce independent fitting for the pre-fitter and the adaptive prior transformation for the overfitted model. Extensive experimental results on high-resolution datasets show that FNLIC achieves competitive compression ratios compared to both traditional and neural lossless image compression methods, with decoding complexity significantly lower than other neural methods of similar performance. The code is at https://github.com/ZZ022/FNLIC.

1. Introduction

Image compression is essential for efficient storage and transmission of images. Various sectors, such as medical imaging and remote sensing, demand strict standards for image fidelity. In these fields, lossless compression techniques are critical because they reduce file size without introducing any distortion while meeting the high demands for accuracy and quality.

Built upon Shannon's source coding theorem [31], current neural lossless compression methods compress images via fitting neural networks to represent the distributions of images from large training datasets. With the effective architecture design, they have achieved superior compression ratios compared to traditional codecs [28, 49]. However, the complexity of high-dimensional distributions in large datasets means that current methods require intricate neural network architectures to effectively represent these distributions. This reliance on complex architectures results in limited decoding efficiency and high computational demands, which limits their practical use.

Recently, an alternative approach for lossy compression based on overfitting has been introduced [8]. Rather than fitting for a dataset, these methods optimize for every image. Originally, the distorted image is represented by network parameters that mapped from coordinates to pixel values. During encoding, network parameters are overfitted to the image and encoded into the bitstream. Subsequently, the Cool-Chic series [17, 24, 25] proposes representing the distorted image via mappings from latent variables to pixel values, with latent variables encoded using autoregressive models. In this approach, both network parameters and latent variables are overfitted and encoded, significantly improving compression ratios to rival advanced standards such as VVC [6], while having significantly lower decoding complexity compared with other neural codecs.

Motivated by the success of the Cool-Chic series [17, 24, 25] in lossy compression, we introduce the overfitting-based method for lossless image compression to offer competitive compression ratios with low complexity. Instead of representing the distorted image, we represent the image's probability distribution through overfitted latent variables and network parameters. This approach can be viewed as an overfitted form of the latent variable model, a method commonly used in neural lossless compression [27, 34]. Once the image's probability distribution is established, the image can be encoded using entropy coding techniques [7, 42]. Since each neural network only needs to represent the distribution of a single image, rather than a large dataset, the overall complexity is significantly reduced, resulting in a more efficient compression process.

However, the network parameters in overfitted latent

*Corresponding author. This work was supported in part by National Natural Science Foundation of China (Grant No. 62036005) and Tencent.

variable models should also be encoded, resulting in a bitrate overhead. Consequently, a more complex network increases the bitrate required for encoding these parameters, which can degrade compression ratios. To enhance the compression ratio of the overfitted model, we propose pre-fitting a lightweight autoregressive model (ARM) to provide beneficial prior information for overfitted latent variable models, as this ARM does not add any bitrate overhead when compressing an image.

To coordinate the pre-fitted ARM with overfitted latent variable models, we introduce two simple yet effective strategies. The ARM is trained with a loss function optimized for compressing images independently, allowing it to learn a prior distribution of the image. This prior information is represented as the image's distribution parameters. During overfitting, we then adaptively transform the prior using an overfitted scale map with a small set of parameters, enabling the model to effectively leverage the prior.

Building on these innovations, we present Fitted Neural Lossless Image Compression (FNLIC), and evaluate it on several high-resolution datasets. On the Kodak dataset, FNLIC achieves a bitrate reduction of approximately 14.3% compared to a model without overfitting, while maintaining similar decoding complexity. Compared to other neural methods, FNLIC offers competitive compression ratios with significantly lower decoding complexity, boasting approximately $45\times$ faster decoding inference compared to similarly performing methods. This efficiency makes FNLIC well-suited for applications where images are encoded once but decoded multiple times.

Our contributions can be summarized as follows:

- We propose overfitting the latent variable model for lossless image compression, which simplifies the distribution the neural network needs to fit, leading to improved efficiency in representing image distributions.
- We introduce a pre-fitted autoregressive model to provide a beneficial prior for overfitted latent variable models via independent fitting and adaptive prior transformation, resulting in improved compression ratios.
- The FNLIC codec, which combines the overfitted latent variable model with the pre-fitted autoregressive model, achieves competitive compression ratios with low decoding complexity on high-resolution datasets, demonstrating its high efficiency.

2. Related Work

Neural Lossless Image Compression Several approaches for neural lossless image compression have been developed, including autoregressive models (ARMs) [30, 36], variational autoencoders (VAEs) [21, 27, 34, 35], normalizing flows [12, 13, 46, 47], and diffusion models [14, 18]. Each method offers different trade-offs between compression ratios and computational complexity. The combination of ARMs and latent variable models (or VAEs) has been shown to improve efficiency [29, 49]. FNLIC fits into this combined approach, further enhanced by a two-phase fitting scheme to improve efficiency.

Content-Adaptive Compression In the lossy compression domain, test-time optimization has been widely studied. Many of these coders use latent variable models and fine-tune models on test data, including both latent variables [44] and network parameters [38]. However, our approach differs from these methods. Our work is focused on the lossless domain, where our model includes an ARM for the image space. Additionally, we do not employ a fine-tuning stage; instead, the overfitter is trained from scratch, while the pre-fitter remains unchanged after its initial training.

Implicit Neural Representations-based Compression Compression using implicit neural representations (INRs) is introduced by COIN [8] for lossy compression. In this approach, a neural network is trained to reconstruct a lossy image, with the trained network serving as the image's representation, which is then encoded into the bitstream. This method capitalizes on the network's capacity to learn compact and efficient image representations. Expanding on this, Combiner [10] introduces Bayesian INRs, applying Bayesian principles to constrain the network's entropy. Despite these innovations, the performance of INR-based models remains relatively weak compared to other state-of-the-art techniques.

Low Complexity Neural Lossless Image Codecs To mitigate the issue of large computational complexity and enhance the practicality of neural image codecs, several methods have been proposed. PILC [16] employs a lightweight ARM in combination with VQ-VAE [37] to achieve compression speeds comparable to PNG [5] on high-performance GPUs. LLICTI [15] explores network and color space designs to enhance compression efficiency. FSAR [48] introduces finite-state entropy coding to improve the efficiency of autoregressive models. Despite these improvements, a considerable compression ratio gap remains when compared to the most advanced traditional codecs, such as JPEG-XL [2].

3. Method

We now present Fitted Neural Lossless Image Compression (FNLIC). FNLIC overfits a latent variable model for each image, enabling effective representation of image distributions through lightweight neural networks. FNLIC additionally employs a pre-fitted lightweight autoregressive model that provides beneficial prior to enhance the compression ratios. To coordinate the overfitting process with the pre-fitted model, FNLIC utilizes independent fitting and adaptive prior transformation strategies. For clarity, Figure 1 illustrates the decoding process, while Algorithm 1 details the complete encoding procedure.

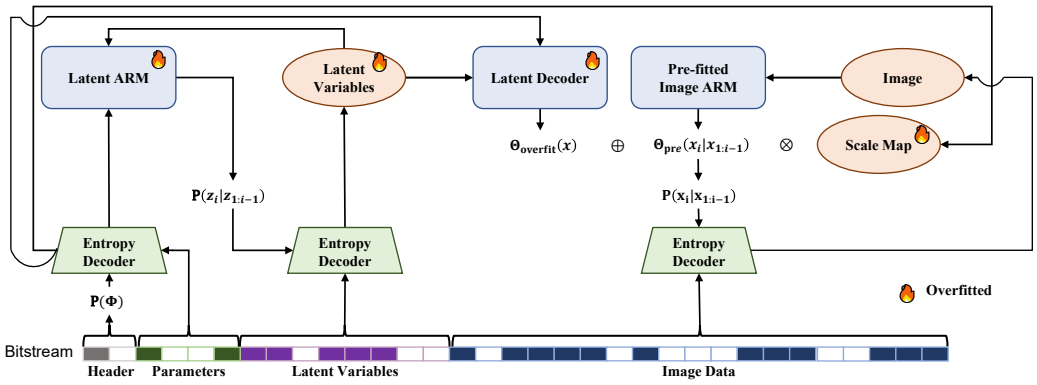

Figure 1. The decoding process of FNLIC. First, the network parameter probability is retrieved from the header. Then, the scale map and network parameters, including the latent ARM and latent decoder, are decoded. Using the latent ARM, the latent variables z are decoded. Next, the overfit-fitted distribution parameters Θ_{overfit} for the original image are determined by the latent decoder with z as input. Finally, the original image x is decoded using Θ_{overfit}, the scale map, and a pre-fitted image ARM.

3.1. Background

Neural Lossless Image Compression Lossless compression relies on entropy coders. Given a symbol s and its probability $p(s)$, advanced entropy coders, such as Arithmetic Coding [42] and Asymmetric Numeral Systems (ANS) [7], can approximately represent the symbol using $-\log_2 p(s)$ bits (we omit the base 2 in the following sections for simplicity).

Current neural lossless image methods use neural networks to model the distribution of large datasets. A neural codec stores a data distribution $p(x)$ through network parameters, which are shared across all images and incur no additional bitrate overhead. According to Shannon's source coding theorem [31], for a set of images following the distribution $p_{\text{real}}(x)$, the expected average code length is the cross-entropy $\mathbb{E}p_{\text{real}}[-\log p(x)]$.

Entropy coders typically operate at the pixel level. The output from a neural codec is a distribution parameter map $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ (n is the number of the pixel of the image), which may include parameters like the mean and scale for a Gaussian distribution. The bitrate for encoding an image x can be expressed as follows:

$$C = \sum_{i=1}^{n} -\log p(x_i; \theta_i). \tag{1}$$

The parameter map has the same dimensions as the original image, and each pixel is encoded based on the probability determined by the corresponding parameter.

Several distribution modeling methods have been proposed. In this paper, we use autoregressive models [36] and latent variable models [27], which we introduce below.

Autoregressive Model Autoregressive models (ARM) predict the probability of x as the product of conditional probabilities. The image is divided into several components $\{x_1, x_2, \ldots, x_n\}$, with decoding performed component by component, such as pixel by pixel [36]. The distribution parameter for each component is derived using an ARM, Φ_{arm}, based on the preceding components, where

$$\theta_i = \Phi_{\text{arm}}(x_{1:i-1}). \tag{2}$$

Latent Variable Model In addition to the input image x, latent variable models utilize latent variables z to provide auxiliary information for image compression. These models are also referred to as variational autoencoders [20]. Latent variables can be either stochastic or deterministic. Stochastic latent variables are sampled from the posterior distribution. While the former approach can achieve higher compression ratios, it requires the bits-back coding method [34], which involves inferring both the latent encoder and decoder during decoding, and requires initial bits that reduce performance when encoding a single image [29].

In contrast, deterministic latent variables do not require sampling and only necessitate the inference of the latent decoder during decoding, making them more suitable for low-complexity methods [16, 48]. To enhance efficiency, we use deterministic latent variables in FNLIC, where the distribution parameters of the image are determined by a latent decoder Φ_{dec} using z as input, where

$$\Theta_x = \Phi_{\text{dec}}(z). \tag{3}$$

3.2. Overfitted Latent Variable Model

Current neural image compression methods have achieved outstanding performance in terms of compression ratios. However, representing the high-dimensional distribution of a large dataset is complex, requiring complex neural networks. As a result, neural codecs often require high decoding complexity to achieve high compression ratios.

Algorithm 1 Encoding Process

procedure OVERFITTING
  Input: $\Phi_{\text{arm},x}$, x,Training Steps N
  Output: z, $\Phi_{\text{arm},z}$, Φ_{dec}, a
  $\Theta_{\text{pre}} \leftarrow \Phi_{\text{arm},x}(x)$ using Equation 2
  Initialize z, $\Phi_{\text{arm},z}$, Φ_{dec}, a
  for $i \leftarrow 1$ to N **do**
    Compute l using Equation 7
    Backward propagation from l
    Update z, $\Phi_{\text{arm},z}$, Φ_{dec}, a
  end for
  Greedy quantize $\Phi_{\text{arm},z}$, Φ_{dec}
  Finetune z
  return z, $\Phi_{\text{arm},z}$, Φ_{dec}, a
end procedure

procedure ENCODING
  Input: $\Phi_{\text{arm},x}$, x, z, $\Phi_{\text{arm},z}$, Φ_{dec}, a
  Output: Bitstream B
  Initialize empty B
  Write header to B
  Encode $\Phi_{\text{arm},z}$, Φ_{dec}, a to B
  $\Theta_z \leftarrow \Phi_{\text{arm},z}(z)$ using Equation 2
  Encode z to B using Θ_z
  $\Theta_{\text{pre}} \leftarrow \Phi_{\text{arm},x}(x)$ using Equation 2
  $\Theta_{\text{overfit}} \leftarrow \Phi_{\text{dec}}(z)$
  $\Theta_x \leftarrow a \times \Theta_{\text{pre}} + \Theta_{\text{overfit}}$
  Encode x to B using Θ_x
  return B
end procedure

To address the issue, we propose using an overfitted latent variable model for lossless image compression, motivated by its success in lossy compression. We draw inspiration from the design of C3 [17], a high-performance overfitting-based lossy codec. We call this overfitted latent variable model the overfitter.

In our method, the probability is represented at the pixel level, including red, green, and blue sub-pixels. The probability of each sub-pixel is modeled by a Logistic distribution parameterized by the mean μ and scale σ. We adopt the weak channel-wise autoregression from [30] to model dependencies in the RGB space as follows:

$$\begin{aligned} \mu_r &= f_r \\ \mu_g &= f_g + \alpha \cdot x_r \\ \mu_b &= f_b + \beta \cdot x_r + \gamma \cdot x_g. \end{aligned} \tag{4}$$

Thus, the probability of the image is represented by a $9 \times H \times W$ parameter map Θ, where each element $\theta = \{f_r, f_g, f_b, \alpha, \beta, \gamma, \sigma_r, \sigma_g, \sigma_b\}$.

The overfitter represents the overfitted parameters Θ_{overfit} via multi-resolution latent variables z and a latent decoder Φ_{dec} using Equation 3. The latent decoder includes a learned upsampler, which upsamples latent variables at different resolutions to match the resolution of the original image, and a synthesizer, which transforms the upsampled latent variables into Θ_{overfit}. Furthermore, a pixel-by-pixel ARM $\Phi_{\text{arm},z}$ is employed to establish the distribution of these latent variables. To encode an image, latent variables and network parameters should be overfitted to the image and encoded into the bitstream.

To reduce the complexity of the pixel-by-pixel ARM, we adopt Finite-State Autoregressive (FSAR) entropy coding as proposed by [48]. Specifically, we use a second-order autoregressive Markov model, where the distribution of the current pixel is determined by the two previously decoded pixels, as shown in Figure 2. Consequently, the number of possible inputs for the latent ARM is L^2, where L is the size of the possible values for a latent variable. Therefore, FNLIC only needs to infer the latent ARM for an L^2 input in parallel, independent of the image resolution. The results are then used to build a tabled ANS table for fast autoregression and entropy coding for the latent variable.

Compared to latent variable models used in dataset-fitted approaches, overfitted models offer distinct advantages. First, while dataset-fitted latent variables are derived from the input image, overfitted variables are free to assume any value, resulting in more effective latent variables. Second, unlike dataset-fitted models, which are not tailored to the test image, overfitted variables are specifically optimized for each image, which eliminates the domain gap issue. Finally, representing the probability of a single image is much simpler than modeling the distribution of a large dataset, resulting in significantly lower complexity.

3.3. Pre-fitted Autoregressive Model

Although overfitting offers several advantages, its compression ratios are constrained by the need to encode network parameters. Increasing the number of parameters to achieve a more precise representation results in a higher bitrate for these parameters. As a result, more complex neural networks may even degrade overall performance.

To further enhance the compression ratios of the overfitters, we leverage dataset-fitted neural lossless compression models. As discussed in Section 3.1, dataset-fitted methods achieve strong compression ratios by representing distributions of large datasets. Therefore, a dataset-fitted model can provide prior information without any overhead from net-

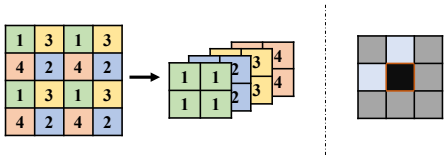

Figure 2. The spatial autoregression in FNLIC. **Left**: Spatial autoregression for the image. **Right**: Spatial autoregression for latent variables.

work parameters, which can assist in the probability representations within the overfitters.

Therefore, we pre-fit an autoregressive model on a comprehensive dataset to provide a beneficial prior. Specifically, we use the probability distribution represented by a parameter map, Θ_{pre}, as the prior, which can be viewed as a prior distribution for each image. We refer to this dataset-fitted model as the pre-fitter. Since our goal is to propose an efficient codec, we employ a lightweight ARM with fast decoding speed. For this, we use the popular four-step checkerboard autoregression [11], paired with a small network architecture, as shown in Figure 2. The image is split into four sub-images, with the distribution parameters for each current sub-image produced by a neural network, $\Phi_{\text{arm},x}$, as described in Equation 2. To model the first sub-image's distribution, we also learn a set of distribution parameters shared by all pixels within this sub-image.

3.4. Coordinating the Pre-fitter and the Overfitter

With the introduction of the pre-fitter, a key challenge is how to coordinate the pre-fitter and overfitter, specifically addressing: 1) how to fit the pre-fitter to learn a beneficial prior, and 2) how the overfitter should use the prior distribution parameters Θ_{pre}. Surrogate models are frequently used in such scenarios. For example, in pre-processing for traditional codecs, surrogate codecs are often employed to simulate traditional codecs and provide gradients [9]. Applied to our method, a straightforward approach would be to use a surrogate latent variable model to fit the pre-fitter. However, this approach is suboptimal. The issue stems from the mismatch between dataset-fitted and overfitted latent variable models [44]. The latent variables in the pre-fitting stage differ significantly from those in the overfitting stage, making the prior parameters unsuitable for use in this way. As shown in Table 3, this strategy can occasionally result in a performance drop.

Independent Fitting To address the first problem, we propose an effective fitting method for the pre-fitter. Our goal is to accurately represent the image's probability using the probability parameter Θ. If the pre-fitted probability parameter is accurate on its own, as reflected by a low bitrate (Equation 1), it can assist the overfitter. Therefore, we independently fit a convolutional ARM $\Phi_{\text{arm},x}$ to learn generalizable priors [45] using the loss of encoding the image solely with Θ_{pre}, where

$$\Phi_{\text{arm},x} = \arg\min_{\Phi_{\text{arm},x}} \mathbb{E}p_{\text{real}}[-\log p(x;\Theta_{\text{pre}})]. \tag{5}$$

Adaptive Prior Transformation To address the second problem, we propose a strategy for the overfitters to obtain the distribution parameter Θ, where

$$\Theta = a \times \Theta_{\text{pre}} + \Theta_{\text{overfit}}. \tag{6}$$

We adaptively transform the prior Θ_{pre} by introducing a scale map a. The scale map a is fitted during overfitting. We consider two special cases: 1) $a = 1$, where our method is similar to predictive coding in traditional codecs. The pre-fitter provides a good initial performance, while the overfitter represents the residual of the distribution parameters. 2) $a = 0$. Since the pre-fitter is trained without awareness of the overfitter, it may reduce the effectiveness of the overfitter for certain images and degrade performance. Setting $a = 0$ can address this issue by eliminating the influence of the pre-fitter when it is not beneficial, thereby improving performance. In other cases, the scale map can adaptively adjust the prior probability to better suit the overfitter. The scale map is represented by 36 parameters—9 for each of the 9 distribution components of Θ—and is applied separately to each of the four sub-images, considering that each sub-image has a different amount of context.

This simple form has several advantages. First, it is lightweight, with negligible computational and bitrate costs. Second, it is easy to optimize. We experimented with more complex methods, but these resulted in degraded performance, likely due to optimization difficulties [43].

Loss Function The loss function is formulated as follows:

$$l = -\log p(x;\Theta_x) - \log p(z;\Theta_z). \tag{7}$$

4. Experiments

4.1. Experimental Setup

Training Settings Following L3C [27], we train our pre-fitter on a subset of the Open Images dataset [23], which includes approximately 360K images. The images are selected and preprocessed in the same way as in L3C. During training, we randomly crop images to a size of 128×128 pixels per step, with a batch size of 96. The initial learning rate is set to 0.0005, and we train for 30 epochs, decaying the learning rate by 0.75 every five epochs. We use the Adam optimizer [19]. It takes approximately two days to use two 1080Ti GPUs for training. Details on the architecture of the pre-fitter and overfitter are provided in the supplementary material.

Table 1. Compression ratios in BPD for three datasets and decoding complexity in terms of inference time, where lower values indicate better performance. The best performance is highlighted in bold. Dec. Inf T. denotes decoding inference time. The first group includes traditional codecs, and the second group includes neural codecs. The inference time is measured on the Kodak dataset where the image resolution is 768×512.

Method	Compression Rate (Bits Per Dimension)						Dec. Inf. T. (ms)	
	Kodak	CLIC2024	DIV2K	Histo24	LoveDA24	Doc24	GPU	CPU
PNG [5]	4.50	4.15	4.27	4.58	4.49	4.75	-	-
JPEG-LS [41]	4.36	3.89	3.99	4.24	4.18	4.72	-	-
JPEG2000 [32]	3.30	3.14	3.13	4.43	3.65	2.53	-	-
WebP [40]	3.18	3.02	3.11	4.21	3.48	2.88	-	-
FLIF [33]	2.90	2.83	2.91	4.05	3.50	2.54	-	-
JPEG-XL [2]	2.87	2.76	2.79	4.08	3.32	2.30	-	-
L3C [27]	3.25	3.11	3.09	4.53	3.69	3.26	45	1176
PILC [16]	-	-	3.41	-	-	-	-	-
LLICTI [15]	3.01	2.82	2.79	4.22	3.49	2.87	49	1325
LC-FDNet [28]	2.93	2.69	2.71	4.12	3.34	2.72	607	14903
DLPR [3]	2.86	2.51	**2.55**	4.12	3.27	2.68	1613	42212
ArIB-BPS [49]	**2.78**	**2.48**	**2.55**	3.95	**3.23**	2.62	13374	344488
FNLIC (ours)	2.88	2.68	2.75	**3.86**	3.30	**2.22**	**13**	**221**

Test Datasets We evaluate our method on six high-resolution datasets: Kodak [22], CLIC2024 validation[1], DIV2K validation [1], ANHIR [4], LoveDA [39], and the Rumsey Validation Data for ICDAR'24 MapText Competition[2]. The first three datasets consist of natural images sharing similar domains with our training data: Kodak contains 24 images at 768×512 resolution, while CLIC2024 validation includes 30 images and DIV2K validation provides 100 images, both with approximately 2K resolution.

The latter three datasets represent out-of-domain (OOD) scenarios: ANHIR features medical histology images, LoveDA comprises remote sensing imagery, and the Rumsey Validation Data contains historical document images from the ICDAR'24 MapText Competition. Following DLPR [3], we generate evaluation subsets by randomly sampling 24 non-overlapping 768×512 patches from each OOD dataset, constructing three datasets (Histo24, LoveDA24, and Doc24).

4.2. Compression Performance

We compare FNLIC with both traditional and advanced neural compression methods. For FNLIC's compression ratios, we measure the encoded file size including header information. Compression ratios for compared methods are obtained from original publications when available, otherwise evaluated through open-source implementations. All measurements are reported in Bits Per Dimension (BPD).

To assess decoding complexity, we employ inference time as our metric. Evaluations are conducted on two hardware platforms: an NVIDIA GeForce 1080Ti GPU and an Intel Xeon E5-2637 v4 @ 3.50GHz CPU. Inference time measurements are performed on the Kodak dataset through five repeated trials, with results presented in Table 1.

The results demonstrate that FNLIC achieves competitive compression ratios. It outperforms JPEG-XL, the state-of-the-art traditional method, on the CLIC2024, DIV2K, and all OOD datasets, while showing slightly inferior performance on the Kodak dataset. Furthermore, FNLIC maintains comparable compression performance to advanced neural methods.

FNLIC's primary advantage lies in decoding efficiency. Compared to L3C and LLICTI, it achieves superior compression ratios with significantly faster inference speeds - over 3× faster on GPU and more than 5× faster on CPU. While matching LC-FDNet's compression performance, FNLIC requires substantially lower decoding complexity, demonstrating more than 45× speed improvement on GPU and over 60× improvement on CPU. Although DLPR and ArIB-BPS achieve better compression ratios, they exhibit decoding times exceeding two orders of magnitude longer than FNLIC. These findings confirm FNLIC's efficiency in both compression performance and decoding speed.

Additionally, FNLIC demonstrates superior generalization capability. Through test-specific overfitting, it effectively handles diverse domains, whereas methods relying solely on pre-trained models show performance degradation on OOD images. For instance, LC-FDNet - which surpasses JPEG-XL on 2 out of 3 in-domain datasets - fails to outperform JPEG-XL on any OOD dataset, whereas FNLIC maintains its advantage over the traditional method.

Encoding Complexity Similar to other overfitting-based methods, FNLIC incurs substantial computational overhead

[1] https://www.compression.cc/
[2] https://zenodo.org/records/11516933

Table 2. Ablation for the effectiveness of overfitting and pre-fitting on the Kodak dataset.

Overfit	Pre-fit	BPD	GPU (ms)	CPU (ms)
✓	✓	2.88	13±0.2	221±20.0
✓		2.93	8±0.1	178±17.8
	✓	3.36	12±0.1	209±18.6

during encoding. Processing a 768×512 image requires approximately 44 seconds per 1,000 optimization iterations, with a complete encoding process typically requiring about 140K iterations. A practical strategy for acceleration involves reducing the number of training iterations. As demonstrated in the supplementary material, FNLIC maintains comparable performance with substantially fewer optimization steps. Detailed analysis of encoding complexity considerations is provided in the supplementary material.

4.3. Ablation Studies

We conduct three ablation studies on the Kodak dataset to demonstrate the effectiveness of the overfitting, pre-fitting components, and coordinating strategy in FNLIC.

4.3.1 Effectiveness of Overfitting

First, we evaluate the impact of overfitting by using a model without overfitting. The architectures of the image ARM and latent decoder are identical to those in FNLIC. To obtain latent variables, we employ a latent encoder, similar to those in popular latent variable models, which extracts latent variables by transforming the original image. To ensure sufficient capacity, the latent encoder is significantly more complex than the latent decoder. In this setup, the ARM and latent variable models are coordinated as $\Theta = \Theta_{\text{pre}} + \Theta_{\text{overfit}}$. We also visualize the upsampled latent variables generated by the upsampler in the latent variable models, as well as the image bitrate and bitrates for each latent variable resolution, in Figure 3.

As shown in Table 2, the overfitting stage provides a significant compression ratio gain, with a 0.48 BPD improvement, albeit with a slight increase in inference time. The difference in inference time between using and not using overfitting lies in the handling of the latent ARM. FSAR utilizes a latent ARM that pre-calculates all possible inputs. For FNLIC, however, the latent ARM inference must be performed for each image, as it varies across images. For dataset-trained latent ARMs, pre-calculation and storage are possible, eliminating the need for its inference.

The performance gain can be attributed to the optimization of the latent variable models. As shown in the first and second rows of Figure 3, overfitting enables the latent variables to learn more effective representations. This effectiveness can be understood from two perspectives. First, overfitting reduces several gaps between dataset-fitted latent variables and the optimal latent variables, as discussed in [44]. Second, in the combination of ARM and latent variable models, posterior collapse [26] may occur, where non-useful latent variables are learned. This effect is visible in the first and third levels of Figure 3 for the dataset-fitted model, while it is absent in the overfitted model.

4.3.2 Effectiveness of Pre-fitting

Additionally, we assess the influence of the pre-fitted ARM by removing it and using only the overfitted latent variable model to encode the image.

As illustrated in Table 2, the introduction of the pre-fitter results in approximately a 0.05 BPD improvement in compression ratio, with an increase in inference time of 63% on a GPU and 24% on a CPU. Given that low-complexity methods are primarily suited for weaker devices, such as CPUs, and that inference time on GPUs is already fast, this trade-off is acceptable. Furthermore, despite the increased complexity, FNLIC still maintains very low decoding complexity compared to existing methods.

As discussed in Section 3.3, the compression ratio of the pure overfit-fitter is constrained by the bitrate of parameters, which is effectively improved by introducing the pre-fitter. Figure 3 further demonstrates the effectiveness of the pre-fitter: the pre-fitted ARM reduces the bitrate of latent variables, particularly at the first level, with a relatively minor increase in the bitrate of the original image, thereby reducing the overall bitrate. The first-level latent variables capture local data modalities, as they share the same resolution as the original image. In previous dataset-fitted methods combining ARM and latent variable models, ARMs are used to learn these local data modalities [29]. Thus, the pre-fitter provides this information, significantly reducing the bitrate of related latent variables.

4.3.3 Effectiveness of the Coordination Strategy

Finally, we validate the effectiveness of our two proposed coordination strategies, as shown in Table 3, by examining: 1) whether to fit the pre-fitter independently, and 2) whether to adaptively transform the prior using an optimized scale map, with a fixed to 1 for comparison.

As illustrated in Table 3, our coordination strategy demonstrates the highest effectiveness. Specifically, as shown in row 2, the straightforward approach of using a surrogate latent variable model to fit the pre-fitter and fusing them with the same method (i.e., $\Theta = \Theta_{\text{pre}} + \Theta_{\text{overfit}}$) does not lead to improved performance when compared to using the overfitter alone. Furthermore, only 10 out of 24 images display enhanced performance, suggesting that the mismatch between the dataset-fitted and overfitted latent

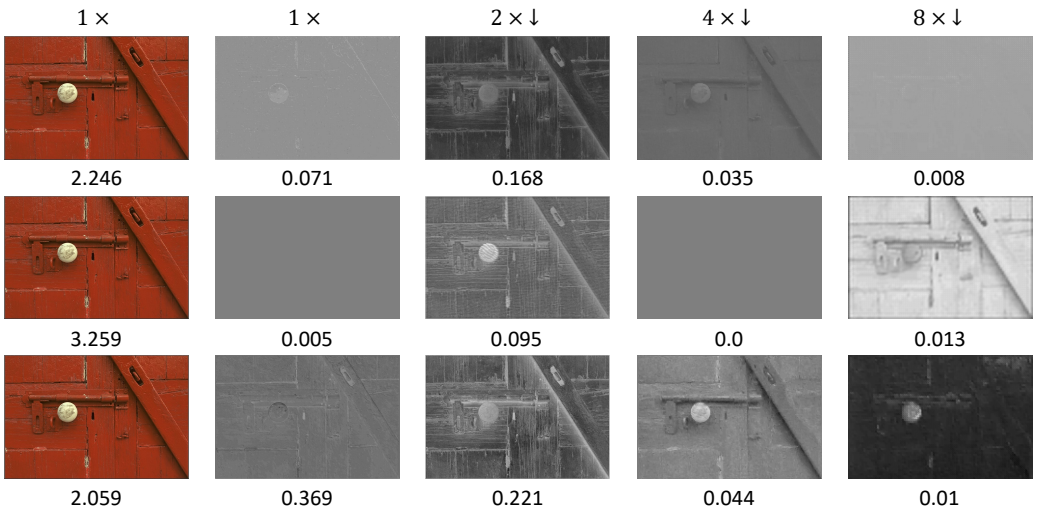

Figure 3. Visualization of the learned latent variables with upsampling and their bit rates. The first row demonstrates the resolution. For better visualization, the value are normalized to [0,1]. In the fisrt column, we also list the bitrate for representing the original image. Top: FNLIC. Mid: FNLIC Without Overfitting. Bottom: FNLIC Without Pre-Fitting.

Table 3. Ablation study on the effectiveness of the coordination strategy. 'Independent' indicates that the surrogate overfitter is not used during training. 'Transformation' denotes the use of the scale map to transform the prior. 'Improve Num' indicates the number of images (out of 24 test images) that show improved performance compared to using the overfitter alone.

Independent	Transformation	BPD	Improve Num
✓	✓	2.88	22
		2.93	10
✓		2.91	16
	✓	2.89	20

variable models has a negative impact on the performance of many images.

However, as shown in row 4, introducing the scale map to adaptively utilize the prior from the pre-fitter achieves better compression ratios, highlighting the effectiveness of the scale map. Further evidence is provided in rows 1 and 3. Notably, rows 1 and 4 also show that a pre-fitter without latent variable awareness demonstrates superior performance, suggesting that it learns more beneficial prior information.

The effectiveness of the adaptive prior transformation is further validated by the number of images with improved compression ratios. The adaptive prior transformation can adjust the prior from being potentially harmful to beneficial, thereby enhancing overall performance. The number of images with improved compression ratios increases from 16 to 22 when using the independently fitted pre-fitter, and from 10 to 20 with the alternative approach.

5. Conclusions and Limitations

Conclusions We have proposed Fitted Neural Lossless Image Compression (FNLIC) for efficient lossless compression. FNLIC employs an autoregressive model pre-fitter, trained on a large dataset to learn generalizable priors, which are utilized for every test image. During encoding, a latent variable model is overfitted for each image, enabling adaptive optimization of the latent variables. We have also proposed independent fitting and adaptive prior transformation for the coordination of the pre-fitter and the overfitter. On several high-resolution image datasets, FNLIC demonstrates competitive compression ratios with significantly lower decoding complexity compared to other neural lossless image compression methods. FNLIC also demonstrates high generalization ability.

Limitations FNLIC has high encoding complexity, making it unsuitable for scenarios requiring fast encoding, such as real-time communications. This high complexity is primarily due to the overfitting process required for each image. A potential solution is the use of more efficient optimization methods that can reduce the time required for encoding without compromising compression performance. Additionally, the bit rate of the network parameters is not optimized during the encoding process. This becomes a significant drawback when dealing with images of small resolution, as the network's bit rate can occupy a substantial portion of the total bit budget, thereby diminishing overall performance. To extend FNLIC for low-resolution images, efficient strategies for network bitrate optimization should be explored in the future work.

References

[1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1122–1131, 2017. 6

[2] Jyrki Alakuijala, Ruud Van Asseldonk, Sami Boukortt, Martin Bruse, Iulia-Maria Comșa, Moritz Firsching, Thomas Fischbacher, Evgenii Kliuchnikov, Sebastian Gomez, Robert Obryk, et al. JPEG XL next-generation image compression architecture and coding tools. In *Applications of Digital Image Processing XLII*, pages 112–124. SPIE, 2019. 2, 6

[3] Yuanchao Bai, Xianming Liu, Kai Wang, Xiangyang Ji, Xiaolin Wu, and Wen Gao. Deep lossy plus residual coding for lossless and near-lossless image compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5): 3577–3594, 2024. 6

[4] Jiří Borovec, Jan Kybic, Ignacio Arganda-Carreras, Dmitry V. Sorokin, Gloria Bueno, Alexander V. Khvostikov, Spyridon Bakas, Eric I-Chao Chang, Stefan Heldmann, Kimmo Kartasalo, Leena Latonen, Johannes Lotz, Michelle Noga, Sarthak Pati, Kumaradevan Punithakumar, Pekka Ruusuvuori, Andrzej Skalski, Nazanin Tahmasebi, Masi Valkonen, Ludovic Venet, Yizhe Wang, Nick Weiss, Marek Wodzinski, Yu Xiang, Yan Xu, Yan Yan, Paul Yushkevich, Shengyu Zhao, and Arrate Muñoz-Barrutia. ANHIR: Automatic non-rigid histological image registration challenge. *IEEE Transactions on Medical Imaging*, 39(10):3042–3052, 2020. 6

[5] Thomas Boutell. Png (portable network graphics) specification version 1.0. Technical report, 1997. 2, 6

[6] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 1

[7] Jarek Duda. Asymmetric numeral systems. *arXiv preprint arXiv:0902.0271*, 2009. 1, 3

[8] Emilien Dupont, Adam Golinski, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. COIN: Compression with implicit neural representations. In *Neural Compression: From Information Theory to Applications – Workshop @ ICLR 2021*, 2021. 1, 2

[9] Mengxi Guo, Shijie Zhao, Hao Jiang, Junlin Li, and Li Zhang. Video compression with arbitrary rescaling network. In *2023 Data Compression Conference*, 2023. 5

[10] Zongyu Guo, Gergely Flamich, Jiajun He, Zhibo Chen, and José Miguel Hernández-Lobato. Compression with bayesian implicit neural representations. *Advances in Neural Information Processing Systems*, 36:1938–1956, 2023. 2

[11] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14771–14780, 2021. 5

[12] Jonathan Ho, Evan Lohn, and Pieter Abbeel. Compression with flows via local bits-back coding. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[13] Emiel Hoogeboom, Jorn Peters, Rianne van den Berg, and Max Welling. Integer discrete flows and lossless compression. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[14] Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022. 2

[15] Fatih Kamisli. Learned lossless image compression through interpolation with low complexity. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(12):7832–7841, 2023. 2, 6

[16] Ning Kang, Shanzhao Qiu, Shifeng Zhang, Zhenguo Li, and Shu-Tao Xia. PILC: Practical image lossless compression with an end-to-end GPU oriented neural framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3739–3748, 2022. 2, 3, 6

[17] Hyunjik Kim, Matthias Bauer, Lucas Theis, Jonathan Richard Schwarz, and Emilien Dupont. C3: High-performance and low-complexity neural compression from a single image or video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9347–9358, 2024. 1, 4

[18] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in Neural Information Processing Systems*, 34:21696–21707, 2021. 2

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[21] Friso Kingma, Pieter Abbeel, and Jonathan Ho. Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables. In *International Conference on Machine Learning*, pages 3408–3417, 2019. 2

[22] Eastman Kodak. Kodak lossless true color image suite (photocd pcd0992), 1993. 6

[23] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. OpenImages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017. 5

[24] Théo Ladune, Pierrick Philippe, Félix Henry, Gordon Clare, and Thomas Leguay. COOL-CHIC: Coordinate-based low complexity hierarchical image codec. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13515–13522, 2023. 1

[25] Thomas Leguay, Théo Ladune, Pierrick Philippe, Gordon Clare, Félix Henry, and Olivier Déforges. Low-complexity overfitted neural image codec. In *IEEE 25th International Workshop on Multimedia Signal Processing*, pages 1–6. IEEE, 2023. 1

[26] James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don't blame the ELBO! a linear VAE perspec-

tive on posterior collapse. *Advances in Neural Information Processing Systems*, 32, 2019. 7

[27] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Practical full resolution learned lossless image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10629–10638, 2019. 1, 2, 3, 5, 6

[28] Hochang Rhee, Yeong Il Jang, Seyun Kim, and Nam Ik Cho. LC-FDNet: Learned lossless image compression with frequency decomposition network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6033–6042, 2022. 1, 6

[29] Tom Ryder, Chen Zhang, Ning Kang, and Shifeng Zhang. Split hierarchical variational compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 386–395, 2022. 2, 3, 7

[30] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017. 2, 4

[31] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. 1, 3

[32] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, 2001. 6

[33] Jon Sneyers and Pieter Wuille. FLIF: Free lossless image format based on maniac compression. In *IEEE International Conference on Image Processing*, pages 66–70. IEEE, 2016. 6

[34] James Townsend, Thomas Bird, and David Barber. Practical lossless compression with latent variables using bits back coding. In *International Conference on Learning Representations*, 2019. 1, 2, 3

[35] James Townsend, Thomas Bird, Julius Kunze, and David Barber. HiLLoC: lossless image compression with hierarchical latent variable models. In *International Conference on Learning Representations*, 2020. 2

[36] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1747–1756, 2016. 2, 3

[37] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[38] Ties van Rozendaal, Iris AM Huijben, and Taco Cohen. Overfitting for fun and profit: Instance-adaptive data compression. In *International Conference on Learning Representations*, 2021. 2

[39] Junjue Wang, Zhuo Zheng, Ailong Ma, Xiaoyan Lu, and Yanfei Zhong. LoveDA: A remote sensing land-cover dataset for domain adaptive semantic segmentation. *arXiv preprint arXiv:2110.08733*, 2021. 6

[40] WebP Image format. https://developers.google.com/speed/webp/. 6

[41] M.J. Weinberger, G. Seroussi, and G. Sapiro. The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. *IEEE Transactions on Image Processing*, 9(8):1309–1324, 2000. 6

[42] Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Commun. ACM*, 30(6): 520–540, 1987. 1, 3

[43] Tongda Xu, Han Gao, Chenjian Gao, Yuanyuan Wang, Dailan He, Jinyong Pi, Jixiang Luo, Ziyu Zhu, Mao Ye, Hongwei Qin, Yan Wang, Jingjing Liu, and Ya-Qin Zhang. Bit allocation using optimization. In *Proceedings of the 40th International Conference on Machine Learning*, pages 38377–38399, 2023. 5

[44] Yibo Yang, Robert Bamler, and Stephan Mandt. Improving inference for neural image compression. *Advances in Neural Information Processing Systems*, 33:573–584, 2020. 2, 5, 7

[45] Mingtian Zhang, Andi Zhang, and Steven McDonagh. On the out-of-distribution generalization of probabilistic image modelling. *Advances in Neural Information Processing Systems*, 34:3811–3823, 2021. 5

[46] Shifeng Zhang, Ning Kang, Tom Ryder, and Zhenguo Li. iFlow: Numerically invertible flows for efficient lossless compression via a uniform coder. *Advances in Neural Information Processing Systems*, 34:5822–5833, 2021. 2

[47] Shifeng Zhang, Chen Zhang, Ning Kang, and Zhenguo Li. iVPF: Numerical invertible volume preserving flow for efficient lossless compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 620–629, 2021. 2

[48] Yufeng Zhang, Hang Yu, Jianguo Li, and Weiyao Lin. Finite-state autoregressive entropy coding for efficient learned lossless compression. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3, 4

[49] Zhe Zhang, Huairui Wang, Zhenzhong Chen, and Shan Liu. Learned lossless image compression based on bit plane slicing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27579–27588, 2024. 1, 2, 6