

WISE: A Framework for Gigapixel Whole-Slide-Image Lossless Compression

Yu Mao¹, Jun Wang², Nan Guan², Chun Jason Xue¹

¹MBZUAI, ²City University of Hong Kong

{yu.mao, jason.xue}@mbzuai.ac.ae {jwang699-c, nanguan}@cityu.edu.hk

Abstract

Whole-Slide Images (WSIs) have revolutionized medical analysis by presenting high-resolution images of the whole tissue slide. Despite avoiding the physical storage of the slides, WSIs require considerable data volume, which makes the storage and maintenance of WSI records costly and unsustainable. To this end, this work presents the first investigation of lossless compression of WSI images. Interestingly, we find that most existing compression methods fail to compress the WSI images effectively. Furthermore, our analysis reveals that the failure of existing compressors is mainly due to information irregularity in WSI images. To resolve this issue, we develop a simple yet effective lossless compressor called WISE, specifically designed for WSI images. WISE employs a hierarchical encoding strategy to extract effective bits, reducing the entropy of the image and then adopting a dictionary-based method to handle the irregular frequency patterns. Through extensive experiments, we show that WISE can effectively compress the gigapixel WSI images to 36 times on average and up to 136 times.

1. Introduction

The ever-increasing utilization of advanced medical imaging techniques, such as Whole-Slide Images (WSI) [18, 30], has led to a surge in the generation of gigapixel medical image data. These detailed volumetric images are indispensable for accurate diagnosis and effective treatment planning. However, their considerable size presents notable challenges for storage and transmission. For instance, a single digital pathology WSI scan can produce a four-dimensional (4D) image of a patient’s tissue sample comprising width, height, color channels, and multiple resolution levels, resulting in a total data volume reaching several gigabytes [15]. As shown in Fig.1, two well-known datasets, C16 [6] and C17 [20], consist primarily of whole slide images (WSIs) ranging in size from 1 to 5GB. The large size of these images presents significant challenges for storage and transmission. Considering a person got a WSI image as a teenager, the WSI records need to be stored

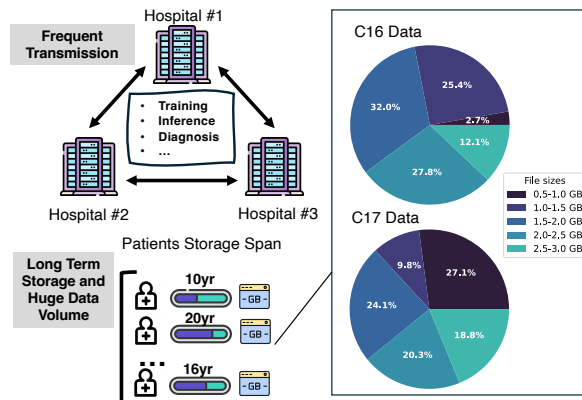


Figure 1. WSIs are extremely large images (~ 3 GB/slide uncompressed) of high resolution (0.25 microns per pixel), presenting a significant data storage challenge for hospitals wishing to adopt digital pathology. In practice, hospitals often ship physical drives via FedEx to transfer WSIs. Therefore, there is a desperate need for better compression of WSIs.

for decades to keep track. Those cold, huge data can cost a lot for storage. Also, as the cloud-based medical ecosystem expands, the frequency of WSI transfers between data centers is increasing due to the growing demands for training, inference, and diagnosis [20, 21, 31].

Given the aforementioned challenges in using WSI images for digital pathology, there exists a clear need for efficient and lossless WSI compressors. However, most existing WSI compression approaches focus on lossy compression, including those classical methods that leverage lossy compressors such as JPEG-2000, as well as those modern compressors that utilize DNN to encode WSI patches into embeddings [5, 11, 12, 15]. While lossy compression techniques can achieve high compression ratios, they risk introducing distortions that could compromise the diagnostic integrity of the WSI images and potentially lead to medical issues. On the other hand, lossless compression techniques can avoid distorting the original WSI information. However, our preliminary experiments with existing lossless compression techniques show that even state-of-the-art lossless compressors can not yield the desired compression

performance. Traditional lossless image compressors like PNG, entropy encoders like Huffman, or dictionary compressors like Gzip, demonstrate limited effectiveness on WSI images. Even modern Neural Network (NN)-based compressors can achieve only limited compression, around 1.6 to 2 times for the non-empty area in WSI images, which is far from satisfactory given that a single WSI image contains tens of gigabytes.

Therefore, in this work, we fill this gap by developing an efficient lossless WSI compressor. First, we observe that most WSI images consist of empty areas that can be efficiently encoded as simple coordinate tuples. While removing these empty regions can easily save some storage space, our findings indicate that the true challenge in losslessly compressing WSI images lies in handling their informative regions. Following this observation, we conduct an in-depth analysis of the compression properties of WSI images, building on a preliminary evaluation of existing lossless compression methods. Our analysis reveals that WSI images generically exhibit *high information irregularity*, where the high-frequency signals are widely distributed across the WSI images, and demonstrate high volatility. The irregular frequency patterns pose significant challenges to prevalent entropy-based or prediction-based compression methods. Based on those findings, we propose a simple yet effective dictionary-based lossless compression method called WISE, specifically designed for WSI images. WISE consists of three major steps: i) including a hierarchical projection coding step, ii) a bitmap coding step, and iii) a dictionary-based compressor. Through i) and ii), WISE effectively reduces the entropy of the sequence and disentangles proper locality patterns for dictionary-based compression. We conduct extensive experiments and demonstrate that by using our proposed WISE method, WSI images can be compressed to 36 times on average and up to 136 times smaller. Our contributions can be summarized as follows:

- To the best of our knowledge, this paper is the first to conduct a comprehensive study on the application of various lossless methods in WSI;
- This paper reveals why lossless methods often fall short with WSI images and offers insights into enhancing the effectiveness of lossless compression for WSI applications;
- We propose a simple yet powerful lossless compression method called WISE based on a composite encoding scheme;
- We conduct extensive experiments that verify the effectiveness of WISE.

2. Related Work

Medical Image Compression has attracted great interest in the community. Several existing works have adopted *lossy* compression techniques such as JPEG-2000 or VQ-

VAE [5, 11, 12, 15], and compress WSI patches into compact embeddings [7, 19, 27]. Reconstruction from such embeddings has also gained popularity [35, 37]; however, these approaches inevitably suffer from information loss inherent in the compressed representations.

Patch-based Lossless Compression Approaches typically divide an image into multiple small patches, which are then compressed individually. Typical patch-based compression methods include traditional image compression techniques, such as PNG [3] and TIFF [2]. In recent years, there has been emerging interest in applying deep learning methods to image compression and it has intrigued multiple advanced compression technologies, such as Bit-Swap [16], IDF [13], IDF++ [28], and ArIB-BPS [36]. However, deep learning methods are still based on entropy encoding and thus does not perform well on WSI slides.

Byte Stream-based Image Compression Approaches treats the image as a byte stream input, onto which to perform the compression operations to reduce the storage requirements. These methods can be further divided into entropy-based and dictionary-based methods. Common entropy-based methods include Huffman coding [14], Arithmetic coding [33], and PixelCNN [25, 29]. Huffman coding [14] is a classical entropy-based coding method that assigns shorter codes to more frequently occurring symbols. Arithmetic coding [33] is a more sophisticated method that assigns a unique floating-point range to the entire input sequence, often offering higher compression rates than Huffman coding. PixelCNN [25, 29] uses convolutional neural networks to model the conditional dependencies between pixels, enabling efficient compression by learning the image’s probability distribution. TRACE [22] and PAC [23] utilize lightweight transformers and MLP to model dependencies between adjacent pixels, respectively. Common dictionary-based methods include Gzip [10], LZMA [38], and Zstandard [8]. Gzip [10] is widely used in file compression, which is built upon the LZ77 [17] compression algorithm combined with Huffman coding for further optimization. LZMA [38] enhances efficiency by using bitfield-specific contexts to represent literals or phrases. Zstandard [8] combines a dictionary-matching stage (LZ77) with a large search window and a dual entropy-coding stage using Huffman coding for literals and Finite State Entropy (FSE) for sequences. However, although these methods can achieve a certain amount of lossless compression ratio on WSI images, the compression ratio is not satisfied.

3. WISE Framework

To overcome the information irregularity issue of the WSI image compression, inspired by the analysis in the previ-

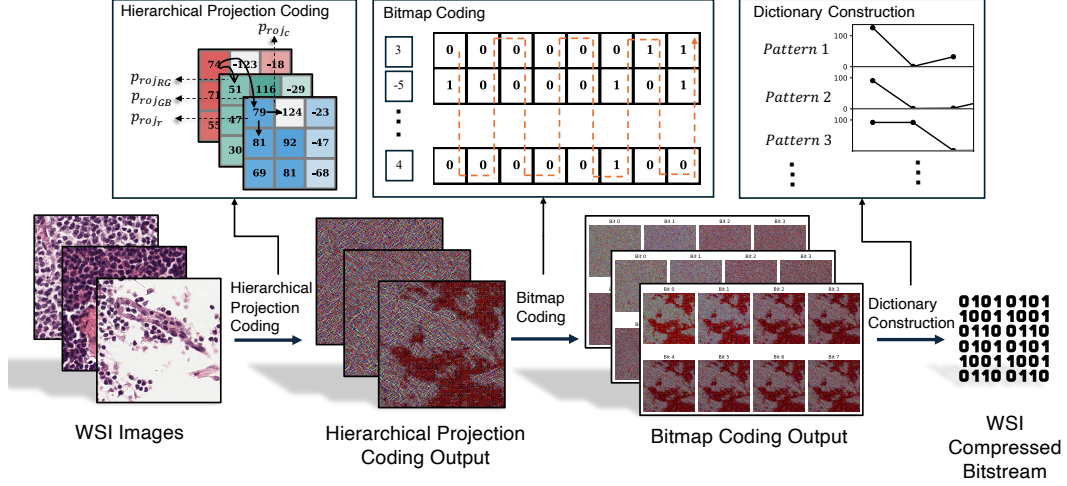


Figure 2. Illustration of the WISE pipeline. WISE compress the input WSI images through three key steps. As WSI images exhibit a high degree of information imbalance, the first step in WISE is effective information extraction, which performs image preprocessing, hierarchical projection encoding, and bitmap encoding, to transform the input image into information-dense encodings. To be specific, effective bits are gathered together for better pattern extraction. Then, in the second step, WISE constructs a dictionary to capture the byte pattern in transformed image for further compression.

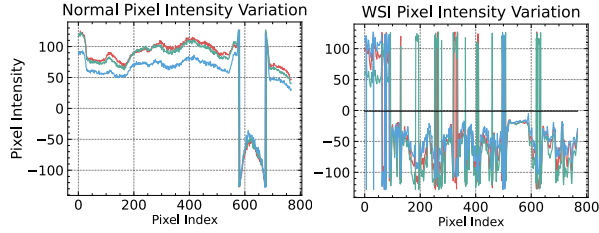


Figure 3. WSI image’s pixel intensity variation is quite different from normal image. It has a large portion of high-frequency features thus affecting the performance of entropy-based compressors.

ous section, we propose a dictionary-based compression method called WISE. In what follows, we first provide an overview of WISE, and then elaborate more on the details of each step in WISE.

3.1. Information Irregularity in WSI Images

Essentially, the success of many compression methods relies on identifying the locality properties within adjacent pixels or the pixel value distributions. Nevertheless, WSI images may demonstrate a different locality pattern from normal images such that existing compression methods fail to achieve satisfactory performance.

To begin with our analysis, we start by conducting an empirical study on the data extracted from a normal image from Kodak dataset [1] and a sample WSI image from TCGA [32]. The results are given in Table 1. Specifically, we consider five prevalent lossless compression methods, covering all the mainstream compression

approaches, including traditional compressors like Huffman [14], Pfor [39], Arithmetic Coding [33], PNG [3], and LZMA [17]. The compression ratio values reflect the effectiveness of each algorithm in reducing WSI image size, where higher values generically indicate better compression performance for the given data. Specifically, the best compressor for normal image data, PNG, can only achieve little-to-no compression ratio on WSI images. On the contrary, the dictionary-based method lzma yields a relatively high compression ratio on WSI images. The same phenomenon happens to the other compressors such as Huffman, Pfor, and PAC. Nevertheless, the simple dictionary-based compression method LZMA can achieve relatively high compression ratios on both normal and WSI images.

To better understand the failure reasons of the existing compression methods. We further analyze the characteristics of the WSI images. As the identification of locality properties has been shown to be quite sensitive to the frequency patterns in the input data [24], we focus on analyzing the frequency characteristics of the WSI images. As illustrated in Fig. 3, compared with normal images, WSI images demonstrate unique properties in the frequency domain, which we term as *information irregularity*. Specifically, the frequency information in the WSI images consists of a large part of the high-frequency features and more local extremes. The *high volatility* of the WSI data poses a significant challenge to capturing the locality for compression, especially for prediction-based compressors [11, 24].

Based on this analysis, we can better understand why dictionary-based methods, which typically perform poorly on multi-dimensional image data, outperform image-

Table 1. Empirical study for a normal image from Kodak dataset and a WSI Image from TCGA.

	Huffman	Pfor	PAC	PNG	LZMA
Normal	1.19	1.42	1.98	2.06	1.43
WSI	1.23	0.99	0.87	1.01	1.93

specific compressors like PNG on WSI images. The reason is that constructing a dictionary instead of predicting the sequential patterns is more robust to the high volatility of the WSI data. Yet, a near 2x compression ratio remains insufficient for WSI data (often with sizes of several gigabytes).

3.2. Pipeline Overview

WSI images are typically stored in a multi-resolution pyramid structure. In this paper, we focus on compressing only the base level of the pyramid, as the other layers can be generated by downsampling this primary level. We process the WSI in a patch-based manner.

The WISE framework consists of four steps, as shown in Fig. 2. In the first step, we identify the non-zero regions in the WSI images, as it is common that a huge part of the WSI images contain meaningless information (i.e., pixels in zero value). In the second step, we incorporate a hierarchical linear projection encoding algorithm to remove the unnecessary information between adjacent regions or pixels. In the third step, we investigate the correlations at an even more fine-grained level to achieve a higher compression ratio. Specifically, we consider the *correlations of the bits* from the same positions in each byte, and further compress the image reorganized via byte transformation. Finally, with proper encodings obtained from the last two steps, we introduce a dictionary-based compression method that captures the local repetitive patterns.

3.3. Preprocessing

A considerable part of sizable WSI images consists of regions outside the tissue area (so-called “white-space” without any clinically relevant information). In this study, we first developed an image-processing algorithm that can remove the unneeded background in a WSI. Specifically, we remove areas where the column sum or the row sum has the value of zero. Another pre-process we apply is to remove the alpha channel. Different from normal image data, the original WSI images contain four channels, where there is additional alpha information. We directly ignore the additional alpha channel as it contains little to no information about the diagnosis. We apply all the downstream operations to the remaining informative areas with three channels.

3.4. Hierarchical Projection Coding

Since the WSI images demonstrate heterogeneous patterns in the frequency domains, we consider a hierarchical encoding structure of the whole image information to achieve maximal compression. Instead of directly storing the entire dataset, the method stores the first-order difference between consecutive data points across three dimensions: row, column, and channel. Each pixel has three reference data points, which are taken from the nearest points in its row, column, and channel, based on physical proximity to the target pixel. This approach is chosen because WSIs often exhibit significant variability, making it challenging to model over long distances. This variability is also the reason why prediction-based methods, as discussed earlier, tend to be ineffective. The main purpose of this technique is to optimize storage space and reduce bandwidth usage by capturing and representing only the differences (or deltas) between sequential data points, such that one could use a smaller regime of values to represent pixels. The detailed algorithm is illustrated in Alg. 1.

We provide a sample to illustrate the effectiveness of the proposed Hierarchical Projection Coding. Figure 4 presents a sample matrix extracted from an informative WSI area, showing its entropy after each step of the linear projection process. The entropy H of a matrix M can be calculated using Shannon’s entropy formula [26]:

$$H(M) = - \sum_{i=1}^n p(x_i) \log_2(p(x_i))$$

where: x_i represents each unique value in the matrix, $p(x_i)$ is the probability of x_i (i.e., the frequency of x_i divided by the total number of elements), n is the total number of unique values in the matrix. As observed, with the application of each coding level, the entropy of the matrix progressively decreases. According to Shannon’s information theory, the lower the entropy, the higher the compression ratio [26]. Evaluation of compression ratio on six WSI datasets will be provided in Sec. 4.2.

3.5. Bitmap Encoding

The encodings given by the Hierarchical Projection Coding provide a new perspective of the original WSI images, where the range of the pixel values is largely shortened.

Not every bit in the current byte stream carries meaningful data. For instance, leading zeros in certain encoding schemes may not represent valid information, with only the bits following the first ‘1’ considered as containing useful data. We define these bits as effective bits, since only after encountering the first ‘1’ do subsequent bits hold significance. For example, a sample sequence such as [1, -1, 2] would have most of its bit positions empty, with only the bits following the first ‘1’ carrying relevant information.

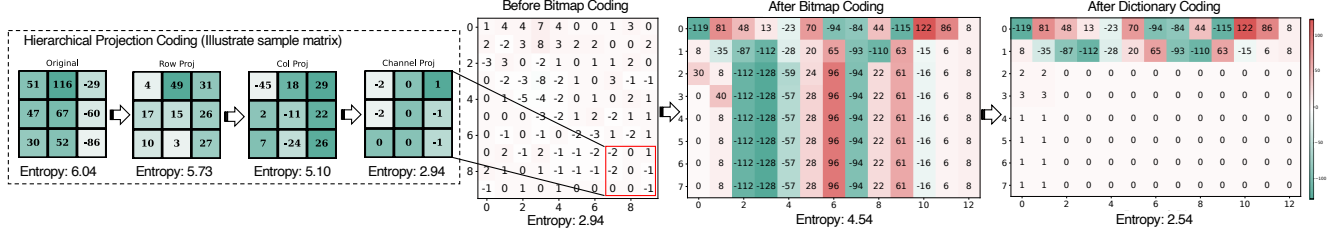


Figure 4. Detailed illustration of the Hierarchical Projection Coding, bitmap coding and dictionary coding on a sample matrix. Starting from the original information (left), Hierarchical Projection Coding performs row projection, column projection, and channel projection sequentially, which gradually reduces the entropy of the projected encodings. Then, the bitmap coding transposes the effective bits, coding the matrix with small values to a highly repetitive matrix, as shown in the bottom middle. This approach would increase the byte-based entropy since the repetitive byte is reduced. Nonetheless, after dictionary coding, the entropy would be further reduced.

Algorithm 1 Hierarchical Projection Coding for WSI Image Patch

Require: Input WSI image patch $X \in \mathbb{R}^{M \times N \times C}$
Ensure: Processed image patch $Y \in \mathbb{R}^{M \times N \times C}$

- 1: **for** each channel $c = 1 \dots C$ **do**
- 2: **for** each pixel (m, n) for $m = 2 \dots M$ **do**
- 3: $\Delta X_{m,n,c} \leftarrow X_{m,n,c} - X_{m-1,n,c}$
- 4: **end for**
- 5: **for** each pixel (m, n) for $n = 2 \dots N$ **do**
- 6: $\Delta^r X_{m,n,c} \leftarrow \Delta X_{m,n,c} - \Delta X_{m,n-1,c}$
- 7: **end for**
- 8: **for** each pixel (m, n) for $c = 2, 3$ **do**
- 9: $Y_{m,n,c} \leftarrow \Delta^r X_{m,n,c} - \Delta^r X_{m,n,1}$
- 10: **end for**
- 11: **end for**
- 12: **return** Y

The distribution of effective bits in the encodings produced by the Hierarchical Projection Coding algorithms exhibits two key characteristics: (a) higher-position bits contain less information than lower-position bits, and (b) the redundancy in low-position information increases. Therefore, it is natural to reorganize the ordering of the delta-encoded information, such that bits from the same positions in each byte have higher predictability. The high predictability enables us to transform the delta-encoded bit information to achieve a higher compression ratio, as shown in Fig. 4.

To gather the effective bits, we transpose the encodings produced by the Hierarchical Projection Coding into bitmaps, then group the bits from the same position across adjacent bytes and re-pack them into new byte representations. As shown in Fig. 4, the proposed bitmap coding transforms the sample WSI matrix into a new format, revealing numerous repetitive patterns that can be collected using a dictionary-based compressor. In the next section, we will demonstrate that applying an additional dictionary collection step on top of this bitmap coding can further reduce the entropy of the sample WSI matrix.

3.6. Dictionary Encoding

As illustrated in Fig. 4, the output of bitmap encoding contains clearly long patterns. Those patterns are raised by the grouping of effective bits and in-effective bits. Therefore, we incorporate a dictionary encoding method at the final stage of the compression process to achieve even lower entropy and a higher compression ratio. For the specific dictionary algorithm, we incorporate Lempel–Ziv–Welch (LZW) algorithm. The LZW Algorithm is a dictionary-based lossless data compression method. Unlike Arithmetic Coding, for LZW compression, there is no need to know the probability distribution of each character in the text alphabet. This allows compression to be done while the message is being received. The main idea behind compression is to encode character strings that frequently appear in the text with their index in the dictionary. The first 256 words of the dictionary are assigned to extended ASCII table characters.

After dictionary coding, as shown in Fig. 4, the entropy is further reduced from 2.94 to 2.54. This reduction is achieved because the dictionary encodes long patterns into shorter, repeated signals, transforming the matrix into a collection dominated by repetitive symbols. Additionally, by concentrating effective and ineffective bits separately, this final dictionary-based encoding step further compresses the information within the WSI image.

4. Experiments

In this section, we conduct extensive experiments on realistic WSI images to answer the following research questions:

- **RQ1:** Can WISE achieve higher compression ratios than existing methods?
- **RQ2:** Can WISE achieve higher compression throughput compared to current methods?
- **RQ3:** What role does each technique in WISE play in improving compression performance?

4.1. Experimental Setups

Datasets We selected five H&E datasets and one IHC dataset as our benchmarks. The H&E datasets include famous and well-known C16 [6], C17 [20], TCGA [32], BNCB [34], and DigestPath [9]. To further increase dataset diversity, we incorporated a recently emerging IHC [4] dataset named IHC4BC, alongside the more commonly used H&E-stained datasets.

Baselines Our selected baseline encompasses nearly all types of lossless compression algorithms, including image encoders, deep learning-based image encoders, entropy encoders, and dictionary-based encoders. Specifically, for image encoders, we selected PNG and TIFF; for deep learning-based image encoders, we chose ArIB-BPS. For entropy encoders, we included the well-known Huffman and arithmetic coding. As for dictionary-based encoders, which are of particular interest to us, we selected three options: GZip, Zstd-22(level 22), and LZMA.

Evaluation Metrics For lossless compression, the compression ratio is the primary performance metric. Additionally, we also consider throughput. Lossless compression does not cause any information loss in the image. Therefore, unlike typical lossy methods, we do not measure PSNR and SSIM for the decoded image in our experiments. However, we still use PSNR in Sec. 4.6 to measure the similarity between bitmaps, aiming to demonstrate that the bitstream after bitmap coding has higher local similarity.

4.2. Compression Performance on Gigapixel WSIs

Tab. ?? presents the main compression results comparing WISE with the existing state-of-the-art compression methods. We present a comparison of average compression ratios between six lossless baseline methods and our approach on three of the most well-known datasets—C16, C17, and TCGA. As shown, WISE achieves a substantial improvement over all other lossless compression methods. Specifically, WISE improves compression rates by 100% to 500% compared to non-dictionary-based methods. For dictionary-based methods, our compression ratio surpasses Gzip by 100%, and even outperforms the current best dictionary-based method, Zstd-22 (highest compression level of Zstd), by 70% to 80%. WISE’s compression improvements are consistent across the different datasets.

To further illustrate the characteristics of WSI data, we provide specific compression ratios for 10 sample WSI images, each approximately 2GB in size. The compression rates vary significantly across images, mainly due to two factors: (i) the substantial blank areas in WSI images, and (ii) variations in tissue density. For instance, the first sample in Fig. 2 clearly contains large blank areas. Regardless

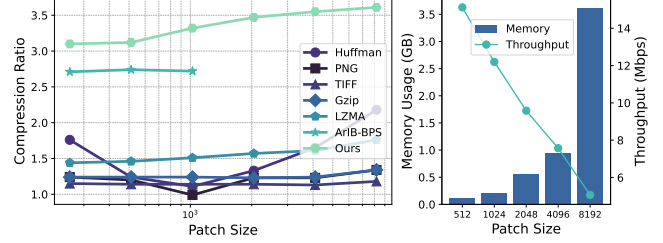


Figure 5. Left: Compression ratio variations on different image sizes. Right: WISE’s memory usage and throughput on different image sizes.

of the image type, however, WISE consistently achieves exceptionally high compression ratios.

We further investigate the compression performance on non-empty patches, as these regions pose the primary challenge in compressing WSIs. Here, we expand our dataset to include five source: C16 (256×256), C17 (256×256), TCGA (256×256), BNCB (256×256), and an IHC dataset (1024×1024). WISE consistently achieves significant compression improvements across different patch sizes, staining techniques, and datasets.

4.3. Comparison with Neural-based Compressor

We also compared our method with NN-based approaches and found that it consistently outperforms the state-of-the-art ArIB-BPS across all datasets, with improvements of 7.1%, 17.6%, and 5% on C16, C17, and TCGA, respectively. Additionally, we compared the efficiency of our method with NN-based compressors. This experiment was conducted on a CPU to ensure a fair comparison. As shown, the NN-based method is extremely slow, with a throughput of only 18.73 KB/s, whereas our method achieves a throughput of 15.1 MB/s, suitable for practical use. Furthermore, the NN-based method requires 3.81 GB of memory even for compressing a 256×256 patch, while WISE only needs 0.1 GB.

4.4. Framework Effectiveness

To further evaluate the practical applicability of our method, we explore its throughput and memory usage across different input image sizes in this section. All experiments were conducted using single-threaded Python code. We focus primarily on the compression efficiency of the informative areas, as the compression of empty areas has minimal impact. While the search algorithm for empty areas is important, it falls outside the scope of this paper, which is focused on compressing the informative regions. As observed, both throughput decreases and memory usage increases with the input size. This is primarily due to the LZW algorithm’s dictionary collection process, where memory usage grows as the input size increases, driven by the increased complexity of the search. However, this is accompanied by an

Table 2. Compression results for C16, C17, and TCGA datasets (without overall average column).

Method	C16 Dataset									
	Compression Ratios on Sample Images									
	Img1	Img2	Img3	Img4	Img5	Img6	Img7	Img8	Img9	Img10
Huffman	1.55	3.93	2.06	4.51	1.49	1.90	1.53	1.53	1.89	1.61
PNG	2.97	6.76	5.12	8.22	2.82	4.41	2.73	2.73	4.06	3.08
TIFF	2.38	9.86	6.15	14.83	2.22	3.81	2.38	2.38	4.11	3.02
Gzip	2.40	10.88	6.56	16.80	2.23	3.88	2.40	2.40	4.58	3.06
Lzma	3.44	16.83	9.25	26.50	3.14	5.46	3.37	3.37	6.58	4.34
Zstd-22	3.12	15.16	8.54	23.96	2.82	4.97	3.62	3.04	11.89	3.90
WISE	4.74	22.94	12.52	37.18	4.48	7.67	5.67	4.54	17.79	6.03

Method	C17 Dataset									
	Compression Ratios on Sample Images									
	Img1	Img2	Img3	Img4	Img5	Img6	Img7	Img8	Img9	Img10
Huffman	3.55	3.77	5.09	3.80	5.91	4.88	4.30	4.67	4.38	3.48
PNG	7.19	8.34	14.01	8.48	4.99	13.39	9.91	12.59	11.92	4.59
TIFF	7.56	8.23	23.35	9.04	27.43	18.89	11.25	17.47	12.04	6.55
Gzip	7.96	8.75	25.93	9.17	68.84	19.31	12.20	17.68	12.85	7.47
Lzma	11.33	12.21	36.82	12.99	100.19	26.92	16.96	25.25	17.99	10.53
Zstd-22	10.16	10.89	32.61	11.61	89.60	23.96	15.14	22.48	16.06	9.41
WISE	15.82	16.56	50.68	17.84	136.15	37.31	23.51	35.56	24.05	14.91

Method	TCGA Dataset									
	Compression Ratios on Sample Images									
	Img1	Img2	Img3	Img4	Img5	Img6	Img7	Img8	Img9	Img10
Huffman	2.11	2.63	2.01	2.34	1.39	1.77	1.73	1.34	1.50	2.43
PNG	3.54	4.41	3.95	2.95	1.97	1.95	2.19	2.11	2.04	4.81
TIFF	3.74	4.93	3.22	5.03	1.71	2.17	2.52	1.69	2.13	6.44
Gzip	3.78	5.01	3.53	5.11	1.79	2.52	2.53	1.69	2.14	6.56
Lzma	5.02	6.66	4.67	6.80	2.45	3.31	3.28	2.28	2.88	9.18
Zstd-22	4.45	6.00	4.16	6.06	2.18	2.97	2.94	2.04	2.57	7.96
WISE	6.26	8.02	6.02	8.35	3.69	4.59	4.30	3.17	3.85	11.16

Table 3. Comparison of our method with other traditional compressor on patches on various WSI datasets including C16, C17, TCGA, BNCB, and IHC4BC.

Method	C16	C17	BNCB	TCGA	IHC
PNG	1.64	1.96	1.46	1.04	2.75
TIFF	1.16	1.15	1.16	1.15	2.32
Huffman	1.08	1.12	1.12	1.10	1.45
Arithmetic	1.09	1.12	1.13	1.13	1.46
Gzip	1.15	1.27	1.15	1.14	2.00
Zstd-22	1.24	1.36	1.17	1.16	2.48
Lzma	1.32	1.46	1.29	1.32	2.60
WISE	2.82	3.20	1.63	2.50	3.85

Table 4. Comparison of WISE with Neural-Network compressor on patches from various WSI datasets including C16, C17, TCGA, BNCB, and IHC4BC. Throu. stands for throughput and Mem. stands for memory usage.

Method	C16	C17	TCGA	IHC	Throu.	Mem.
ArIB-BPS	2.63	2.72	2.38	3.81	18.73KB/s	3.81GB
WISE	2.82	3.20	2.50	3.85	15.1MB/s	0.1GB

improvement in compression ratio. It is worth noting that when the input size reaches $8192 \times 8192 \times 3$, our method achieves a memory usage comparable to ArIB-BPS, but with a significantly lower memory footprint of only 3 GB.

4.5. Ablation Study

Effectiveness of Proposed Techniques We conducted ablation experiments on C16, C17, and TCGA to demonstrate the effectiveness of each proposed technique. The baseline shown in the table is a basic dictionary-based algorithm, LZMA. Results indicate that applying LZMA alone is not highly effective, even though, as discussed in the previous section, LZMA provides a certain level of compression compared to other lossless algorithms. Medcomp improves the compression ratio of the dictionary-based method to 113.5%~130.2%. Specifically, the proposed Hierarchical encoding achieves relative improvements over the baseline by 69.6%, 76.9%, and 73.5% on C16, C17, and TCGA, respectively. Furthermore, the proposed bitmap encoding can further increase the compression ratio to 125.6%, 130.2%, and 113.5%.

Table 5. Ablation Study on various WSI datasets including C16, C17, TCGA.

Method	C16	C17	TCGA
Baseline	1.25	1.39	1.17
+Hierarchical Linear Projection	2.12	2.46	2.03
+Bitmap Encoding	2.82	3.20	2.50

Compression ratio under different input image sizes.

In Fig. 5, we show the variation in compression ratios for six baselines and our method across different input image sizes. The ArIB-BPS, a neural network-based approach, was tested on a single 24GB Nvidia-4090 GPU. Due to memory constraints, it was limited to processing images of size 1024×1024 and smaller. As observed, the compression ratio for images smaller than 1024×1024 remains largely unchanged. Another key observation is that all dictionary-based compression methods show improved compression ratios as the input size increases. This is intuitive, as dictionary-based methods can better exploit previously collected patterns when more input data is available. In contrast, the compression ratios for entropy-based methods are less consistent and do not show the same trend.

4.6. Empirical Study

We present an empirical study on a specific WSI patch to validate several conclusions discussed earlier from a different perspective. Figs. 6 and Fig 7 illustrate the similarity between the original WSI patch and its bitmap representation after delta coding, with similarity measured using PSNR. Bitmap transposition improves the efficiency of dictionary collection. From Fig. 6, it can be observed that after performing a bitmap split on the original image, the bitmap similarity for each channel is around 3 to approximately 18. Moreover, the similarity between lower-bit bitmaps is lower, while the similarity between higher-bit bitmaps is higher. Next, we examine Fig 7, which shows the similarity between the bitmaps of the sample patch after applying the proposed hierarchical linear coding. The first conclusion from this figure is that the similarity between the bitmaps after hierarchical linear coding significantly increases compared to the original patch, rising from 3 to 48. In fact, some higher-bit bitmaps reach an infinite PSNR value, as they have no information variance and are identical, confirming that "the range of pixel values is largely shortened by hierarchical linear coding" as stated in Sec. 3.5. Hierarchical linear coding further concentrates information in the lower bit positions. The second conclusion is that, due to the high similarity between the bitmaps after hierarchical linear coding, they are more easily grouped into similar patterns, making them more compressible. Indeed, during the experimental process, we observed a similar phenomenon in the dictionary coding stage. The collected patterns across dif-

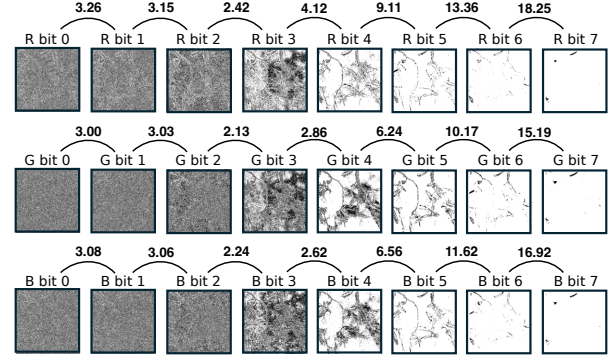


Figure 6. PSNR of bitmaps before Bitmap Coding. As illustrated, bitmaps in higher bit positions have larger empty areas and higher PSNR values, indicating they contain less information..

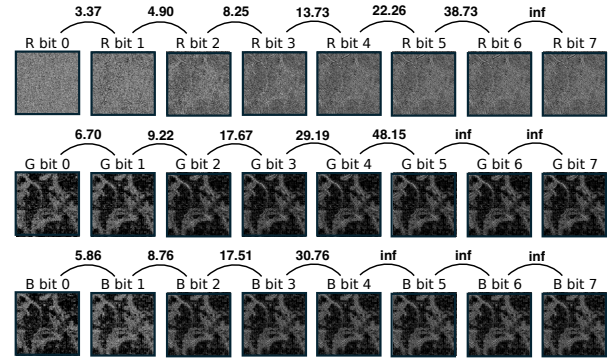


Figure 7. PSNR of bitmaps after Bitmap Coding. As illustrated, the PSNR between bitmaps increases significantly, indicating more repetitive patterns among bitmaps, which enhances the efficiency of dictionary coding.

ferent channels and bitmaps showed high similarity.

5. Conclusions

In this work, we explored applying compression techniques to WSI images. Through the study of existing compression techniques on WSI images, we identified several unique challenges in compressing WSI images, including how to deal with information sparsity and information irregularity. To resolve those challenges, we developed a simple yet effective WSI specialized compressor called WISE. WISE is built upon a hierarchical encoding strategy and a dictionary-based compression method, which can be further combined with various entropy-based compression methods to further improve the compression ratios. Extensive experiments on realistic WSI benchmarks show that WISE achieves up to $136\times$ and on average $36\times$ compression over prior methods, while retaining low decoding complexity and preserving critical diagnostic content, making it highly practical for deployment in real-world digital pathology workflows across diverse tissue types and scanning conditions.

References

- [1] Kodak lossless true color image suite. <http://r0k.us/graphics/kodak/>. Accessed: 2024-11-14. 3
- [2] N. Abe and et al. Tiff: Tagged image file format, 1986. Accessed: 2024-11-07. 2
- [3] David Adams and et al. Png: The portable network graphics (png) format, 1996. Accessed: 2024-11-07. 2, 3
- [4] Amir Akbarnejad, Nilanjan Ray, Penny J Barnes, and Gilbert Bigras. Predicting ki67, er, pr, and her2 statuses from h&e-stained breast cancer images. *arXiv preprint arXiv:2308.01982*, 2023. 6
- [5] Agnes Barsi, Suwendu Chandan Nayak, Sasmita Parida, and Raj Mani Shukla. A deep learning-based compression and classification technique for whole slide histopathology images. *International Journal of Information Technology*, pages 1–10, 2024. 1, 2
- [6] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermesen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017. 1, 6
- [7] Tsai Hor Chan, Fernando Julio Cendra, Lan Ma, Guosheng Yin, and Lequan Yu. Histopathology whole slide image analysis with heterogeneous graph representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15661–15670, 2023. 2
- [8] Y. Collet. Zstd github repository from facebook., 2016. 2
- [9] Qian Da, Xiaodi Huang, Zhongyu Li, Yanfei Zuo, Chenbin Zhang, Jingxin Liu, Wen Chen, Jiahui Li, Dou Xu, Zhiqiang Hu, et al. Digestpath: A benchmark dataset with challenge review for the pathological detection and segmentation of digestive-system. *Medical Image Analysis*, 80:102485, 2022. 6
- [10] Peter Deutsch. GZIP file format specification version 4.3. *RFC*, 1952. 2
- [11] Shahriar Faghani, D Chamil Codipilly, Mana Moassefi, Prasad G Iyer, and Bradley J Erickson. Optimizing storage and computational efficiency: An efficient algorithm for whole slide image size reduction. *Mayo Clinic Proceedings: Digital Health*, 1(3):419–424, 2023. 1, 2, 3
- [12] Henrik Helin, Teemu Tolonen, Onni Ylinen, Petteri Tolonen, Juha Napankangas, and Jorma Isola. Optimized jpeg 2000 compression for efficient storage of histopathological whole-slide images. *Journal of pathology informatics*, 9(1): 20, 2018. 1, 2
- [13] Emiel Hooeboom, 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] D. A. Huffman. *A Method for the Construction of Minimum-Redundancy Codes*. 1952. Accessed: 2024-11-07. 2, 3
- [15] Jason Keighley, Marc de Kamps, Alexander Wright, and Darren Treanor. Digital pathology whole slide image compression with vector quantized variational autoencoders. In *Medical Imaging 2023: Digital and Computational Pathology*, pages 344–353. SPIE, 2023. 1, 2
- [16] Kingma et al. Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables. In *ICML*, pages 3408–3417. PMLR, 2019. 2
- [17] Kinsner et al. The lempel-ziv-welch (lzw) data compression algorithm for packet radio. In *[Proceedings] WES-CANEX'91*, pages 225–229. IEEE, 1991. 2, 3
- [18] Neeta Kumar, Ruchika Gupta, and Sanjay Gupta. Whole slide imaging (wsi) in pathology: Current perspectives and future directions. *Journal of Digital Imaging*, 33:1034 – 1040, 2020. 1
- [19] Tristan Lazard, Marvin Lerousseau, Etienne Decencière, and Thomas Walter. Self-supervised extreme compression of gigapixel images. 2
- [20] Geert Litjens, Peter Bandi, Babak Ehteshami Bejnordi, Oscar Geessink, Maschenka Balkenhol, Peter Bult, Altuna Halilovic, Meyke Hermesen, Rob van de Loo, Rob Vogels, et al. 1399 h&e-stained sentinel lymph node sections of breast cancer patients: the camelyon dataset. *GigaScience*, 7(6):giy065, 2018. 1, 6
- [21] Chen Liu, Danqi Liao, Alejandro Parada-Mayorga, Alejandro Ribeiro, Marcello DiStasio, and Smita Krishnaswamy. Diffkillr: Killing and recreating diffeomorphisms for cell annotation in dense microscopy images. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 1
- [22] Yu Mao, Yufei Cui, Tei-Wei Kuo, and Chun Jason Xue. Trace: A fast transformer-based general-purpose lossless compressor. In *Proceedings of the ACM Web Conference 2022*, pages 1829–1838, 2022. 2
- [23] Yu Mao, Jingzong Li, Yufei Cui, and Jason Chun Xue. Faster and stronger lossless compression with optimized autoregressive framework. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2023. 2
- [24] 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. 3
- [25] Tim Salimans, Andrej Karpathy, Xi Chen, and D.P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, pages 5446–5454, 2017. Accessed: 2024-11-07. 2
- [26] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948. 4
- [27] David Tellez, Diederik Höppener, Cornelis Verhoef, Dirk Grünhagen, Pieter Nierop, Michal Drozdal, Jeroen Laak, and Francesco Ciompi. Extending unsupervised neural image compression with supervised multitask learning. In *Medical Imaging with Deep Learning*, pages 770–783. PMLR, 2020. 2
- [28] van den Berg et al. Idf++: Analyzing and improving integer discrete flows for lossless compression. In *ICLR*, 2020. 2

- [29] A. Van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1747–1756, 2016. Accessed: 2024-11-07. [2](#)
- [30] Jun Wang, Yu Mao, Nan Guan, and Chun Jason Xue. Advances in multiple instance learning for whole slide image analysis: Techniques, challenges, and future directions. *arXiv preprint arXiv:2408.09476*, 2024. [1](#)
- [31] Jun Wang, Yu Mao, Nan Guan, and Chun Jason Xue. Shapcat: A interpretable multi-modal framework enhancing wsi classification via virtual staining and shapley-value-based multimodal fusion. *arXiv preprint arXiv:2410.01408*, 2024. [1](#)
- [32] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013. [3](#), [6](#)
- [33] Witten et al. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987. [2](#), [3](#)
- [34] Feng Xu, Chuang Zhu, Wenqi Tang, Ying Wang, Yu Zhang, Jie Li, Hongchuan Jiang, Zhongyue Shi, Jun Liu, and Mulan Jin. Predicting axillary lymph node metastasis in early breast cancer using deep learning on primary tumor biopsy slides. *Frontiers in oncology*, 11:759007, 2021. [6](#)
- [35] Qingchao Zhang, Xiaojing Ye, and Yunmei Chen. Nonsmooth nonconvex ldct image reconstruction via learned descent algorithm. In *Developments in X-Ray Tomography XIII*, page 132. SPIE, 2021. [2](#)
- [36] 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. [2](#)
- [37] Jun Zhuang and Dali Wang. Geometrically matched multi-source microscopic image synthesis using bidirectional adversarial networks. In *International Conference on Medical Imaging and Computer-Aided Diagnosis*, pages 79–88. Springer, 2021. [2](#)
- [38] Ziv et al. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, 23(3): 337–343, 1977. [2](#)
- [39] Marcin Zukowski, Stefan Heman, Niels Nes, and Peter A. Boncz. Super-scalar ram-cpu cache compression. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE’06)*, pages 59–59. IEEE, 2006. [3](#)