Learned Compression for Compressed Learning

Dan Jacobellis and Neeraja J. Yadwadkar

University of Texas at Austin
Austin, TX, 78712, USA
danjacobellis@utexas.edu
neeraja@austin.utexas.edu

Abstract

Modern sensors produce increasingly rich streams of high-resolution data. Due to resource constraints, machine learning systems discard the vast majority of this information via resolution reduction. Compressed-domain learning allows models to operate on compact latent representations, allowing higher effective resolution for the same budget. However, existing compression systems are not ideal for compressed learning. Linear transform coding and end-to-end learned compression systems reduce bitrate, but do not uniformly reduce dimensionality; thus, they do not meaningfully increase efficiency. Generative autoencoders reduce dimensionality, but their adversarial or perceptual objectives lead to significant information loss. To address these limitations, we introduce WaLLoC (**Wa**velet **L**earned **Lo**ssy **C**ompression), a neural codec architecture that combines linear transform coding with nonlinear dimensionality-reducing autoencoders. WaLLoC sandwiches a shallow, asymmetric autoencoder and entropy bottleneck between an invertible wavelet packet transform. Across several key metrics, WaLLoC outperforms the autoencoders used in state-of-the-art latent diffusion models. WaLLoC does not require perceptual or adversarial losses to represent high-frequency detail, providing compatibility with modalities beyond RGB images and stereo audio. WaLLoC's encoder consists almost entirely of linear operations, making it exceptionally efficient and suitable for mobile computing, remote sensing, and learning directly from compressed data. We demonstrate WaLLoC's capability for compressed-domain learning across several tasks, including image classification, colorization, document understanding, and music source separation. Our code, experiments, and pre-trained audio and image codecs are available at `https://ut-sysml.org/walloc/`.

1 Introduction

In the last decade, deep neural networks (DNNs) have rapidly evolved from simple classifiers [1, 2] to domain-specific and multi-modal foundation models [3, 4]. With this shift, models are increasingly able to make use of minute and high-frequency signal details. For example, when increasing the resolution of PaliGemma from 224^2 to 896^2 pixels (Figure 1), its ability to analyze documents increases from 44% to 85% ANLS [4]. However, operating at this increased resolution requires significantly more GPU memory (21 vs 8 GB) and leads to 4× higher latency.

Compressed-domain learning [5, 6, 7] has been proposed to improve the trade-off between model accuracy and compute needs. In this paradigm, the model operates on low-dimensional (lossy) compressed data, thereby enabling dramatic reductions in compute cost and inference latency while maintaining model accuracy. However, existing lossy compression methods, coming from three main categories, are not ideal for compressed-domain learning. (a) Linear transform coding methods (e.g.,

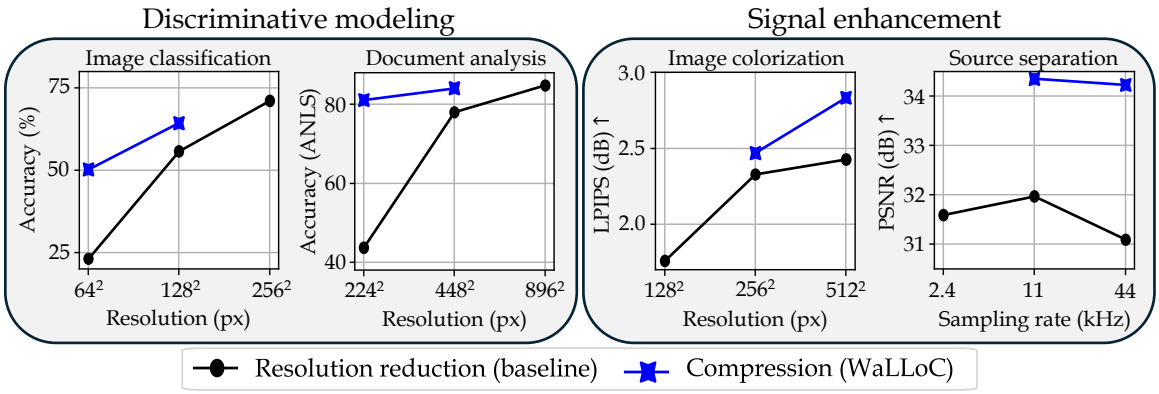

Figure 1: In discriminative models (left), resolution reduction increases training and inference efficiency, but significantly degrades accuracy. Replacing resolution reduction with WaLLoC leads to significantly higher accuracy, while providing the same degree of acceleration. For signal enhancement (right), WaLLoC provides better quality when scaling to high resolutions compared to directly operating on image pixels or audio samples.

JPEG, MP3) reduce bitrate via energy-compacting time-frequency transforms, but do not meaningfully reduce dimensionality or increase efficiency of downstream models. (b) End-to-end learned codecs [8] achieve better rate-distortion performance and modestly reduce dimension via nonlinear autoencoders, but high encoding overhead negates the benefits of compressed learning. (c) Generative autoencoders [7, 9] significantly reduce dimension, but do so by synthesizing rather than preserving details—leading to poor performance in discriminative tasks [10].

In this work, we introduce WaLLoC (Wavelet Learned Lossy Compression), an architecture for learned compression that simultaneously satisfies three key requirements of compressed-domain learning:

1. **Computationally efficient encoding** to reduce overhead in compressed-domain learning and support resource constrained mobile and remote sensors. WaLLoC uses the computationally cheap and invertible wavelet packet transform [14] to expose signal redundancies prior to autoencoding. This allows us to replace the encoding DNN with a single linear layer (<100k parameters) without significant loss in quality. As shown in Figure 2, WaLLoC incurs less than five percent of the encoding cost compared to other neural codecs.
2. **High compression ratio** for storage and transmission efficiency. Lossy codecs typically achieve high compression by combining quantization and entropy coding. However, naive quantization of autoencoder latents leads to unpredictable and unbounded distortion. Instead, we apply additive noise during training as an entropy bottleneck [8], leading to quantization-resilient latents. When combined with entropy coding, WaLLoC achieves nearly 6× higher compression ratio compared to the VAE used in Stable Diffusion 3 [12], despite offering a higher degree of dimensionality reduction and similar quality (Figure 2, Table 1).
3. **Dimensionality reduction** to accelerate compressed-domain modeling. WaLLoC's encoder projects high-dimensional signal patches to low-dimensional latent representations, providing a reduction of up to 20×. This allows WaLLoC to be

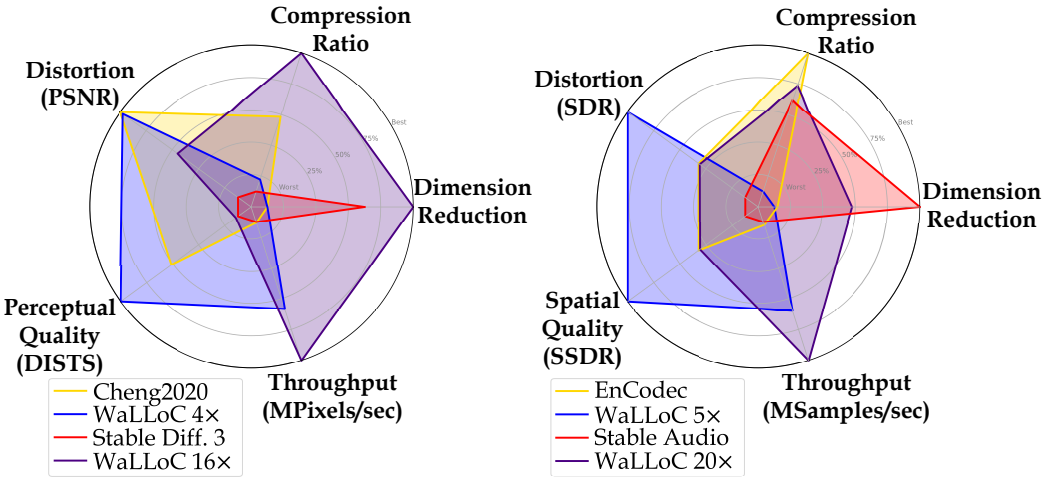

Figure 2: Comparison of our proposed method (WaLLoC) with other autoencoder designs for RGB Images (Cheng2020 [11], Stable Diffusion 3 [12]) and stereo audio (EnCodec [13], Stable Audio [9]). Additional metrics are reported in Tables 1 and 2.

used as a drop-in replacement for resolution reduction while providing superior detail preservation and downstream accuracy.

Our main contributions are as follows:

- We evaluate the trade-offs between three existing approaches to lossy compression—(1) linear transform coding, (2) end-to-end learned compression, and (3) generative autoencoders. We identify key limitations of each when used as a replacement for resolution reduction in machine learning models.
- We introduce WaLLoC, a modality-agnostic lossy compression framework that simultaneously provides (1) efficient encoding, (2) favorable rate-distortion trade-off, and (3) uniform dimensionality reduction.
- Using our proposed framework, we build RGB image and stereo audio codecs that outperform other autoencoder designs across several key metrics (Figure 2). We evaluate WaLLoC's efficacy for accelerating various machine learning models via compressed domain operation. Across each of the four tasks —image classification, colorization, document understanding, and music source separation— WaLLoC outperforms resolution reduction by a wide margin (Figure 1).

2 Background: Compressed-Domain Learning

Methods for compressed-domain learning can be grouped based on the type of compression (1) linear transform coding [5], (2) end-to-end learned compression [6, 15], and (3) and generative autoencoders [7, 9].

Linear transform coding. Conventional lossy compression standards—such as JPEG and MP3 [14] —are based on linear transform coding (LTC). Linear and invertible transforms like the discrete cosine transform (DCT) or discrete wavelet transform (DWT) eliminate redundancies while concentrating signal energy into fewer coefficients nearly optimally and remaining computationally efficient. Quantization

allocates bits to each frequency band according to perceptual models, leading to high compression ratios with minimal perceived distortion. LTC is often combined with resolution reduction (e.g. chroma downsampling in JPEG), but does not provide consistent or uniform dimensionality reduction. LTC can improve downstream learning [5] but does not address the computational issues of scaling DNNs to high resolution.

End-to-end learned compression. Nonlinear autoencoders that are jointly optimized for both rate and distortion [8] achieve higher compression ratios than LTC, but require more computation [16] and offer limited dimensionality reduction—typically $4\times$ [11]. Efficient decoding, and machine vision without decoding have been explored [17, 15], but encoding overhead remains significant.

Generative autoencoders. Compressed-domain learning underpins recent breakthroughs in high-resolution diffusion [7], masked autoencoding [18], and autoregressive [19] generative models [7]. These applications use a low-resolution generative model paired with a generative, adversarial, and dimensionality-reducing autoencoder (GADR-AE)—which we define as any autoencoder offering $> 4\times$ dimensionality reduction (DR) and trained using adversarial and perceptual losses [20]. GADR-AEs produce low-dimensional latent representations that are up to 64 times smaller than the original input [9]. However, they lose significant detail in the process, so adversarial and perceptual objectives are employed to re-synthesize details in the decoder [7]. Existing GADR-AEs are computationally cheap compared to the generative models they enable, but expensive compared to discriminative models. For example, compared to the widely used EfficientNet model [21], the encoder used in Stable Diffusion's VAE has $> 6\times$ more parameters (34.3M vs 5.3M) and requires $> 400\times$ more GFLOPs (163 vs 0.39) [7].

3 Proposed Method: Design and Implementation

WaLLoC's design aims at achieving three goals: computationally efficient encoding, high compression ratio, and uniform dimensionality reduction. We note several key insights that allow us to address the limitations of previous designs that stand in the way of achieving these goals. Each of these goals, limitations, and insights motivate the core design components of WaLLoC, shown in Figure 3.

3.1 Achieving computationally efficient encoding.

Two main barriers stand in the way of efficient encoding. (a) poor scaling of autoencoder performance with resolution, and (b) difficulty in preserving quality with lightweight encoders.

(a) Resolution scaling. In existing autoencoder designs [8, 11, 7, 13, 9], a hierarchy of DNN layers progressively reduce the spatial or temporal resolution while increasing the channel dimension. However, the initial layers of the encoder and the final layers of the decoder operate at the original resolution, leading to significant memory and computational requirements [22]. The wavelet packet transform (WPT), shown in Figure 4, is a linear and invertible transform that performs an analogous operation. In each level of the WPT, the signal is divided into high- and low-frequency components, then downsampled by a factor of two. By recursively applying this process, the WPT

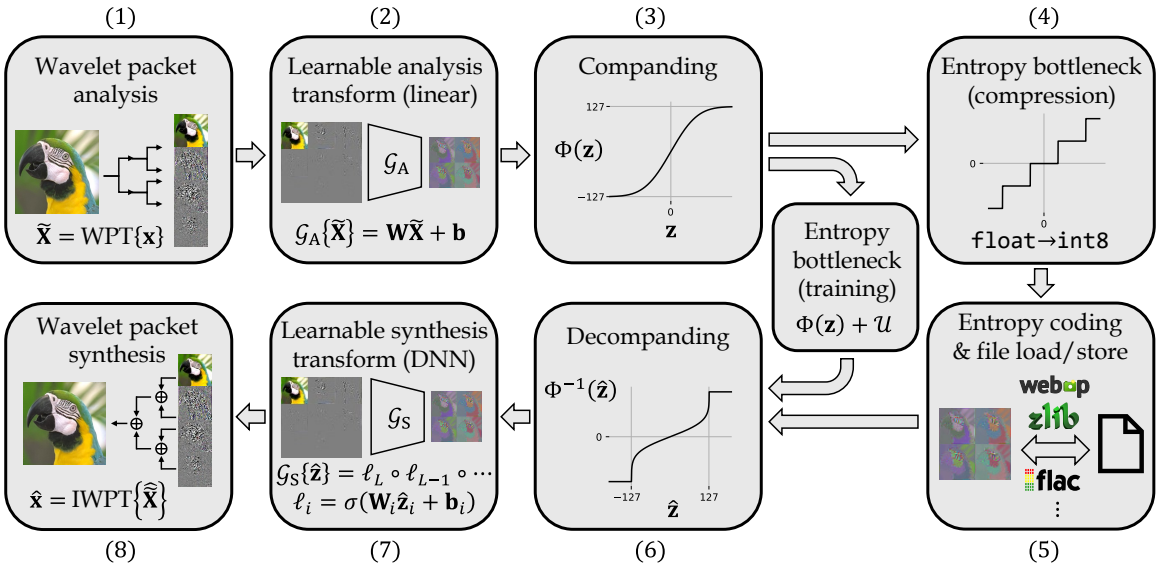

Figure 3: WaLLoC's encode-decode pipeline. The entropy bottleneck and entropy coding steps are only required to achieve high compression ratios for storage and transmission. For compressed-domain learning where dimensionality reduction is the primary goal, these steps can be skipped to reduce overhead and completely eliminate CPU-GPU transfers.

allows spatial and temporal resolution to be traded off for frequency resolution with minimal computation and no loss of information. In WaLLoC, we exploit this property by sandwiching the learnable analysis and synthesis transforms between the WPT and its inverse—allowing all neural network layers to operate at low resolution.

(b) Loss of quality in lightweight encoders. Previous efforts use reduced hidden dimension and distillation to reduce the computational cost of pixel-based autoencoders but incur a significant loss of detail in the process [23]. However, the WPT's ability to isolate important signal components from redundancies alleviates this issue. Additionally, it is possible to exploit asymmetry between the encoder and decoder. The decoder objective—disentangling mixed signal components—is difficult and requires a complex DNN-based transform. In contrast, the encoder objective—discarding signal redundancies—becomes trivial after applying the WPT. Thus, WaLLoC sandwiches an asymmetric autoencoder—consisting of a shallow, linear analysis transform and a deep, nonlinear synthesis transform—between the WPT and its inverse.

3.2 Achieving high compression ratio.

Quantization is the primary mechanism used in lossy compression to reduce bit rate and achieve a high compression ratio. However, the GADR-AEs that provide good dimensionality reduction are not compatible with quantization. For example, quantization of Stable Diffusion's VAE latents leads to severe distortion [23] However if quantization is applied, very high compression ratios can be achieved via entropy coding. In WaLLoC, we incorporate an entropy bottleneck—additive noise applied during training that guarantees quantization resilience during inference [8]. We optimize the noise scale for 8-bit quantization, allowing us to use standard lossless codecs (e.g PNG or WeBP) for entropy coding. This combination provides an additional

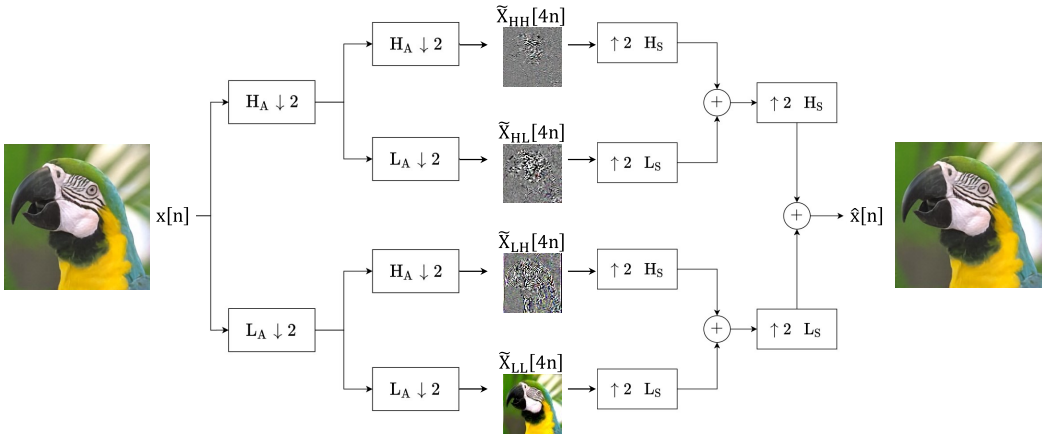

Figure 4: Example of forward and inverse WPT with $J = 2$ levels. Each level applies filters L_A and H_A independently to each of the signal channels, followed by downsampling by a factor of two ($\downarrow 2$). An inverse level consists of upsampling ($\uparrow 2$) followed by L_S and H_S, then summing the two channels. The full WPT $\widetilde{\mathbf{X}}$ of consists of J levels.

compression multiplier of up to 12× compared to reducing the dimension only.

3.3 Achieving uniform dimensionality reduction.

In addition to quantization, neural codecs achieve high compression ratios via a loss term that encourages sparse, rather than low-dimensional latents [8]. Using this objective, it is possible to drive the energy of many of the latent dimensions to zero [24]. However, this type of non-uniform dimensionality reduction is difficult to exploit in compressed-domain learning. In WaLLoC, the analysis transform uniformly reduces the dimension by a fixed rate, making it a suitable replacement for resolution reduction in accelerating downstream models.

3.4 WaLLoC Implementation

WaLLoC's encoder consists of five stages as shown in Figure 3: (1) wavelet packet transform (WPT) to trade-off spatial or temporal resolution with channel resolution (2) learned analysis transform to reduce dimensionality (3) companding to whiten the latent distribution (4) entropy bottleneck to provide resilience to quantization and (5) entropy coding to provide high compression ratios. The decoder consists of the reverse operations: (5) entropy decoding, (6) decompanding, (7) learned synthesis transform, and (8) inverse WPT. We now provide detailed explanations for each component.

Wavelet packet transform. Figure 4 shows the workflow of the wavelet packet transform (WPT) and its inverse. We use the Cohen–Daubechies–Feauveau (CDF) 9/7 wavelet [14] to construct the a dyadic filterbank consisting of highpass analysis (H_A), lowpass analysis (L_A), highpass synthesis (H_S), and lowpass synthesis (L_S) filters. The CDF 9/7 wavelet is chosen for its balance between computational efficiency and energy compaction. Since these same filters are used in the JPEG 2000 standard, they are widely supported in software. The WPT reduces the input resolution $R_{\mathbf{x}}$ and increases the input channel count $C_{\mathbf{x}}$ by a factor 2^J for 1D signals (audio) and by 4^J for 2D signals (images), but is linear and invertible. For stereo audio, we use

$J = 8$, resulting in $C_{\widetilde{\mathbf{X}}} = 512$ channels after the WPT. For RGB images, we use $J = 3$, resulting in $C_{\widetilde{\mathbf{X}}} = 192$.

Autoencoder and entropy bottleneck. The output of the WPT $\widetilde{\mathbf{X}}$ is projected to a latent representation $\mathbf{z}$ via a learnable analysis transform $\mathcal{G}_{\mathrm{A}}$, which consists of a single linear layer. The latent dimension $C_{\mathbf{z}}$ is a hyperparameter chosen based on the desired degree of dimensionality reduction. To achieve quantization-resilient latent representations, we adopt the entropy bottleneck method from end-to-end learned compression [8], which consists of adding uniform noise $\mathcal{U}[-0.5, 0.5]$ to the latent representation during training. Since the sub-band wavelet coefficients of many natural signals follow a generalized Gaussian distribution (GGD) [25], we apply the Gaussian CDF $\Phi(\mathbf{z})$ as a companding operation prior to the entropy bottleneck. Thus, the final encoder output is $\hat{\mathbf{z}}_{\mathrm{t}} = \Phi(z) + \mathcal{U}$ during training and $\hat{\mathbf{z}}_{\mathrm{c}} = \mathrm{round}\left(\Phi(\mathbf{z})\right)$ during the compression pipeline. We scale the inputs and outputs of the companding operation Φ to guarantee latents in the range [-127, 127], which in turn guarantees that $\hat{\mathrm{z}}_{\mathrm{c}}$ does not underflow or overflow when quantized to a signed 8-bit integer. The decoder consists of a learnable synthesis transform $\mathcal{G}_{\mathrm{S}}$ followed by the IWPT. $\mathcal{G}_{\mathrm{S}}$ is a convolutional neural network consisting the same residual blocks used in Stable Audio [9] and Stable Diffusion 3 [12] for 1D and 2D signals respectively. We use a hidden dimension of $C_{\mathrm{hidden}} = 768$ for both the RGB image decoder and stereo audio decoders. Additional implementation details are available in our public code repositories [1].

Entropy coding. After quantization, an additional lossless compression step can be applied. We performed preliminary tests using zlib , PNG (Deflate), and the lossless mode of WebP . We found that WebP's entropy coding provided the best compression ratio—even for audio signals—while maintaining high throughput and compatibility with ML frameworks like PyTorch. Since WebP expects 24-bit RGB inputs, we rearrange the multi-channel 8-bit latent tensor into groups of three and concatenate channel groups along the temporal or spatial dimensions.

Training. We train four codecs—two for stereo audio (5×, 20×) and two for RGB images (4×, 20×)—on The lossless MUSDB18-HQ [26] and LSDIR [27] datasets. In each case, the training objective is to minimize mean squared reconstruction error when latents are subjected to uniform additive noise in the range [-0.5,0.5].

4 Evaluation

We conduct a comprehensive evaluation of WaLLoC to demonstrate its efficacy for compressed domain learning. Our evaluation consists of two main parts. (1) **Compression trade-off analysis.** We compare WaLLoC against other lossy codecs in terms of the trade-off between dimensionality reduction, compression ratio, distortion, perceptual quality, and computation (Section 4.1). (2) **Compressed learning and resolution scaling.** We train and evaluate various machine learning models on representations produced by WaLLoC, and compare their resolution scaling properties to pixel-based and sample-based versions (Section 4.2).

[1] Code repository for WaLLoC. Code and and experiments for compressed-domain learning.

Method	DR	CR	Enc.	Dec.	PSNR	MS-SSIM	LPIPS_{dB}	DISTS_{dB}
WEBP	1	**40.6**	**22.1**	**2746**	28.2	0.96	5.94	13.1
Cheng2020	4	21.8	0.289	0.139	**33.8**	**0.99**	<u>8.82</u>	<u>16.9</u>
WaLLoC	4	8.53	<u>14.0</u>	<u>0.47</u>	<u>33.5</u>	**0.99**	**11.2**	**19.3**
SD 3.0	<u>12</u>	6.00	0.195	0.101	20.9	0.84	8.33	13.8
WaLLoC	**16**	<u>35.2</u>	**22.1**	0.466	27.5	<u>0.97</u>	6.51	13.9

Table 1: RGB image compression comparison. Metrics: dimensionality reduction (DR), compression ratio (CR), encoding (Enc.) and decoding (Dec.) throughput (Megapixels/sec, CPU), distortion (PSNR, MS-SSIM) and perceptual quality (LPIPS_{dB}, DISTS_{dB}). We report $\text{LPIPS}_{\text{dB}} = -10\log_{10}(\text{LPIPS})$ and $\text{DISTS}_{\text{dB}} = -10\log_{10}(\text{DISTS})$ so that higher values are better for each metric. For each metric, the best performing method is in boldface and the second best is underlined.

4.1 Compression trade-off analysis

We compare WaLLoC against other popular conventional and neural codecs [12, 9, 13, 11] across five key metrics: (1) degree of dimensionality reduction, (2) compression ratio, (3) distortion, (4) perceptual quality, and (5) computation. For images, distortion is measured via PSNR and MS-SSIM [28], while perceptual quality is evaluated via LPIPS[29] and DISTS [30]. For audio, distortion is measured via PSNR, SSDR, and SRDR [31], and perceptual quality is evaluated via CDPAM [32]. For both audio and images, the computational cost is measured in terms of average encoding and decoding throughput (megapixels or megasamples per second). Measurements are made on three different platforms: Low-power CPU (Raspberry Pi), High-power CPU (Intel i9), and GPU (RTX 4090).

Results of compression trade-off analysis. Figure 2, Table 1, and Table 2 summarize the trade-offs between rate, distortion, perception, computation, and dimension between different types of compression. For RGB Images, WaLLoC achieves nearly 12× higher compression ratio (35:1 vs 6:1) compared to the VAE used in Stable Diffusion 3, despite offering a higher degree of dimensionality reduction (16× vs 12×) and similar quality (13.9 dB vs 13.8 dB DISTS). Compared to Cheng et al. [11], WaLLoC achieves more than 48× higher encoding throughput (14.0 vs 0.29 MPix/sec) and similar quality (19.3 dB vs 16.9 dB DISTS). For stereo audio, WaLLoC achieves significantly higher spatial quality (22.5 dB vs 15.7 dB SSDR) than Stable Audio's VAE, but with more than 300× higher encoding throughput. Examples of decoded images from the LSDIR validation set are provided on Hugging Face [2]. Additional results, including GPU and Raspberry Pi throughput, are available in our code repository [3].

4.2 Compressed learning and resolution scaling

Next, we describe our methodology for evaluating compressed domain learning.

(a) Applications, models, and datasets. We evaluate WaLLoC on 4 machine perception tasks: (1) image classification, (2) image colorization, (3) document understanding and (4) music source separation. For classification and colorization, we train

[2]Examples of decoded images

[3]Repository containing full code, experiments, and results

Method	DR	CR	Enc	Dec	PSNR	SSDR	SRDR	CDPAM
Opus	1.0	**119**	11.5	**102**	30.4	16.7	5.03	40.4
WaLLoC	4.74	21.3	<u>77.8</u>	11.2	**39.0**	**33.3**	**13.9**	41.1
EnCodec	5.0	<u>114</u>	2.75	3.03	31.9	<u>22.7</u>	6.69	<u>47.4</u>
WaLLoC	<u>18.9</u>	76.3	**121**	<u>12.2</u>	<u>33.3</u>	22.5	<u>8.06</u>	36.6
Stable Audio	**64.0**	64.0	0.308	0.30	28.4	15.7	2.03	**49.7**

Table 2: Stereo audio compression results. Abbreviations are the same as Table 1.

ViT-Ti models with conditional position encoding [33] on the ImageNet-1k dataset. For music source separation, we train a CNN to separate the vocal track from music segments in MUSDB18-HQ. The CNN consists of 12 identical convolutional layers structured identically to Stable Audio's mid block [9]. For document understanding, we use PaliGemma [4] fine-tuned at varying resolution on the DocVQA [34] dataset, and report the average normalized levenshtein similarity (ANLS) on the test set.

(b) Resolution scaling strategy. For **image classification**, we reduce the input sequence length by 4× or 16× compared to the baseline of 256^2 pixels and 16^2 patches, but keep the area of each patch constant ($1/16^2$). We report the accuracy of models trained on reduced resolution inputs with models trained on the identically WaLLoC latents. For **document understanding**, training models on the scale of PaliGemma is outside the scope of this work. Instead, we evaluate on decoded WaLLoC representations using the highest-resolution PaliGemma variant (896^2). To emulate the effect of resolution reduction with this high-resolution variant, we downsample images to the desired resolution, (224^2 or 448^2), then apply Lanczos resampling to interpolate back to 896^2. For **Image colorization** and **music source separation**, we increase the input patch size proportionally to the resolution to keep the sequence length—and therefore the required computation—roughly constant.

Results of compressed-domain learning and resolution scaling. Figure 1 shows the improvement in performance when using WaLLoC-derived representations instead of resolution reduction. Across each of the four tasks, WaLLoC provides superior accuracy to naive resolution reduction while providing the same improvement in latency and memory consumption. For discriminative models, WalloC profoundly increases accuracy of efficient image classification (50.6% vs 23.1% accuracy) and document understanding (81.1 vs 43.7 ANLS). For signal enhancement, WaLLoC provides superior scaling to high resolution and large patches—offering a 16.7% improvement in colorization LPIPS and a 3.1 dB improvement in PSNR for source separation.

5 Conclusion and future work

We introduced WaLLoC, a compression framework to support compressed-domain learning. Our experiments demonstrate that WaLLoC significantly accelerates downstream models without sacrificing accuracy, achieving up to 20× dimensionality re-

duction with minimal encoding cost. Future work will explore extending WaLLoC to applications involving high-resolution signal types for which existing compression methods fall short, such as hyperspectral images or whole-slide microscopy. These domains present additional challenges but also offer greater potential benefits due to increased signal redundancies.

6 Acknowledgments

We thank the anonymous reviewers for their helpful feedback. We thank the members of the UT-SysML research group for their insightful discussions to improve this work. This work was supported by the UT ECE junior faculty start-up fund, UT iMAGiNE consortium and its industrial affiliates, an award from the UT Machine Learning Lab (MLL), the AMD Chair Endowment, the Cisco Research Award, and the Amazon Research Award.

7 References

[1] A. Krizhevsky et al., "Imagenet classification with deep convolutional neural networks," *NeurIPS*, 2012.

[2] S. Hershey et al., "Large-scale audio classification," in *ICASSP*, 2017.

[3] A. Archit et al., "Segment anything for microscopy," *bioRxiv*, 2023.

[4] L. Beyer et al., "Paligemma: A versatile 3b vlm for transfer," *arXiv:2407.07726*, 2024.

[5] M. Ehrlich and L. Davis, "Deep residual learning in the jpeg transform domain," in *ICCV*, 2019.

[6] S. Park et al., "Seit: Storage-efficient vision training with tokens using 1% of pixel storage," in *ICCV*, 2023.

[7] R. Rombach et al., "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.

[8] Johannes et al. Ballé, "End-to-end optimized image compression," in *ICLR*, 2017.

[9] Z. Evans et al., "Stable audio open," *arXiv:2407.14358*, 2024.

[10] M. Goldblum et al., "Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks," *NeurIPS*, 2024.

[11] Z. Cheng et al., "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *CVPR*, 2020.

[12] P. Esser et al., "Scaling rectified flow transformers for high-resolution image synthesis," in *ICML*, 2024.

[13] A. Défossez et al., "High fidelity neural audio compression," *arXiv:2210.13438*, 2022.

[14] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 2008.

[15] J. Ascenso et al., "The jpeg ai standard: Providing efficient human and machine visual data consumption," *IEEE Multimedia*, 2023.

[16] David Minnen and Nick Johnston, "Advancing the rate-distortion-computation frontier for neural image compression," in *ICIP*, 2023.

[17] Yibo Y. Yang and S. Mandt, "Computationally-efficient neural image compression with shallow decoders," in *ICCV*, 2023.

[18] H. Chang et al., "Maskgit: Masked generative image transformer," in *CVPR*, 2022.

[19] J. Copet et al., "Simple and controllable music generation," *NeurIPS*, 2024.

[20] P. Esser et al., "Taming transformers for high-resolution synthesis," in *CVPR*, 2021.

[21] M. Tan and L. Quoc, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *ICML*, 2019.
[22] L. Beyer, "On the speed of ViTs and CNNs," `lb.eyer.be/a/vit-cnn-speed`, 2024.
[23] O. Bohan, "Taesd: Tiny autoencoder for stable diffusion," 2023.
[24] D. He et al., "Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *CVPR*, 2022.
[25] P. Westerink et al., "Subband coding of color images," *Subband Image Coding*, 1991.
[26] Z. Rafii, "The musdb18 corpus for music separation," 2017.
[27] Y. Li et al., "Lsdir: A large scale dataset for image restoration," in *CVPR*, 2023.
[28] Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, 2004.
[29] R. Zhang et al., "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.
[30] K. Ding et al., "Image quality assessment: Unifying structure and texture similarity," *IEEE transactions on pattern analysis and machine intelligence*, 2020.
[31] K. Watcharasupat and A. Lerch, "Quantifying spatial audio quality impairment," in *ICASSP*, 2024.
[32] P. Manocha et al., "Cdpam: Contrastive learning for perceptual audio similarity," in *ICASSP*, 2021.
[33] Z. Tu et al., "Maxvit: Multi-axis vision transformer," in *ECCV*, 2022.
[34] M. Mathew et al., "Docvqa: A dataset for vqa on document images," in *IEEE/CVF winter conference on applications of computer vision*, 2021.

Appendix

Figure 5: Cheng et al. 2020 [11]

Figure 6: Stable Diffusion 3 VAE [12]

Figure 7: WaLLoC 4×

Figure 8: WaLLoC 16×

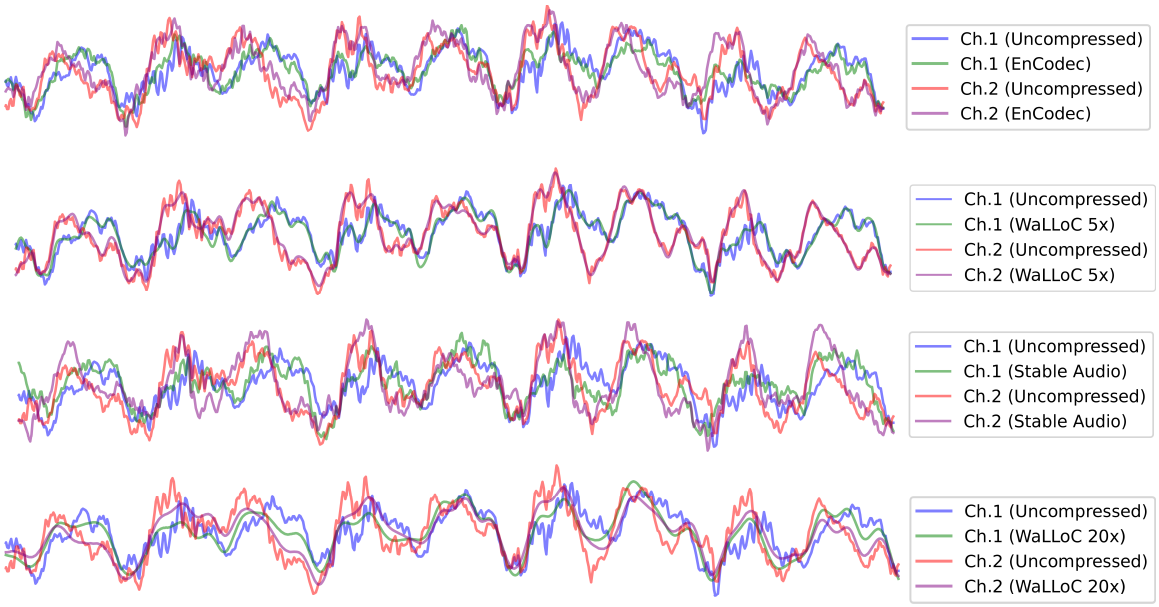

Figure 9: Stereo reconstruction of an audio segment from the MUSDB test set.

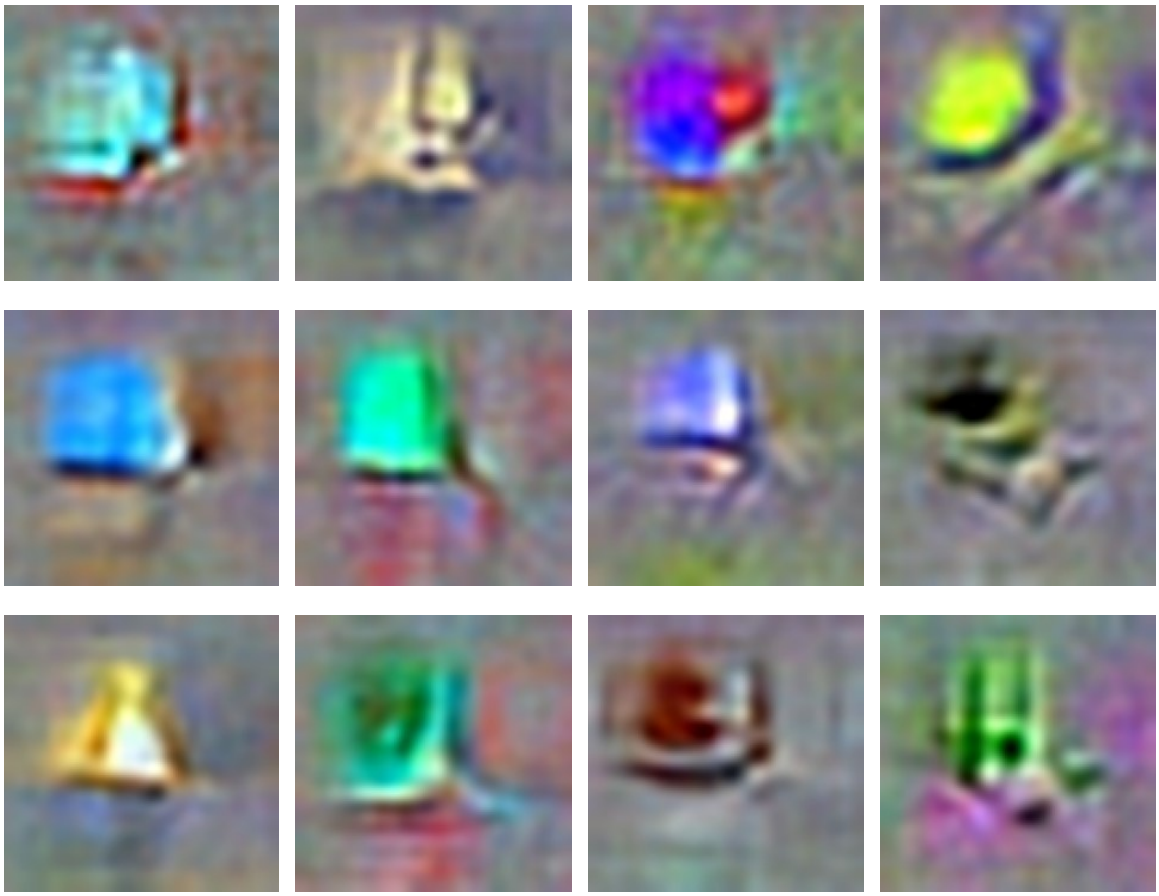

Figure 10: Result of using the $C_{\mathbf{z}} = 12$ RGB codec (WaLLoC 16×) to decode a $12 \times 3 \times 3$ latent with all elements equal to zero except except for channel i, which is set to $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 31 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

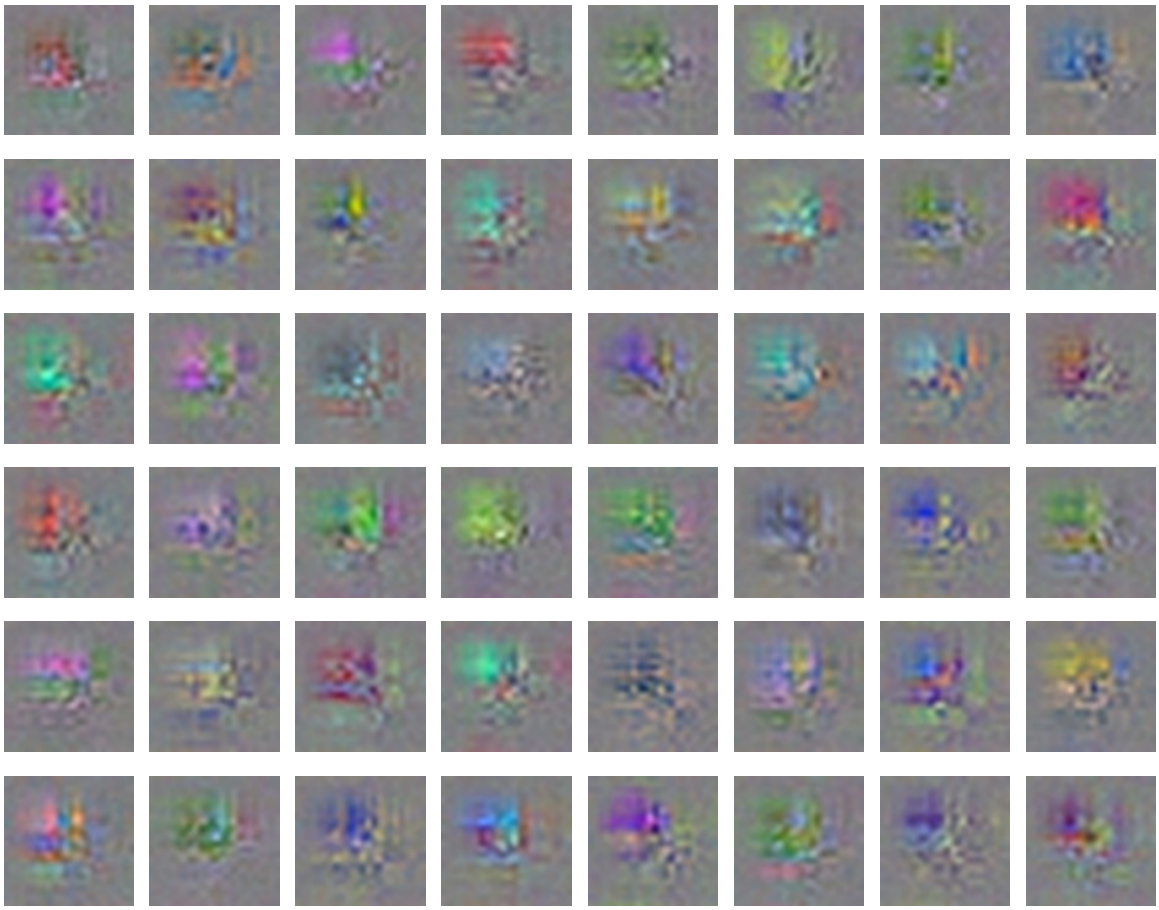

Figure 11: Result of using the $C_{\mathbf{z}} = 48$ RGB codec (WaLLoC 4×) to decode a $48 \times 3 \times 3$ latent with all elements equal to zero except except for channel i, which is set to $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 31 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

Task	Resolution Equivalent	WaLLoC Variant	Performance (Resize)	Performance (Compress)	Change
Classification (Acc., %)	64^2 px	16×	23.1	50.3	↑27.2
	128^2 px	4×	55.8	64.3	↑8.5
	256^2 px	–	71.1	–	–
Doc. VQA (ANLS)	224^2 px	16×	43.7	81.1	↑37.4
	448^2 px	4×	78.0	84.1	↑6.1
	896^2 px	–	84.8	–	–
Colorization (LPIPS, dB)	128^2 px	–	1.76	–	–
	256^2 px	4×	2.33	2.47	↑0.14
	512^2 px	16×	2.43	2.83	↑0.40
Source sep. (PSNR, dB)	2.4 kHz	–	31.1	–	–
	11 kHz	5×	32.0	34.4	↑2.4
	44 kHz	18×	31.8	34.2	↑2.4

Table 3: Results of resolution scaling experiments.