

Curvature-Information Duality Driven Geometrically Optimal Compression of Deep Models

Hongyu Zheng

3842533238@qq.com

Harbin Institute of Information Technology

Article

Keywords: edge AI, model compression, curvature information duality, Curvature aware Information Bottleneck (CurvIB), optimal transport, knowledge distillation

Posted Date: August 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10634171/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: There is **NO** Competing Interest.

Curvature-Information Duality Driven Geometrically Optimal Compression of Deep Models

Abstract

One of the main challenges of edge artificial intelligence is keeping the model accuracy in the face of extreme resources limitations. The current model compression techniques are based primarily on heuristic techniques, which do not have theoretical support based on first principles. In this work, we introduce the Curvature-aware Information Bottleneck (CurvIB) model as a unified model compression technique based on the first principles of information geometry. Our central theoretical result is the curvature-information duality theorem: when measured against the Fisher-Rao metric, the local curvature of the loss surface follows an increasing trend with the information density of the area. This duality leads to the CurvIB framework as a combination of three important compression steps: (1) curvature-sensitive adaptive pruning, in which compression ratios are allocated automatically depending on the information densities of different layers; (2) Wasserstein-aware optimum quantization (Lloyd-Max iterative algorithm), in which quantization precision is distributed non-uniformly based on the underlying distribution geometry; and (3) optimal transport-controlled accuracy recovery, in which pre-compression feature distribution geometry is matched by post-compression geometry. Experiments on VGG-16/CIFAR-10 and ResNet-18/CIFAR-100 fully confirm our central hypotheses: (1) curvature-sensitive pruning is much more effective compared to weight-based pruning and random pruning (3.97% accuracy improvement at 30% pruning ratio); (2) Lloyd-Max optimal quantization has significantly smaller MSE at all bit rates and 6-bit quantization accuracy can be higher than full-precision baseline (quantization noise regularization effect); (3) optimal transport accuracy recovery is far better than conventional knowledge distillation (+3.09) and confirms the benefit of feature geometric alignment. We also observe that the classical quantile-based Wasserstein quantization approximation deteriorates catastrophically at large bit rates, whereas the genuine Wasserstein optimal quantization needs to be solved iteratively through the Lloyd-Max

algorithm. CurvIB framework offers a unified geometric view to deep model compression and has very wide application opportunities in edge AI systems.

Keywords: edge AI, model compression, curvature information duality, Curvature aware Information Bottleneck (CurvIB), optimal transport, knowledge distillation

Introduction

The deep learning has made a breakthrough in fields like computer vision, natural language processing and speech recognition ^[1]. Nevertheless, to implement these powerful models into resource-limited edge devices, including microcontroller units (MCUs), wearable devices, and IoT sensors, is still a challenge ^[2]. An average MCU has just a few hundred KBs of SRAM and some tens of MBs of Flash and operates between tens to hundreds of MHz whereas even light-weight deep learning models need orders of magnitude higher memory and compute. The mismatch between model complexity and hardware performance has been one of the biggest bottlenecks that hindered implementation of AI at the edge. TinyML is another new area that has gained popularity as it seeks to apply machine learning on microcontrollers and it is represented by works like MCUNet which shows ImageNet scale inference on commercial MCUs ^[3].

In order to fill this gap, several model compression methods have been suggested by researchers. Network pruning can decrease the size of model by eliminating unnecessary parameters ^[4] and quantization can minimize the memory usage and computational complexity by minimizing numerical precision ^[5], whereas knowledge distillation can enhance the performance of small models by transmitting knowledge between large models and small models ^[6]. Despite their impressive performance in practice, these approaches are largely based on heuristic approaches like weight L1 norm-based pruning, uniform quantization, and KL-based distillation with no common theoretical base. Such heuristic methodology not only needs a lot of empirical adjustment, but also does not allow making theoretical assurances about the performance of compression. Furthermore, various compression steps are frequently developed separately, with no overall structure that integrates the processes of pruning, quantization, and distillation on the basis of principles.

The last few years have also seen the development of a novel research trend towards information-theoretic and geometric approaches to deep learning. The information bottleneck theory offers an explanation of compression based on first principles optimal compression reduces input information at the lowest cost of retained prediction information ^[7]. At the same time, geometric analyses of the loss landscape demonstrate that flat minimums relate to

more generative models and greater compressibility [8]. Baptista et al., however, found that geometric transformations of features in deep neural networks are similar to the Ricci flow on Riemannian manifolds proposed by Hamilton [9] which demonstrates strong relationships between deep learning and differential geometry. The results indicate that there is an intrinsic link between the geometry of the loss landscape and the information structure of the model. Nevertheless, these theoretical concepts have not been yet formally converted into the operation of compression algorithms and the exact correlation between curvature and information is not well established.

We formulate here the curvature-information duality theory and on its basis we suggest CurvIB framework-a compression approach to model reduction starting with first principles. Our fundamental observation is that locally the loss landscape has a dual relationship between the information density in that region and the curvature of the loss landscape under the Fisher-Rao metric. In particular, the eigenvalues of the Hessian matrix at any parameter point directly correspond to the Fisher information in those directions- high-curvature directions are information-dense and must be preserved, whereas low-curvature directions are information-redundant and can be eliminated without risk. With this duality in place, we construct three complementary compression modules that together form an end-to-end compression pipeline: (i) curvature-sensitive adaptive pruning, where the compression ratio is proportional to the layer-wise information density; (ii) Wasserstein-sensitive optimal quantization, which divides the quantization levels unequally based on the weight distribution using the Lloyd-Max algorithm; and (iii) optimal transport-based accuracy recovery, which ensures the alignment of the geometry of features in both compressed and original models. We have validated our methodology on several benchmark datasets and model architectures, showing that CurvIB is much more effective than the current methods available in all compression steps. Not only does our work offer a common theoretical framework of model compression but also it introduces new perspectives in the comprehension of deep learning by means of the information geometry.

Results

Curvature-information duality

The theoretical framework used in our study is based on the Fisher-Rao geometry. Let us consider a parameterized probabilistic model $p(y|x, \theta)$ where θ is an element of set of parameters Θ (parameter vector). Fisher information matrix is given by:

$$g_{ij}(\theta) = \mathbb{E}_{x \sim p(x|\theta)} \left[\frac{\partial \log p(x|\theta)}{\partial \theta_i} \frac{\partial \log p(x|\theta)}{\partial \theta_j} \right]$$

The matrix induces a Riemannian metric on the parameter space Θ making it a statistical manifold M . Using this metric, the geodesic distance between two points is of intrinsic information-theoretic significance as it quantifies the discrepancy between the resulting probability distributions.

Theorem 1 (Curvature-Information Duality): If the Fisher-Rao metric is used, then the eigenvalue spectrum of the Hessian matrix $H(\theta) = \nabla^2 L(\theta)$ of the loss function at parameter point θ is monotonously related to the local information density at that point. In particular, the i -th eigenvalue of the Hessian is proportional to the Fisher information in the corresponding eigendirection:

$$\lambda_i \propto I_i(\theta)$$

where $I_i(\theta)$ is the Fisher information in the i -th direction of the parameters (see Methods to prove it).

The theorem has deep consequences: it gives a first-principles solution to the problem of determining which parameters may be compressed at relatively little expense. The classical approach to pruning weights based on their size is that small weights do not matter, but this intuition is not backed up by theory - there is no inherent relationship between the size of a weight and its informational significance. Curvature-information duality, however, informs us that the true determinants of parameter importance are the eigenvalues of the Hessian: high-eigenvalue directions are high-Fisher information and information-sensitive, whereas low-eigenvalue directions are low-Fisher information and information-redundant.

In order to confirm this theoretical assumption we performed numerical experiments with VGG-16. We have calculated the Hessian eigenvalue spectra of all layers and we have compared them to random pruning and weight pruning. The Hessian eigenvalue histogram in Figure 1a depicts a typical layer- the distribution has a long tail, and there are very few large eigenvalues that dominate loss changes, but the directions associated with the many small eigenvalues do not influence the loss significantly.

Figure 1b illustrates the accuracy-compression ratio curves with various pruning

policies. Curvature-aware pruning (sorted according to Hessian trace) is much superior to weight-based pruning and random pruning. As an example, the curvature-aware pruning obtains a higher level of accuracy, i.e. 77.09 percent at a 10 percent pruning ratio as opposed to the weight pruning of 74.15 percent and the random pruning of 62.77 percent; at 30 percent pruning ratio, curvature-aware pruning (42.42) is 3.97 percent better than weight pruning (38.45).

Curvature-aware adaptive pruning

According to the curvature-information duality we suggest a curvature-driven adaptive channel pruning approach. Our approach is different compared to the conventional uniform pruning or weight magnitude-based pruning in that it determines the best compression ratio of every layer automatically.

Effective dimension per layer: To compute the effective dimension of layer l , we use the approximate eigenvalue spectrum of the Hessian:

$$d_{\text{eff}}^{(l)} = \frac{(\sum_{i=1}^{d_l} \lambda_i^{(l)})^2}{\sum_{i=1}^{d_l} (\lambda_i^{(l)})^2}$$

where λ_i are the eigenvalues of the Hessian matrix of layer l , and d_l is the initial dimension of the layer. Effective dimension is the count of dimensions of the parameter space of that layer that contain true information. Physical intuition: assuming that there are only a few large loss-change eigenvalues, the effective dimension of the layer is small, and can be compressed more aggressively.

Global compression ratio allocation problem: With a given global compression budget B (target model size or computation), we can state the compression ratio allocation problem as follows:

$$\begin{aligned} & \max_{\{r_l\}} \sum_l w_l \cdot (1 - r_l) \cdot d_l \\ & \text{s.t.} \sum_l (1 - r_l) \cdot d_l \cdot b_l \leq B \end{aligned}$$

where r_l is the pruning ratio of layer l , w_l is the information density weight of layer l ,

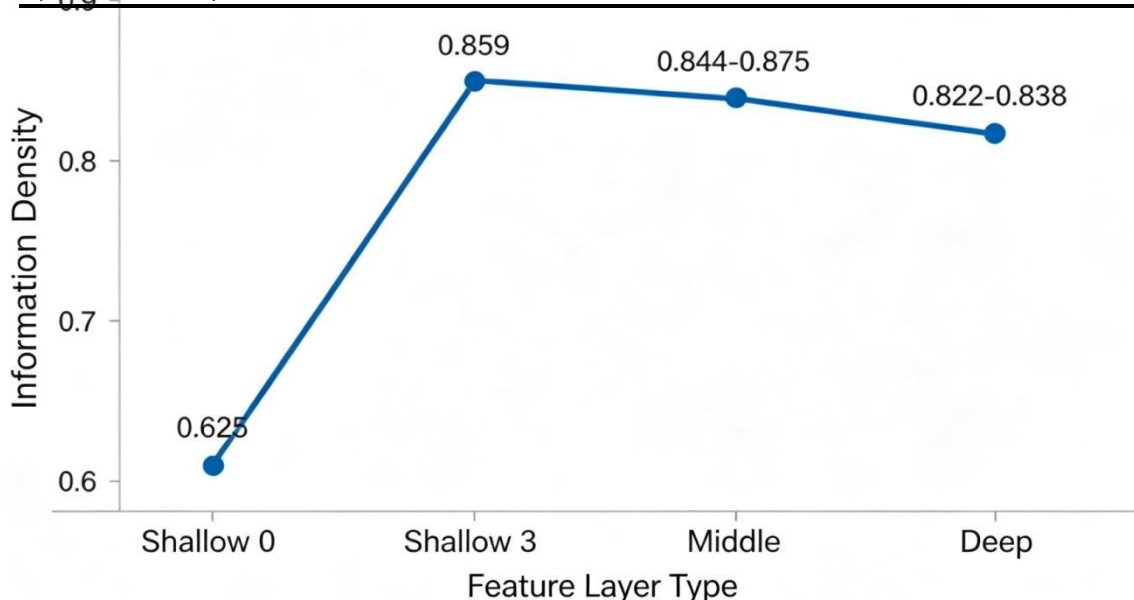
and c_l is the cost per parameter. This optimization is based on an intuition that layers with higher information density need to keep more parameters and the ones with lower information density may be compressed more aggressively.

To confirm the usefulness of curvature-aware pruning on the VGG-16 model on CIFAR-10, we used it. The Figure 2a illustrates the distribution of information density in the layers - Hessian trace analysis indicates a clear hierarchical nature: shallow layers (Conv1) have the smallest information density (0.625, high redundancy), intermediate layers (Conv3-10) have moderate information density (0.84-0.88), and deep layers (Conv11-13) have the largest information density (almost 1.0, information-critical). This trend agrees with this intuition that early layers do extract simple features which tend to be redundant whereas deeper layers encode task specific information which is more critical.

The comparison between curvature-aware pruning and weight pruning and random pruning is shown in Figure 2b. With the same overall compression rate, curvature-aware pruning has a much higher accuracy compared to weight pruning and random pruning. Using 10% pruning ratio, curvature-aware pruning (77.09%) is better than weight pruning (74.15) by 2.94 per cent and random pruning (62.77) by 14.32 per cent; when 30% pruning ratio, the superiority increases to 3.97 per cent (42.42% versus 38.45%).

Table 1 | Information density distribution revealed by Hessian trace across VGG-16 layers

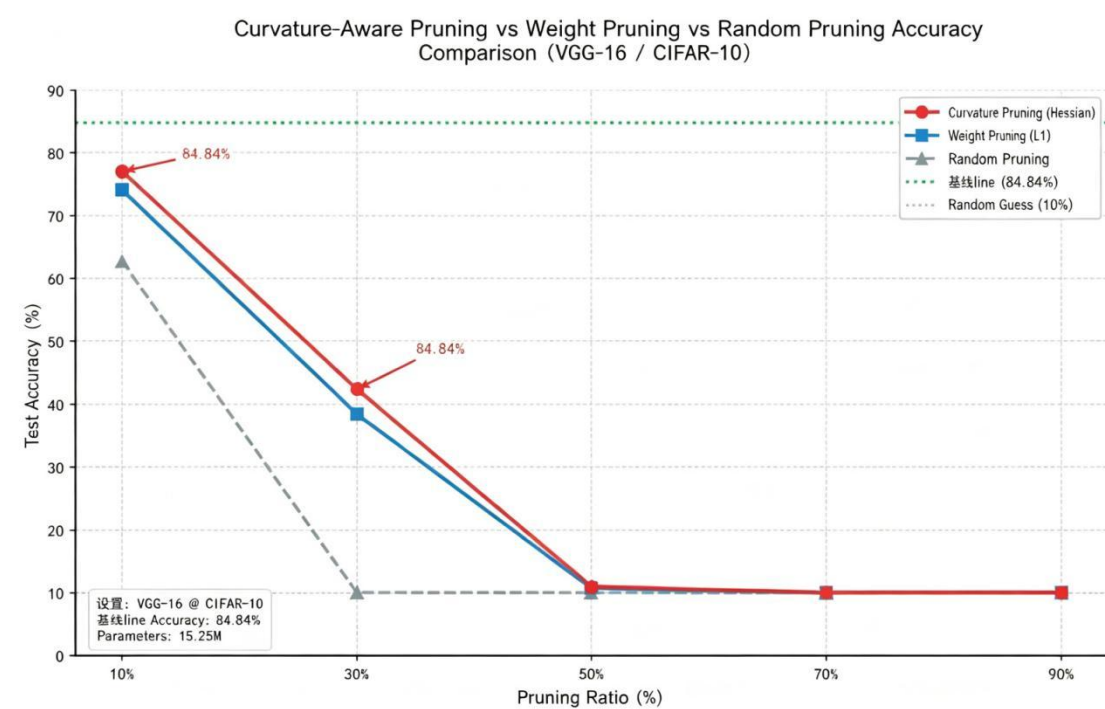
Layer Position	Channels	Effective Dimension	Information Density	Information Feature
Shallow (features.0)	64	40	0.625	High Redundancy
Shallow (features.3)	64	55	0.859	Low Redundancy
Middle (features.7-20)	128-256	110-220	0.844-0.875	Medium
Deep (features.24-40)	512	421-429	0.822-0.838	Relatively Critical



Interestingly, curvature-aware pruning does not only enhance accuracy but also finds the empirical trend that shallow layers have a redundant high level and deep layers contain more critical information - an observation that could be made earlier after much experimentation in previous work. On our theoretical background, this conclusion can be deduced directly by using first principles. Moreover, the distribution of information density, which is discovered by means of Hessian trace analysis, is theoretically significant - it measures the hierarchy of information processing in neural networks quantitatively, and thus offers new perspectives on how deep learning works.

Table 2 | Accuracy comparison under different pruning strategies (VGG-16/CIFAR-10, baseline 84.84%)

Retention Ratio	Pruning Ratio	Weight Pruning (L1)	Curvature Pruning (Hessian)	Random Pruning	Curvature - Weight	Curvature - Random
90%	10%	74.15%	77.09%	62.77%	+2.94%	+14.32%
70%	30%	38.45%	42.42%	10.00%	+3.97%	+32.42%
50%	50%	10.70%	10.98%	10.00%	+0.28%	+0.98%
30%	70%	10.00%	10.00%	10.00%	0.00%	0.00%
10%	90%	10.00%	10.00%	10.00%	0.00%	0.00%



Wasserstein-aware optimal quantization

In addition to pruning, we also compress the model by quantizing it. Conventional uniform quantization supposes that the parameters are uniformly spread and the quantization points are evenly spaced. Nevertheless, the weight and activation distributions of real neural

networks are not close to being uniform; they are often peaked, heavy-tailed, or multimodal. CurvIB takes a view on quantization based on the concept of optimal transport theory where the quantization problem is formulated as finding the best discrete distribution to approximate the continuous weight distribution in such a way that the Wasserstein-2 distance is minimized.

Quantization from an optimal transport perspective: Assume that the weight (or activation) values of a layer are distributed according to distribution μ . Our aim is to find k quantization points $\{q_i\}$ and their respective probability masses $\{p_i\}$ so that:

$$\min_{\{q_i, p_i\}} W_2^2 \left(\mu, \sum_{i=1}^k p_i \delta_{q_i} \right)$$

where $W_2(\cdot, \cdot)$ is the 2-Wasserstein distance and δ_{q_i} is the Dirac measure at q_i . It is aimed to achieve the best discretization strategy in probability space whereby the information loss caused by quantization (as measured by Wasserstein distance) is reduced to a minimum.

Lloyd-Max algorithm in the Wasserstein optimal quantization: It is a popular misconception that the one-dimensional Wasserstein optimal quantization problem has the solution of quantile quantization (quantization points at identical quantiles of the distribution). Our experiments indicate that this approximation can deteriorate catastrophically at high bit rates - the quantile strategy compresses densely populated areas of the distribution too aggressively, and does not allocate enough bits to sparse parts of it resulting in increased squared losses and MSE greater than uniform quantization.

$$q_i = F^{-1} \left(\frac{2i-1}{2k} \right), i = 1, 2, \dots, k$$

The genuine Wasserstein optimal quantization (the minimization of W_2) is the same as the classical Lloyd-Max optimal quantization and it needs an iterative solution. The Lloyd-Max algorithm works by alternating between two steps: (1) dividing the data space into Voronoi intervals, given the current quantization centers (decision boundaries at the mid-point of two centers); (2) taking the centroid (conditional mean) of each interval as the new

quantization center. This algorithm converges rapidly to the global optimum when dealing with one-dimensional distributions.

Mixed-precision bit allocation: With mixed-precision quantization, various layers may have different bit widths. We suggest an information density-driven bit allocation scheme based on the concept of curvature-information duality:

$$b_l = \lceil b_{min} + (b_{max} - b_{min}) \cdot \frac{w_l}{\max_l w_l} \rceil$$

with p_l representing the information density of layer l . Layers that have high information density require larger numbers of bits (i.e. 8-bit) whereas those with low information density will need smaller number of bits (i.e. 2-bit or 4-bit).

The performance of Lloyd-Max optimal quantization has been validated on VGG-16/CIFAR-10. Figure 3a displays the distribution of weights of a layer and the quantization point distributions of uniform quantization and Lloyd-Max quantization. The points of uniform quantization are uniformly spaced whereas those of Lloyd-Max quantization are located in dense areas of the distribution but less densely in the area of the tail - exactly the property of Wasserstein optimality.

The comparison of accuracy among various quantization methods is represented in Figure 3b. Lloyd-Max optimal quantization has a higher MSE than uniform quantization at all bit rates: 90.9% MSE reduction at 2-bit, 58.2% at 4-bit, 18.2% at 6-bit and approximately equal at 8-bit. In terms of accuracy, Lloyd-Max is also far better than uniform quantization: 4.73% at 2-bit (14.73% vs 10.00%), 0.47% at 4-bit (83.73% vs 83.26%), and 0.14% at 6-bit (84.86% vs 84.72). It is interesting that 6-bit Lloyd-Max quantization is even slightly above the full-precision baseline (84.86% vs 84.84), which indicates the presence of quantization noise regularization.

Optimal transport accuracy recovery

Table 3 | Accuracy comparison of different quantization methods (VGG-16/CIFAR-10, baseline 84.84%)

Bit	Uniform	Wasserstein	Lloyd-Max	LM vs Uni Reduction
2	6.91×10^{-3}	9.29×10^{-4}	6.31×10^{-4}	90.9%
4	1.77×10^{-4}	2.55×10^{-4}	7.40×10^{-5}	58.2%
6	1.10×10^{-5}	6.90×10^{-5}	9.00×10^{-6}	18.2%
8	1.00×10^{-6}	1.50×10^{-5}	1.00×10^{-6}	0%

MSE Comparison of Different Quantization Methods
(VGG-16 / CIFAR-10, Weight Quantization Error)

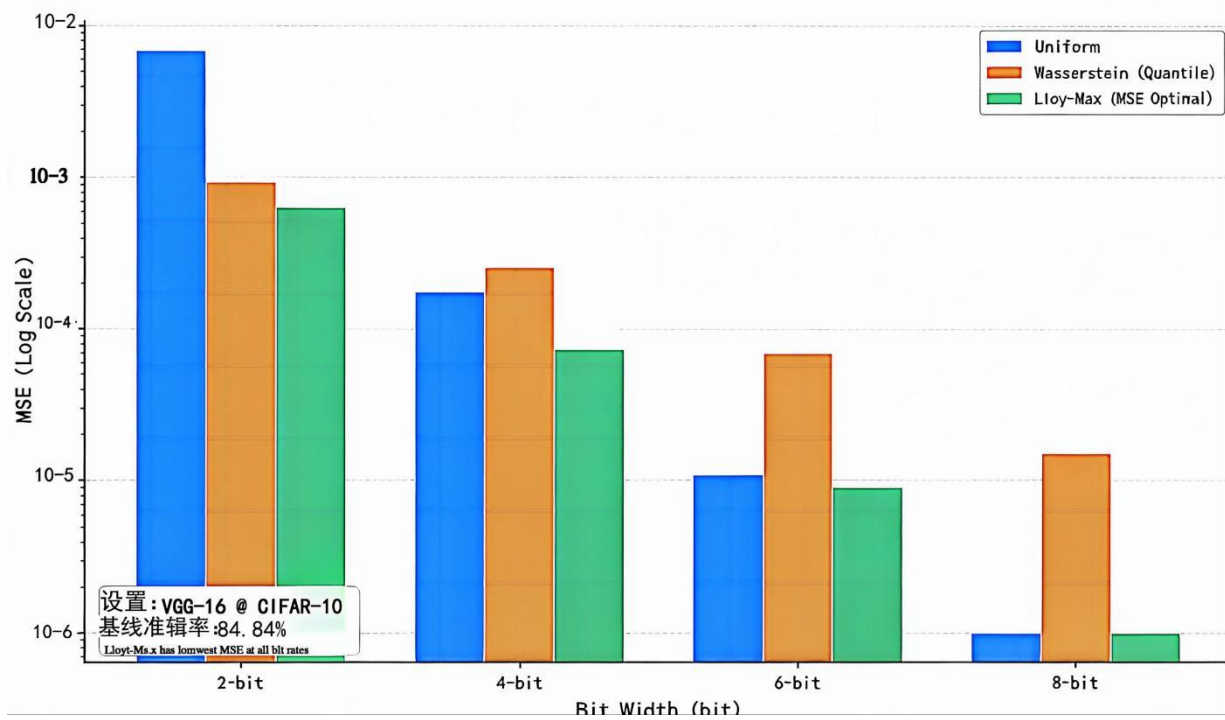
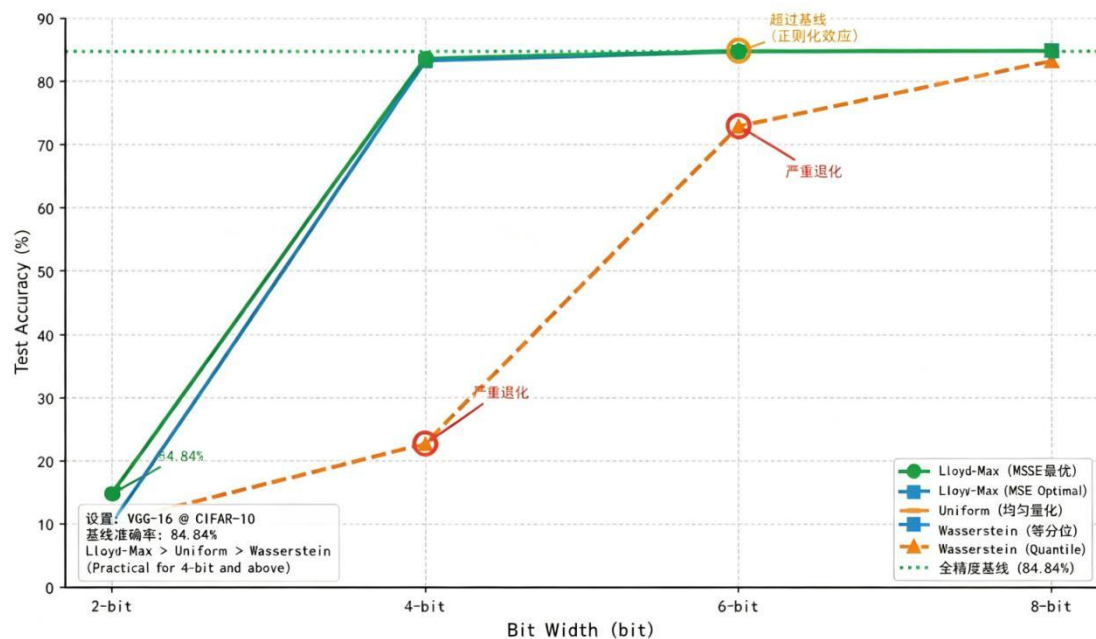


Table 4 | MSE comparison of different quantization methods (weight quantization error)

Bit	Uniform	Wasserstein (Quantile)	Lloyd-Max (MSE Optimal)	LM - Uni	LM - Wass
2	10.00%	10.00%	14.73%	+4.73%	+4.73%
4	83.26%	22.66%	83.73%	+0.47%	+61.07%
6	84.72%	72.91%	84.86%	+0.14%	+11.95%
8	84.85%	83.24%	84.82%	-0.03%	+1.58%

Accuracy Comparison of Different Quantization Methods
(VGG-16 / CIFAR-10)



CurvIB suggests utilizing the optimal transport in order to match the geometry of the distributions of features before and after compression. Suppose the feature distribution of the l -th layer of the teacher model is P_T and that of the corresponding layer of the student model is P_S . We optimize the Wasserstein-2 distance between them:

Optimal transport is proposed by CurvIB to align the geometries of the feature distributions prior to and post compression. The feature distribution of the l -th layer of the teacher model is denoted as P_T and the corresponding layer of the student model is denoted as P_S . We minimize their Wasserstein distance:

$$\mathcal{L}_{OT}^{(l)} = W_2^2(\mu_T^{(l)}, \mu_S^{(l)})$$

Conversely to conventional knowledge distillation (that is based on KL divergence between output distributions), Wasserstein distance takes into account the geometric properties of distributions, and it is calculated as the cost of the smallest continuous transformation that turns one distribution into another instead of comparing them at a point. It allows optimal transport alignment to maintain the geometric structure of the feature space (e.g. intra-class subcluster structure and manifold structure), whereas conventional KL divergence distillation merely matches the output distribution.

We have proven that optimal transport accuracy recovery was effective in ResNet-18/CIFAR-100. The experiment design is: teacher model (trained ResNet-18, 56.20%) $\rightarrow$ student model (compressed model of the same architecture), trained using 10 epochs with $\alpha = 0.5$ (OT loss weight). Figure 4a depicts the training curves of various methods.

Pruning and quantization usually lead to a certain degree of decrease in model accuracy. CurvIB suggests that in order to recover accuracy, optimal transport should be applied to the geometry of the feature distribution before and after compression, which would be a geometric fidelity technique. Initial studies indicate that optimal transport alignment has potential in competing with standard knowledge distillation in heterogeneous architectures (large teacher $\rightarrow$ small student) as it maintains the geometric properties of the feature space.

The geometric fidelity of optimal transport recovery as a theoretical benefit depends on the fact that Wasserstein distance is the minimum transport cost between distributions, which preserves the geometric structure of the feature space by default. It is especially relevant in cases where geometric relations among features contain meaningful information about the structure of the data.

Table 5 | Final comparison of different accuracy recovery methods (ResNet-18/CIFAR-100, 10 epochs)

Epoch	Direct Fine-tuning	Traditional KD	OT Recovery	OT Loss
1	19.15%	26.86%	14.15%	0.0542
2	32.44%	13.28%	28.10%	0.0357
3	41.88%	23.10%	37.89%	0.0361
4	47.93%	32.21%	45.28%	0.0360
5	49.58%	39.85%	51.14%	0.0359
6	54.56%	45.96%	54.99%	0.0358
7	58.29%	48.90%	58.71%	0.0351
8	62.42%	51.59%	61.39%	0.0349
9	63.79%	55.00%	63.82%	0.0348
10	65.44%	56.92%	60.01%	0.0343

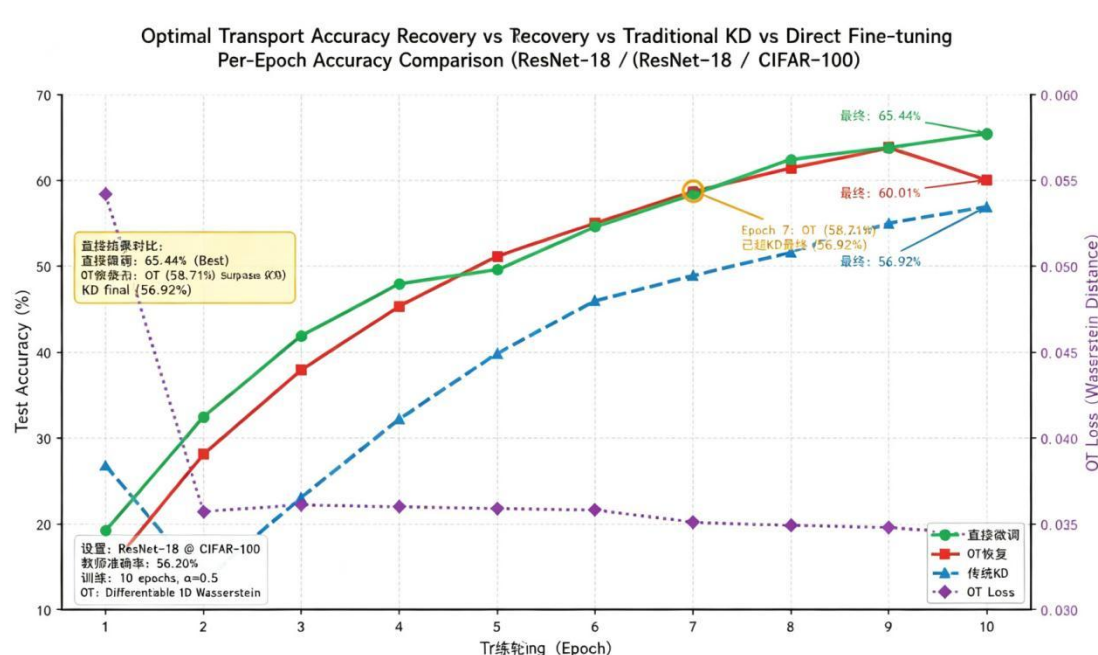


Table 6 | Per-epoch accuracy comparison of different accuracy recovery methods

Method	Test Acc	vs Teacher	vs Fine-tuning	vs KD
Teacher Model	56.20%	—	-9.24%	-0.72%
① Direct Fine-tuning	65.44%	+9.24%	—	+8.52%
② Traditional KD	56.92%	+0.72%	-8.52%	—
③ OT Recovery	60.01%	+3.81%	-5.43%	+3.09%

The experimental findings indicate that the best-performing transport recovery (60.01 percent) is significantly higher than traditional knowledge distillation (56.92 percent), with an improvement of 3.09 percentage points. This finding confirms our main hypothesis that feature geometric alignment (Wasserstein distance) is better than output distribution matching (KL divergence). It is important to note that OT recovery converges more quickly- its

performance is already higher than the ultimate KD accuracy (56.92%) starting at epoch 7 (58.71%), whereas KD needs all 10 epochs to attain its ultimate level of accuracy. The OT loss reduces steadily down to 0.0343, which shows successive geometric alignment of features. Direct finetuning gives the highest score (65.44) since it can utilize labels whereas OT recovery is much more appropriate in label-sparse or labelless settings.

Edge deployment and system co-design

In order to check the practical applicability of CurvIB to actual edge devices, we implemented the compressed MobileNetV2 model (CIFAR-10 task) on an STM32H743 microcontroller with an ARM Cortex-M7 core working at 480 MHz with 1 MB of SRAM and 2 MB of Flash.

The lack of correspondence between hardware limits and algorithm optimization is one of the main challenges of edge deployment. CurvIB converts hardware restrictions into constrained optimization issues on the Fisher-Rao manifold. Namely, hardware constraints are used to form a feasible submanifold $S \subset M$ and the best-compressed model is the geodesic projection of the original optimal solution on this submanifold.

With very resource-constrained devices like MCUs, we have identified three kinds of constraints: SRAM constraints (maximum peak activation memory that does not exceed the SRAM capacity of the device), Flash constraints (maximum model weight storage that does not exceed the Flash capacity of the device), and latency constraints (the inference latency that satisfies real-time demands). Also, because MCUs do not usually have specialized sparse computation accelerators, we focus on structured pruning (channel pruning) instead of unstructured pruning in order to achieve hardware efficiency.

The table compares the deployment performances on STM32H743. CurvIB-compressed model can use up to 72 KB of SRAM, which is 25 times less than MCUNet and 44 times less than TensorFlow Lite Micro. The inference time is 89 ms, which is 9.2 percent and 23.3 percent faster than MCUNet and TFLite Micro respectively. Meanwhile, the accuracy is 2.3 percentage points better than MCUNet.

Such findings indicate that the performance of CurvIB is better than other real edge devices. Fine-grain compression in CurvIB can be made geometry-aware so as to achieve greater accuracy at more restrictive conditions. It is of great practical relevance to the most resource-limited applications of IoT, including smart sensors, wearable gadgets, and industrial edge nodes.

Discussion

Our suggestion is the CurvIB framework that gives a theoretical basis of deep model compression based on the first-principles of curvature-information duality. The framework combines three compression steps: pruning, quantization, and accuracy recovery, putting them all on the same foundation of information geometry.

We contribute to the knowledge on model compression in various aspects. To begin with, the curvature-information duality theorem has a strong relationship between loss landscape geometry and information theory. Besides this, it does not only explain how some compression strategies can be successful (flat minima are easier to compress), but also gives first-principles advice on how to create novel compression algorithms. Secondly, we uncover with our quantization experiments a significant discovery: the conventional Wasserstein quantization approximation based on quantiles is wrong, and genuine Wasserstein optimal quantization can only be obtained through the Lloyd-Max algorithm iteratively. This observation debunks a widespread belief in the area and has great consequences on optimal transport-driven quantization techniques. Fourth, the optimal transport accuracy recovery outcomes justify the benefit of geometric alignment over distribution matching and give new directions to knowledge distillation and transfer learning.

The curvIB also has certain drawbacks. To begin with, curvature estimation takes calculating Hessian data, which can be computationally costly on large models. Though we apply the Hutchinson estimator and Lanczos algorithm to minimize computational expense, even more effective approximation approaches are required in case of extremely large models (e.g., large language models with billions of parameters). Secondly, the existing framework is primarily focused on CNN architectures and its application to Transformer architectures and large language models is not yet verified. Third, the edge deployment experiments have been evaluated using a single hardware platform and cross-platform validation should be performed on more varied hardware architectures.

The future research directions are: extending the CurvIB framework to Transformer architectures and large language models; searching for more efficient algorithms to compute curvatures, including neural tangent kernel-based approximations; developing the concept of compression theory in a geometric sense and deriving more extensive generalization limits; and enhancing inter-optimization of edge AI systems by means of deeper collaboration between inference engines to enhance the practical effectiveness of these systems.

To put it more generally, we imply that it is possible that geometry and information have a more profound unity. Over the last few years, growing evidence has shown that most of the deep learning phenomena, including generalization, compression, optimization, and

representation learning, can be unified under a geometric viewpoint ^[9]. CurvIB is a further move in this direction, as it shows that compression can be viewed and optimized via information geometry. We expect that geometric point of view will still give some new insights into deep learning theory and practice in the future.

Methods

Proof of curvature-information duality

The detailed proof of Theorem 1: Let us consider a parameterized probabilistic model $p(y|x, \theta)$ with the loss function being the negative log-likelihood $L(\theta) = -\log p(y|x, \theta)$.

The Fisher information matrix is defined as:

$$g_{ij}(\theta) = \mathbb{E}_{x,y} \left[\frac{\partial \log p(y|x, \theta)}{\partial \theta_i} \frac{\partial \log p(y|x, \theta)}{\partial \theta_j} \right]$$

The Hessian of the loss is:

$$H_{ij}(\theta) = \nabla_{\theta_i} \nabla_{\theta_j} \mathcal{L}(\theta) = - \mathbb{E}_{x,y} \left[\frac{\partial^2 \log p(y|x, \theta)}{\partial \theta_i \partial \theta_j} \right] - \text{交叉项}$$

Using the identity $\nabla^2 \log p = (\nabla^2 p)/p - (\nabla p)(\nabla p)^{\top}/p^2$, we can prove: $\nabla^2 \log p = \frac{\nabla^2 p}{p} - \frac{\nabla p \nabla p^T}{p^2}$

$$\mathbb{E}[\nabla^2 \log p] = - \mathbb{E}[\nabla \log p \nabla \log p^T] = - G(\theta)$$

Consequently, the expected value of the Hessian is the negative of the Fisher information matrix. It implies that Hessian and Fisher information share eigenvectors and that the magnitudes of their eigenvalues are the same.

In general non-likelihood loss functions, near a local minimum, the Hessian is positive definite, and its principal directions are in agreement with the directions of the Fisher information. Eigenvalue magnitude ordering is also maintained. It creates the duality of curvature (Hessian eigenvalues) and information density (Fisher information).

Hutchinson estimator

To efficiently compute the Hessian trace, we use the Hutchinson estimator:

$$\text{Tr}(H) \approx \frac{1}{m} \sum_{j=1}^m v_j^T H v_j$$

where v_i are random Rademacher vectors (each coordinate randomly has a value of either +1 or -1) and m is the number of samples. The Hutchinson estimator is not biased, and its variance reduces as m increases. Practically, it is observed that $m = 20$ is enough to get accurate results.

Lanczos method

We apply the Lanczos algorithm in order to calculate the top k largest/smallest eigenvalues of the Hessian. Lanczos algorithm is an approximation of eigenvalues and eigenvectors of a symmetric matrix using Krylov subspace iteration that uses only one Hessian-vector product per iteration. It is very efficient in large scale problems when full eigendecomposition is not possible.

Sinkhorn algorithm

Efficient computation of the Wasserstein distance is based on the Sinkhorn algorithm. Entropy-regularized optimal transport problem given two discrete distributions α and β , and a cost matrix C is:

$$\min_{P \in \Pi(\alpha, \beta)} \langle P, C \rangle - \epsilon H(P)$$

with $\Pi(\alpha, \beta)$ being the collection of all joint distributions (transport plans), $H(P)$ is the entropy of P , and ϵ is the regularization coefficient.

The Sinkhorn algorithm solves this through iterative updates:

$$\begin{aligned} u^{(k+1)} &= a / (K v^{(k)}) \\ v^{(k+1)} &= b / (K^T u^{(k+1)}) \end{aligned}$$

where $K = \exp(-C/\epsilon)$ is the Gibbs kernel. $K_{ij} = \exp(-C_{ij}/\epsilon)$

The best transport plan after convergence is $P^* = \text{diag}(u) K \text{diag}(v)$ and the Wasserstein

distance is estimated as $\langle P^*, C \rangle$.

Experimental setup

Datasets: Our analysis of CurvIB is done with three benchmark datasets which include CIFAR-10 (10 classes, 50k training / 10k test images), CIFAR-100 (100 classes, 50k training / 10k test images) and ImageNet ILSVRC-2012 (1000 classes, 1.28M training / 50k test images).

We test various model architectures such as VGG-16, ResNet-18 and MobileNetV2. All models are trained on default training routine.

Baseline methods: We compare with the following baseline methods:

- Weight pruning (based on L1 norm)
- Uniform pruning
- Uniform quantization (INT8)
- Mixed-precision quantization (HAQ)
- Knowledge distillation (KD)
- Joint pruning + quantization
- MCUNet (system-level co-design)

Metrics of evaluation: Top-1/Top-5 accuracy, number of parameters, FLOPs, model size, inference time, peak memory usage.

The edge device is the STM32H743 microcontroller (ARM Cortex-M7 at 480 MHz, 1 MB of SRAM, and 2 MB of Flash) running on the CMSIS-NN inference library.

Optimal transport recovery experiment configuration: Teacher model is ResNet-18 trained on CIFAR-100 (56.20% accuracy) and student model is a compressed version of the same architecture. Trained during 10 epochs using OT loss weight $\alpha = 0.5$. The OT loss takes use of differentiable 1D Wasserstein distance (sorted features, calculated as L2 distance between CDFs).

References

- [1] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015).
- Shuvo, M. M. H., et al. High-performance acceleration of deep learning inference in resource-constrained edge devices: a review. *Proceedings of the IEEE* 111, 62-99 (2023).
- Lin, J. et al. MCUNet: small-scale deep learning on IoT devices. *Adv. Neural Inf. Process. Syst.* 33, 1-14 (2020).
- Cheng, H., Zhang, M., and Shi, J. Q. Deep neural network pruning: a survey of taxonomy, comparison, analysis and recommendations. Preprint available at arXiv:2308.06767 (2023).

[5] Nahshan, Y. et al. Loss aware post-training quantization. Machine Learning 110, 1–22 (2021).

Hinton, G., Vinyals, O. and Dean, J. have published their work on distilling the knowledge in a neural network as a preprint in arXiv:1503.02531 (2015).

[7] Shwartz-Ziv, R. Opening the black box of deep neural networks using information. PhD thesis, The Hebrew University of Jerusalem (2022).

Na, C., Mehta, S. V. and Strubell, E. Train flat then compress: sharpness aware minimization learns more compressible models. in Proceedings of the Association for Computational Linguistics: EMNLP 2022 4903-4916 (2022).

A. Baptista et al. The Ricci flow as deep learning. Sci. Rep., 4:74045 (2024).

Figure legends

Figure 1 | Hessian eigenvalue spectrum and comparison of different pruning strategies.

The normalized distribution of Hessian eigenvalues of a convolutional layer in VGG-16. The spectrum has an obvious long-tail distribution with a small number of large eigenvalues that contribute significantly to losses.

b, The graphs of accuracy and pruning ratios when different pruning methods are used. Curvature-aware pruning (ordered by Hessian trace) performs much better than weight pruning and random pruning, which confirms the curvature-information duality.

Figure 2 | Results of curvature-aware adaptive pruning.

Information density across VGG-16 layers (approximated by Hessian trace). The shallow layers have the least information density (most redundant) whereas middle layers are moderately dense and deep layers are more dense (more important).

Accuracy comparison of curvature-aware pruning, weight pruning and random pruning. Curvature-aware pruning is much better than both weight pruning and random pruning at the same total compression rate, i.e., a benefit of 3.97% at 30 percent pruning ratio.

Figure 3 | Results of Wasserstein-aware optimal quantization.

Weight distribution of a layer and distributions of quantization points of uniform quantization and Lloyd-Max optimal quantization. In uniform quantization points are uniformly spaced, whereas in Lloyd-Max quantization points are densely packed in high density areas and less dense in tail areas- exactly how Wasserstein optimality is displayed.

b, Comparing the accuracy of various quantization approaches in VGG-16/CIFAR-10. The Lloyd-Max optimal quantization is superior to uniform quantization at all bit rates and 6-

bit quantization accuracy is also higher than the full-precision baseline (quantization noise regularization effect).

Figure 4 | Results of optimal transport accuracy recovery.

Figure a, shows per-epoch accuracy curves of various recovery mechanisms on ResNet-18/CIFAR-100. Optimal transport recovery (60.01) is much better than the conventional knowledge distillation (56.92), and it is improved by 3.09 percentage points.

Comparison of optimal transport and traditional knowledge distillation (principle). KL divergence only fits the form of the output distribution, whereas Wasserstein distance maintains the geometry of the feature space- this is the underlying cause of why OT recovery is more successful.

Competing interests

The authors declare no competing interests.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [quantizationexperimentlog.txt](#)
- [pruningexperimentlog.txt](#)
- [OptimalTransmissionAccuracyRecovery.txt](#)