Paper Review of the Image Calculator and Potential Next Steps

Hongyu Hè
hongyu.he@inf.ethz.ch

1 INTRODUCTION

In this document, I first review the Image Calculator (IC) paper (§2), highlighting its strengths (§2.2) and opportunities for improvement (§2.3), as well as some minor remarks (Appendix A). Inspired by the IC and related work, I propose SMMIL, a framework for self-designing multimodal datasets for multimodal models (§3).

2 REVIEW OF THE IMAGE CALCULATOR

In this section, I review the IC paper. To maintain clarity, I use "Fig." and "§" to refer to the figures and sections of this document, respectively; "Figure" and "Section" are exclusively used to refer to that of the reviewed paper. My evaluation only reflects my personal perspectives on the paper, considering my current level of understanding.

2.1 Brief Summary of Contribution

The authors propose the IC, a novel approach to image storage formats for AI applications. It challenges the use of JPEG, designed for human vision, by creating a dynamic storage format that adapts to specific AI tasks and datasets. The paper's novelty lies in generating an extensive design space encompassing various image storage options and employing efficient search methods within this space. By considering factors like inference time, accuracy, and storage, the IC achieves reductions of up to 8.2× in storage and 14.2× in inference time compared to JPEG, while maintaining or even improving accuracy. Its main results demonstrate a promising step forward in instance-optimized data processing in AI.

2.2 Strong Points

Below, I detail the strong points (**S**s) of the paper on IC.

(S1) **Ideal data format differs between AI models and humans, and varies across AI tasks.** A key takeaway from this paper is that data representations optimized for human consumption might not be inherently effective for AI models. It also varies across different tasks, for example, classification tasks do not require the pixel-level information as segmentation tasks do [13, 32], and therefore, can potentially afford higher compression rates. As a result, reconsidering current storage formats becomes crucial to exploring the potential adaptation or redesigning of formats specific to certain AI models and tasks. These insights inspire me to propose SMMIL (§3).

(S2) **First-principled approach towards data system optimization [20, 21].** This work showcases the effectiveness of dissecting the problem domain into atomic decisions, forming a tradeoff continuum for optimization. Such a design space allows for explicit reasoning and systematic (and possibly automated) exploration.

(S3) **Attacking the memory wall leading to substantial performance improvement.** This work is timely important as many researchers still solely focus on model-centric optimization, despite AI models typically constituting only about 65% of the entire machine learning (ML) pipeline [26, 35]. This paper demonstrates that by improving input storage and processing, model inference performance can be substantially improved due to less computation and data movement. As many AI models are memory bound [8, 18, 52], it enables deploying models on less powerful devices (Section 5.6) and could enable new application on-device or at the edge.

(S4) **Instance-optimized, end-to-end approach.** In the post-Moore's Law era, holistic methods thrive, while generic and static solutions fall short, especially at scale. Figure 17 in the paper well illustrates the limitations of isolated optimizations. While these ad-hoc methods may enhance some metrics (e.g., GPU processing time and data transfer overhead), they often do so at the expense of worsening other aspects, such as decoding time in this case. These approaches lack the ability to navigate the tradeoff space holistically (**S2**), resulting in suboptimal outcomes. The tradeoff space varies across tasks and datasets, which necessitates the need for instance-specific methods.

Furthermore, optimizing a system is akin to treating an ailment in a person; alleviating just one symptom will not cure the disease — end-to-end optimization demands that every stage of the process is considered within the tradeoff space. This methodology draws parallels with compiler-driven design techniques [4, 28, 40]. I believe that not only can ML models benefit from target-specific compilation, but present-day data systems also need such techniques to navigate challenges posed by energy and memory walls (§3).

(S5) **Using cheap proxies to guide design space exploration.** The authors leverage a smaller model type (ResNet50) as an efficient accuracy predictor to effectively explore the design space. Transfer learning is used to further facilitate the extensive sensitivity analysis on a multitude of dimensions such as subsampling strategies and compression rates. This approach has been widely used, for example, in the context of low-cost neural architecture search (e.g., [1, 12, 34, 42, 46]), and its application to this particular problem setting demonstrates its unique value in guiding the design process.[1]

2.3 Opportunities for Improvement

In this section, I summarize the opportunities for improvement (**O**s). *Should certain parts seem overly critical, please feel free to disregard those comments.*

(O1) **IC's Focus on DCT-Centric Compression Limits its Scope.** While acknowledging the innovative optimization techniques

[1] Admittedly, I was quite skeptical about this approach in the beginning, since the results of the sensitivity study are from ML models that have similar architectures mainly composed of fully connected and convolutional layers. Surprisingly, given the performance results in later sections, the proxy model seems to be effective even for models with significantly divergent architectures, such as Swin Transformer and MaxViT. Transformers deviate from convolution-based models, heavily relying on the attention mechanism, with Swin leaving out convolutional layers altogether.

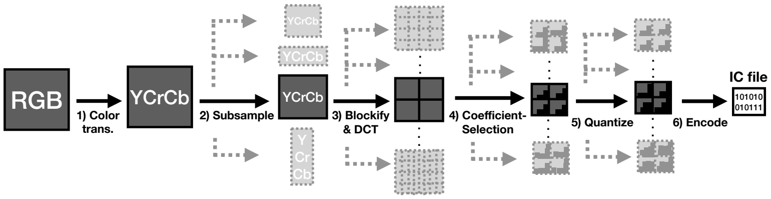


Fig. 1: The IC compression pipeline.

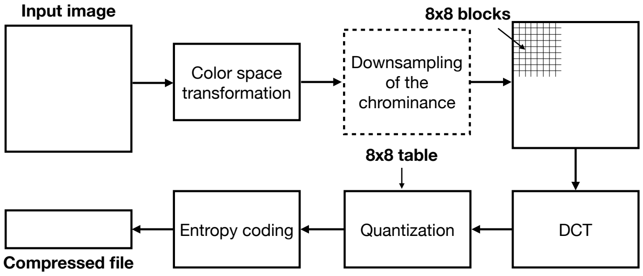
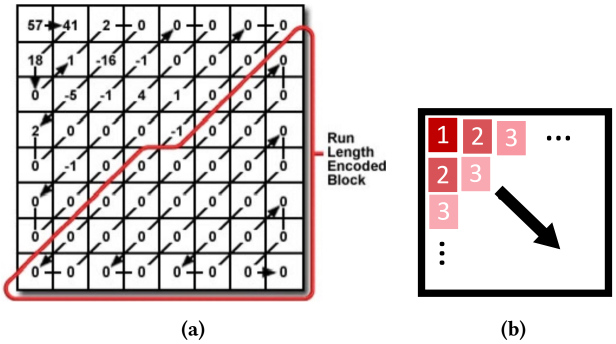


Fig. 2: Standard JPEG compression pipeline [36].



(a)

(b)

Fig. 3: (a) Entropy coding or "zigzag scanning" in JPEG (image source: [44]); (b) Proposed DCT coefficient selection strategy.

employed in developing IC, I believe the overall problem formulation remains primarily confined to a JPEG-like encoding-decoding pipeline (CODEC) anchored on discrete cosine transform (DCT). The proposed IC pipeline (Fig. 1) closely follows the JPEG CODEC (Fig. 2), with the primary distinction being the number of tunable knobs exposed to the optimization procedure.

Strictly speaking, JPEG is *not* a file format but an image CODEC standard offering several algorithmic options [7].[2] While JPEG primarily allows adjusting the quantization factor (Section 5.4), IC introduces a broader range of optimization options. Given the extensive research on both lossy and lossless CODECs that eschew DCT (e.g., [5, 9, 19, 23, 31, 36]), it would be interesting to expand the comparison of IC to include these significantly different algorithms, rather than primarily focusing on JPEG variants and lossless compression methods like zip and npz.

Likewise, storage formats mentioned in Section 4.1 like PNG, HEVC, and BMP do not use DCT either. Including them as comparison targets would provide a more comprehensive assessment of the proposed approach. For instance, employing the PNG format could result in even higher accuracy compared to JPEG, while occupying more storage space. Such comparisons could potentially create lager and more diverse "gaps" between the IC and traditional storage formats. Note that ML researchers often overlook image formats in their analyses, making these comparisons valuable.

(O2) **(Potentially) unnecessary experiments.** Firstly, the paper's conclusion in Section 4.2.1, stating that *"Brightness and Color Information Are Equally Important for AI Models,"* is a well-established notion in the ML community [43, 50, 54]. This claim is supported by many studies, and further experiments to corroborate it might not be essential.[3]

Secondly, the experiments conducted on coefficient selection strategies (Section 4.2.3 and Figures 4, 5) also appear unnecessary. DCT inherently produces blocks of coefficients representing the weights of cosine waves, already arranged in ascending order of their corresponding signal frequency from the upper-left corner to the lower-right corner. The higher-frequency coefficients, concentrated in the bottom-left corner and capturing fine-grained details like texture, are known to be as less crucial than their lower-frequency ones. This lower relevance of higher-frequency coefficients is reflected in commonly used JPEG quantization tables, imposing greater penalties on these coefficients, Hence, the two takeaways, *"Frequencies of the DCT Coefficients Increase Over The Spatial Dimensions"* and *"Low-Frequency Coefficients Are Much Useful Than High Frequency Coefficients,"* seem redundant.

Moreover, the best-performing *"Strategy 1"* depicted in Figures 4 and 5 are similar to, if not the same as, the principles of entropy coding in JPEG (Fig. 2), also commonly known as zigzag encoding. This selection/scanning method not only prioritizes encoding the most important information (lower-frequency coefficients) but also often generates trailing zeros (due to heavier quantization factors mentioned above), enabling efficient compression (Fig. 3a). In other words, *"Strategy 1"* (Fig. 3b) is effective not just valuable for ML models but also for humans in general, which is why it is employed by JPEG. Moreover, classification tasks generally do not require fine-grained details from the image (provided by the higher-frequency DCT coefficients). Therefore, it is unsurprising to see that *"Strategy 1"* leads to much better performance in Figure 6.

Given the above points, it could be beneficial to consider replacing Section 4.2.3 and its associated figures (Figure 4, 5, and 6) with alternative experiment details from the technical report, such as the examination of quantization factors' impact (Section 4.2.4), which could offer more compelling insights and broaden the scope of included experiments.

[2] Technically, JPEG File Interchange Format (JFIF) [51] is the file format corresponding to images encoded using the (default) JPEG algorithm.

[3] In fact, some groups at ETH AI Center are working on addressing the impact of brightness on self-driving cars.

(O3) Specifications of input representations. I may have missed it, but how the inputs of the IC pipeline (i.e., the datasets, models, and hardware) are represented is unclear. These representations are an integral part of the methodology and play a significant role in its effectiveness. For instance, hardware can be represented by performance statistics obtained from a set of representative workloads [16]. It can also be represented by learned hardware embeddings in the latent space [2]. Thus, I believe specifying the input representations explicitly is important.

(O4) Interactions among design dimensions. The authors employ a bottom-up approach to prune the design space, conducting sensitivity analysis on individual design decisions mostly independently. While this method expedites the elimination of redundant dimensions at an early stage, it neglects the potential interactions among these dimensions, which can lead to both positive and negative compounding effects. This assumption of independence may overlook potentially valuable combinations of design primitives. Although considering these interactions would further enlarge the huge search space, alternative methods exist, such as a top-down approach that eliminates factors based on their overall impact on performance [1].

(O5) Leveraging relative rankings over absolute accuracy. This comment refers to **S5** — the author propose utilizing a cheaper model as a low-cost proxy to guide the search process. While acknowledging the merits of this method, the reliance on absolute accuracy values may not be the most efficient approach. In contrast, a common strategy involves constructing an even more lightweight proxy model capable of predicting the relative rankings (e.g., error to top-k and Kendall's Tau) among architectural choices [1, 38, 42]. This approach could obviate the need to train a proxy model of the same family and, in turn, the need for transfer learning.

(O6) Concerns about bucket sampling. The proposed bucket sampling technique hinges on the premise that numerous design decisions exhibit correlation and cluster along a tradeoff continuum. While this method efficiently eliminates undesirable decisions in batches, there are concerns regarding the precise definition of "critical buckets," particularly in the context of section 4.3.1. Clarity is needed on the criteria for identifying these critical buckets and determining the cutoff threshold.

Moreover, the paper asserts that this sampling approach results in error rates as low as 1%, a claim also supported by Figure 8. However, the absence of a proper baseline for comparison makes it challenging to contextualize this result. For instance, if an alternative scattershot sampling method already achieves an approximate 2% error rate, the feasible headroom for improvement is not significantly in the first place. Therefore, establishing a concrete baseline for the comparison could be valuable in terms of accurately gauging the effectiveness of bucket sampling technique.

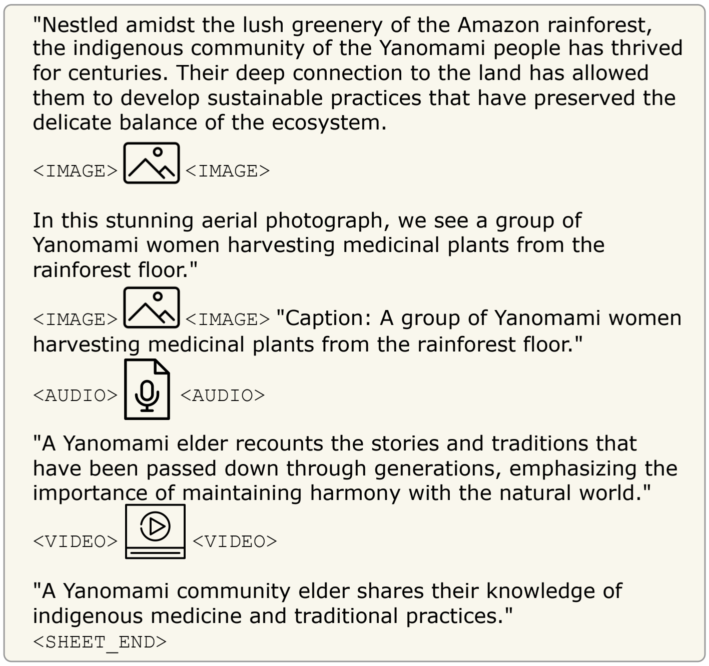
(O7) Considering heuristics over brute-force search as a baseline search algorithm. Compared to a brute-force method, a slightly stronger baseline would be employing simple heuristics. For example, results from Figures 7-10 reveal a strong correlation between model quality and compression rate. Hence, a sensible heuristic might subtly prioritize the quantization factor while balancing other dimensions.

Moreover, if procedures involving diverse storage types like PNG and BMP (**O1**) were explored, leveraging straightforward heuristics could directly be leveraged for generating desired IC formats. For instance, tailoring the search algorithm to lean toward PNG-like formats for datasets predominantly comprising characters and diagrams (featuring simple patterns and sharp transitions) and favoring DCT-based methods for more detailed photos could be more intuitive and efficient than exhaustive brute-force search.

(O8) Uncontrolled, substantial performance variability. This comment pertains to experiments that sweep either the inference time or the storage size. While acknowledging the considerable flexibility offered by the IC, the observed wide-ranging performance variability is particularly worrying. For instance, in Figure 11, IC can lead to considerable model performance degradation, reaching an accuracy of approximately 0.2, without offering explicit knobs to control such variations. This variability poses a significant concern since stability and predictability of performance are vital in production. The absence of mechanisms to explicitly control or mitigate this variability could impede industry adoption.

Consequently, the claim of a net 9.2× speedup needs refinement. The authors should define an accuracy threshold/cutoff below which the resulting model is considered unusable. Additionally, despite the authors' efforts in warming caches before each experiment, in my experience, the inherent variability in ML model inference times across measurements might still persist. Therefore, considering metrics like tail latency (e.g., p95) that aggregate measurements above the cutoff accuracy could offer a more ideal assessment, especially for production-grade considerations. This improvement could better reflect practical deployment scenarios and address concerns related to performance predictability and stability.

(O9) Integrating energy efficiency as a metric of interest. In the cloud, ML workloads are power bound [22, 37, 41, 45, 52, 56], where energy consumption stands as a bottom line of cloud providers [3, 24]. Recognizing that performance does not translate directly to energy efficiency, it is pivotal to extend the scope of metrics to encompass energy considerations. Relying solely on performance metrics dismisses the significant impact of power consumption, and incorporating energy efficiency in the design space could drastically change the tradeoff continuum [55]. Thus, understanding and improving the sustainability and energy efficiency of cloud applications necessitates a holistic view that integrates power consumption metrics alongside performance evaluations.



"Nestled amidst the lush greenery of the Amazon rainforest, the indigenous community of the Yanomami people has thrived for centuries. Their deep connection to the land has allowed them to develop sustainable practices that have preserved the delicate balance of the ecosystem.

<IMAGE> 🖼 <IMAGE>

In this stunning aerial photograph, we see a group of Yanomami women harvesting medicinal plants from the rainforest floor."

<IMAGE> 🖼 <IMAGE> "Caption: A group of Yanomami women harvesting medicinal plants from the rainforest floor."

<AUDIO> 🎤 <AUDIO>

"A Yanomami elder recounts the stories and traditions that have been passed down through generations, emphasizing the importance of maintaining harmony with the natural world."

<VIDEO> ▶ <VIDEO>

"A Yanomami community elder shares their knowledge of indigenous medicine and traditional practices."
<SHEET_END>

Fig. 4: Example multimodal interleaved dataset.

3 SMMIL: SELF-DESIGNING MULTIMODAL INTERLEAVED DATA FORMATS

Inspired by the IC and recent work on instance-optimized data systems [10, 11, 14, 15, 21, 25, 27, 33], I propose SMMIL, self-designing interleaved data formats for multimodal models.

3.1 Motivation

At Apple, my research focused on bridging the gaps between modalities (text, images, and video) and tasks (captioning, VQA, and summarization) of multimodal LLMs. The key to achieving multimodal capabilities is extensive fine-tuning on interleaved datasets (e.g., [30, 49, 57]). These datasets, meticulously crafted with special tokens, interleave data from various modalities in an integrated setting (Fig. 4).

Exposing LLMs to diverse modalities within a single context window enables multimodal instruction tuning and in-context learning, allowing them to seamlessly adapt to a wide range of tasks. This approach aligns with Google DeepMind's development of Gemini [47], a recent multimodal advancement, which the company explicitly built *"from the ground up to be multimodal."*

Unfortunately, the process of training such a gigantic multimodal model from scratch requires a huge amount of interleaved pretraining data, often orders of magnitude larger than what fine-tuning alone would demand. This shift towards data-centric AI race (especially in the industry) further emphasizes the indispensable role of multimodal interleaved datasets in fueling the development of cutting-edge multimodal models.

3.2 Gaps

Despite the growing prominence of multimodal interleaved datasets, their efficient representation, update, versioning, and storage remain underexplored areas of research. To my knowledge, even leading technology companies still rely on rudimentary approaches that simply concatenate data from different modalities and compress the entire bundle for storage.

This simplistic approach often leads to excessive storage overhead, as the different modalities are lumped together, incurring redundancies and inefficiencies. On the other hand, splitting modalities for separate storage introduces the challenge of runtime loading, injection, and formatting, which are computationally prohibitive. To avoid these runtime I/O costs, companies often resort to upfront construction of interleaved datasets, requiring days of processing time. However, this approach makes minor updates or changes — such as altering the special token for images from <IMG> to <IMAGE> or adjusting the frequency of modalities (e.g., from two to three images every 200 text tokens) — extremely cumbersome, often necessitating the reconstruction of the entire dataset.

Such inefficiencies call for new solutions that tackle the tradeoffs among data representation, storage overhead, and I/O costs. Such advancements will pave the way for more effective and efficient use of multimodal interleaved datasets, unlocking their full potential in driving the development of cutting-edge AI models.

3.3 Research Questions and Challenges

The aforementioned gaps underscore the need for a comprehensive solution that addresses the dynamic nature of multimodal interleaved datasets and their interplay with different LLMs and tasks. This leads to the main research question (**MRQ**) of SMMIL:

MRQ: How can we design efficient storage formats for interleaved datasets that *automatically adapt to* different multimodal models and tasks?

Answering the **MRQ** demands overcoming several fundamental challenges.

Adaptive Structure. The interleaved dataset structure must dynamically adjust to the specific requirements of different tasks. For instance, while captioning tasks may favor one image followed by a short text sequence, VQA tasks might benefit from higher image frequencies and longer text sequences. Likewise, the optimal layout may vary depending on the type and architecture of the target LLM, since the model type dictates models' capacity and limitations in their receptive fields (e.g., the context window length). Most importantly, the dataset structure should be able to adapt to changes in the model and tasks over time.

Effective Modality Representation. AI models perceive the world differently than humans do (**S1**), often requiring task-specific representations for the same modality. For example, consider audio data: Mel-frequency cepstral coefficients (MFCCs) summarize spectral characteristics. MFCCs are commonly used in speech recognition, speaker identification, and music genre classification. In contrast, Short-time Fourier transform (STFT) spectrograms break down audio into time-based frequency components, employed in music analysis, sound event detection, and audio synthesis.

Moreover, choosing appropriate CODECs for different modalities presents an equally complex challenge. Should each modality be compressed separately using different methods, or is a unified CODEC preferable? Determining optimal quantization techniques for each modality, task, and LLM adds further complexity to the design decision-making process.

Efficient Search. The design space of SMMIL can be significantly larger and more intricate than that of IC, as it effectively involves solving multiple IC-like problems together. To efficiently navigate this multifaceted optimization landscape, techniques such as active learning, Bayesian optimization, or reinforcement learning (RL) could be employed.

However, the key challenge lies in finding an effective, low-cost acquisition function, i.e., the reward/cost model (**S5**), for guiding the search process. Several factors lead to this challenge. First, the optimization landscape might be more complex to navigate due to the addition of many more design dimensions that interplay delicately with each other as described above. Consequently, it is less likely to find a single dominant factor (e.g., the compression rate) toward which the search procedure can be biased. Moreover, prior work [4, 10, 21, 25, 39] involving automated design space exploration has shown the importance of obtaining *indicative* and *timely* feedback during the search process. Since multimodal LLMs typically have a much larger capacity, obtaining such feedback from them becomes harder as they may be insensitive to minor format tweaks, unless substantial amounts of tuning data (and potentially time [17, 53]) is invested, which could inflate the cost of the search process beyond that of the IC.

3.4 Research Approach

Finding instance-optimized dataset structures and suitable modality representations implies traversing a vast combinatorial tradeoff space. Building upon insights from the IC (**S2**, **S4**), I plan to start by breaking down this tradeoff space into *first-principle primitives*. I intend to adopt a compiler-driven strategy [29, 40], creating an abstraction hierarchy that optimizes these primitives in a top-down manner. Specifically, the approach will first optimize the high-level structure of multimodal datasets, leaving modality representations as opaque nodes [4, 28]. Given the subset of optimized dataset structures and layouts, subsequent efforts will then focus on exploring and selecting suitable representations for each modality within. This hierarchical approach aims to efficiently navigate the tradeoff space, ensuring a structured and systematic exploration of primitive combinations.

To mitigate the challenges posed by large tradeoff space, a potential strategy involves *embedding constraint solvers* within the learned search algorithm. For example, integrating simple RL methods with constraint solvers has shown to provide a robust framework to manage this complexity, leveraging effective human heuristics as hard constraints and propagate them eagerly [6, 28, 48]. These heuristics, grounded in domain expertise, could delineate which structure/representations suit specific tasks and indicate sensitivity or insensitivity to information loss (for CODEC selection).

In summary, SMMIL aims to improve multimodal interleaved datasets by enabling dynamic adaptation of the data formats to various LLMs and tasks. It can pave the way for more efficient, scalable, and versatile multimodal data handling, unlocking the full potential of these valuable resources for driving advancements in multimodal AI. The breadth and depth of this topic invite further exploration into followup research questions, such as the efficient indexing, injection, and update of multimodal interleaved data (in terms of both performance and energy), which are beyond the scope of this document given the page limits.

REFERENCES

[1] Mohamed S Abdelfattah, Abhinav Mehrotra, Łukasz Dudziak, and Nicholas D Lane. 2021. Zero-cost proxies for lightweight nas. *arXiv preprint arXiv:2101.08134* (2021).

[2] Yash Akhauri and Mohamed S Abdelfattah. 2023. Multi-Predict: Few Shot Predictors For Efficient Neural Architecture Search. *arXiv preprint arXiv:2306.02459* (2023).

[3] Luiz André Barroso, Urs Hölzle, and Parthasarathy Ranganathan. 2019. *The datacenter as a computer: Designing warehouse-scale machines.* Springer Nature.

[4] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. 2018. {TVM}: An automated {End-to-End} optimizing compiler for deep learning. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18).* 578–594.

[5] John Clyne, Pablo Mininni, Alan Norton, and Mark Rast. 2007. Interactive desktop analysis of high resolution simulations: application to turbulent plume dynamics and current sheet formation. *New Journal of Physics* 9, 8 (2007), 301.

[6] Basile Clément and Albert Cohen. 2022. Translation Validation of Tensor Programming Languages. In *PACM PL, OOPSLA 2022.*

[7] The Joint Photographic Experts Group committee. 2023. Workplan & Specs of JPEG 1. https://jpeg.org/jpeg/workplan.html

[8] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems* 35 (2022), 16344–16359.

[9] Sheng Di and Franck Cappello. 2016. Fast error-bounded lossy HPC data compression with SZ. In *2016 ieee international parallel and distributed processing symposium (ipdps).* IEEE, 730–739.

[10] Jialin Ding, Umar Farooq Minhas, Jia Yu, Chi Wang, Jaeyoung Do, Yinan Li, Hantian Zhang, Badrish Chandramouli, Johannes Gehrke, Donald Kossmann, et al. 2020. ALEX: an updatable adaptive learned index. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data.* 969–984.

[11] Jialin Ding, Vikram Nathan, Mohammad Alizadeh, and Tim Kraska. 2020. Tsunami: A learned multi-dimensional index for correlated data and skewed workloads. *arXiv preprint arXiv:2006.13282* (2020).

[12] Lukasz Dudziak, Thomas Chau, Mohamed Abdelfattah, Royson Lee, Hyeji Kim, and Nicholas Lane. 2020. Brp-nas: Prediction-based nas using gcns. *Advances in Neural Information Processing Systems* 33 (2020), 10480–10490.

[13] William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research* 23, 1 (2022), 5232–5270.

[14] Paolo Ferragina and Giorgio Vinciguerra. 2020. The PGM-index: a fully-dynamic compressed learned index with provable worst-case bounds. *Proceedings of the VLDB Endowment* 13, 8 (2020), 1162–1175.

[15] Alex Galakatos, Michael Markovitch, Carsten Binnig, Rodrigo Fonseca, and Tim Kraska. 2019. Fiting-tree: A data-aware index structure. In *Proceedings of the 2019 international conference on management of data.* 1189–1206.

[16] X Yu Geoffrey, Yubo Gao, Pavel Golikov, and Gennady Pekhimenko. 2021. Habitat: A {Runtime-Based} computational performance predictor for deep neural network training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21).* 503–521.

[17] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556* (2022).

[18] Samuel Hsia, Udit Gupta, Bilge Acun, Newsha Ardalani, Pan Zhong, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. 2023. MP-Rec: Hardware-Software Co-design to Enable Multi-path Recommendation. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3.* 449–465.

[19] Xiaomeng Huang, Yufang Ni, Dexun Chen, Songbin Liu, Haohuan Fu, and Guangwen Yang. 2016. Czip: A Fast Lossless Compression Algorithm for Climate Data. *International Journal of Parallel Programming* 44 (2016), 1248–1267.

[20] Stratos Idreos, Konstantinos Zoumpatianos, Manos Athanassoulis, Niv Dayan, Brian Hentschel, Michael S. Kester, Demi Guo, Lukas M. Maas, Wilson Qin, Abdul Wasay, and Yiyou Sun. 2018. The Periodic Table of Data Structures. *IEEE Data Eng. Bull.* 41 (2018), 64–75. https://api.semanticscholar.org/CorpusID:52190923

[21] Stratos Idreos, Kostas Zoumpatianos, Brian Hentschel, Michael S Kester, and Demi Guo. 2018. The data calculator: Data structure design and cost synthesis

from first principles and learned cost models. In *Proceedings of the 2018 International Conference on Management of Data*. 535–550.

[22] Gamze Islamoglu, Moritz Scherer, Gianna Paulin, Tim Fischer, Victor JB Jung, Angelo Garofalo, and Luca Benini. 2023. Ita: An energy-efficient attention and softmax accelerator for quantized transformers. In *2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 1–6.

[23] Jeremy Iverson, Chandrika Kamath, and George Karypis. 2012. Fast and effective lossy compression algorithms for scientific datasets. In *Euro-Par 2012 Parallel Processing: 18th International Conference, Euro-Par 2012, Rhodes Island, Greece, August 27-31, 2012. Proceedings 18*. Springer, 843–856.

[24] Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, et al. 2023. Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*. 1–14.

[25] Andreas Kipf, Ryan Marcus, Alexander van Renen, Mihail Stoian, Alfons Kemper, Tim Kraska, and Thomas Neumann. 2020. RadixSpline: a single-pass learned index. In *Proceedings of the third international workshop on exploiting artificial intelligence techniques for data management*. 1–5.

[26] Ana Klimovic. [n. d.]. Rethinking Data Storage and Preprocessing for ML. https://www.sigarch.org/rethinking-data-storage-and-preprocessing-for-ml/

[27] Tim Kraska, Alex Beutel, Ed H Chi, Jeffrey Dean, and Neoklis Polyzotis. 2018. The case for learned index structures. In *Proceedings of the 2018 international conference on management of data*. 489–504.

[28] Chris Lattner, Mehdi Amini, Uday Bondhugula, Albert Cohen, Andy Davis, Jacques Pienaar, River Riddle, Tatiana Shpeisman, Nicolas Vasilache, and Oleksandr Zinenko. 2021. MLIR: Scaling compiler infrastructure for domain specific computation. In *2021 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*. IEEE, 2–14.

[29] Chris Arthur Lattner. 2002. LLVM: An infrastructure for multi-stage optimization. (2002).

[30] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. 2023. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726* (2023).

[31] Peter Lindstrom. 2014. Fixed-rate compressed floating-point arrays. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2674–2683.

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10012–10022.

[33] Ryan Marcus, Parimarjan Negi, Hongzi Mao, Nesime Tatbul, Mohammad Alizadeh, and Tim Kraska. 2021. Bao: Making learned query optimization practical. In *Proceedings of the 2021 International Conference on Management of Data*. 1275–1288.

[34] Abhinav Mehrotra, Alberto Gil CP Ramos, Sourav Bhattacharya, Łukasz Dudziak, Ravichander Vipperla, Thomas Chau, Mohamed S Abdelfattah, Samin Ishtiaq, and Nicholas Donald Lane. 2020. NAS-Bench-ASR: Reproducible neural architecture search for speech recognition. In *International Conference on Learning Representations*.

[35] Derek G Murray, Jiri Simsa, Ana Klimovic, and Ihor Indyk. 2021. tf. data: A machine learning data processing framework. *Proceedings of the 47th International Conference on Very Large Data Bases* (2021).

[36] Tina Nikoukhah, Miguel Colom, Jean-Michel Morel, and Rafael Grompone von Gioi. 2022. A Reliable JPEG Quantization Table Estimator. *Image Processing On Line* 12 (2022), 173–197.

[37] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350* (2021).

[38] Phitchaya Mangpo Phothilimthana, Amit Sabne, Nikhil Sarda, Karthik Srinivasa Murthy, Yanqi Zhou, Christof Angermueller, Mike Burrows, Sudip Roy, Ketan Mandke, Rezsa Farahani, et al. 2021. A flexible approach to autotuning multi-pass machine learning compilers. In *2021 30th International Conference on Parallel Architectures and Compilation Techniques (PACT)*. IEEE, 1–16.

[39] Sanket Purandare, Abdul Wasay, Stratos Idreos, and Animesh Jain. 2023. μ-TWO: 3× Faster Multi-Model Training with Orchestration and Memory Optimization. *Proceedings of Machine Learning and Systems* 5 (2023).

[40] Jonathan Ragan-Kelley, Connelly Barnes, Andrew Adams, Sylvain Paris, Frédo Durand, and Saman Amarasinghe. 2013. Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. *Acm Sigplan Notices* 48, 6 (2013), 519–530.

[41] Matteo Risso, Alessio Burrello, Giuseppe Maria Sarda, Luca Benini, Enrico Macii, Massimo Poncino, Marian Verhelst, and Daniele Jahier Pagliari. 2023. Precision-aware Latency and Energy Balancing on Multi-Accelerator Platforms for DNN Inference. *arXiv preprint arXiv:2306.05060* (2023).

[42] Yu Shen, Yang Li, Jian Zheng, Wentao Zhang, Peng Yao, Jixiang Li, Sen Yang, Ji Liu, and Bin Cui. 2023. Proxybo: Accelerating neural architecture search via bayesian optimization with zero-cost proxies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 9792–9801.

[43] Inderjeet Singh, Satoru Momiyama, Kazuya Kakizaki, and Toshinori Araki. 2021. On brightness agnostic adversarial examples against face recognition systems. In *2021 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 1–5.

[44] Kenneth R. Spring. 2023. JPEG Image Compression. https://www.olympus-lifescience.com/en/microscope-resource/primer/java/olympusmicd/digitalimaging/jpegcompression/.

[45] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243* (2019).

[46] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. 2019. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2820–2828.

[47] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).

[48] S. VenkataKeerthy, Siddharth Jain, Anilava Kundu, Rohit Aggarwal, Albert Cohen, and Ramakrishna Upadrasta. 2023. RL4ReAl: Reinforcement Learning for Register Allocation. In *CC 2023*.

[49] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, et al. 2023. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942* (2023).

[50] Xingxing Wei, Ying Guo, and Bo Li. 2021. Black-box adversarial attacks by manipulating image attributes. *Information sciences* 550 (2021), 285–296.

[51] Wikipedia. 2023. JPEG File Interchange Format. https://en.wikipedia.org/wiki/JPEG_File_Interchange_Format

[52] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems* 4 (2022), 795–813.

[53] Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2023. To Repeat or Not To Repeat: Insights from Scaling LLM under Token-Crisis. *arXiv preprint arXiv:2305.13230* (2023).

[54] Bo Yang, Kaiyong Xu, Hengjun Wang, and Hengwei Zhang. 2022. Random Transformation of image brightness for adversarial attack. *Journal of Intelligent & Fuzzy Systems* 42, 3 (2022), 1693–1704.

[55] Jie You, Jae-Won Chung, and Mosharaf Chowdhury. 2023. Zeus: Understanding and Optimizing {GPU} Energy Consumption of {DNN} Training. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 119–139.

[56] Luca Zanatta, Alfio Di Mauro, Francesco Barchi, Andrea Bartolini, Luca Benini, and Andrea Acquaviva. 2023. Directly-trained Spiking Neural Networks for Deep Reinforcement Learning: Energy efficient implementation of event-based obstacle avoidance on a neuromorphic accelerator. *Neurocomputing* 562 (2023), 126885.

[57] Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. 2023. Multimodal c4: An open, billion-scale corpus of images interleaved with text. *arXiv preprint arXiv:2304.06939* (2023).

A REMARKS

In this section, I offer minor remarks (**R**s) relating to typos and formatting, as they may be useful for the camera-ready version. *Feel free to skip any remarks that appear overly pedantic.*

(**R1**) Section 2: “pixel domain” → “spatial domain.” This is a somewhat unconventional terminology. “spatial domain” may be more mathematically intuitive.

(**R2**) Section 2: might be useful to specify that Eq. (1) is type-II DCT.

(**R3**) Section 2: “This way, images are represented in the structured frequency domain rather than the unstructured pixel domain.” The term “unstructured pixel domain” might not be entirely accurate. The spatial/pixel domain should be structured in the sense that each pixel carries specific information about color, intensity, and position within the image. It is not inherently unstructured, but rather organized differently from the frequency domain.

(**R4**) Section 4.2: “[...] and the value it brings in end-to-end performance.” → “[...] and the value it brings **to** end-to-end performance.”

(**R5**) Section 4.2.3: “compress” → “compresses”

(**R6**) Section 4.2.3: “Coefficients” → “coefficients”

(**R7**) Section 4.2.3: “Low-Frequency Coefficients Are Much Useful Than High Frequency Coefficients” → “Low-Frequency Coefficients Are Much **More** Useful Than High Frequency Coefficients.”

(**R8**) Section 4.3.1: text formatting in Eq. (3), e.g., acc_{bucket_i}.

(**R9**) Section 4.3.1: Eq. (3) may be imprecise, and the ranges of variables should be specified — what if $i = 0$ or $i = s88$?

(**R10**) Figure 9: for “ImageNet-subset”, I am unsure why transfer-training more epochs would make the performance worse. Is it due to overfitting? Maybe worth clarifying. Also, the caption states that the model construction time is an order of magnitude better, which is not evident from the figure itself, and to which configuration it refers to is also unclear.

(**R11**) Section 5.1: For “We tune the batch size for each model and use the highest-performing batch size for each,” it should be clearly stated that the batch size is kept the same when comparing JPEG and the proposed approach. This sentence reads as if batch size was configured and independently tuned for each setup.

(**R12**) Section 5.1: “[...] we use the same parameters [...]” → “[...] we use the same **model configurations** [...]”

(**R13**) Section 5.9: “figure 17” → “Figure 17”

(**R14**) Section 5.9: “[...] benefits in image analysis tasks beyond clustering [...]” → “[...] benefits in image analysis tasks beyond **classification** [...]”

(**R15**) The figures are a bit scattered, which can cause surprises in reading. I tend to softly pin all figures (except for the embedded ones) at the top of the pages.