Continuous Efficient Rank Evolution Strategy for Model Compression

Cheng-Yu Sie, Ching-Chi Lee, Che-Rung Lee, and Chuan-Kang Ting
Department of Computer Science, National Tsing Hua University, Hsinchu 300044, Taiwan
s113062541@m113.nthu.edu.tw, s112033631@m112.nthu.edu.tw, cherung@cs.nthu.edu.tw, ckting@cs.nthu.edu.tw

***Abstract*—Model compression constitutes a complex optimization problem characterized by vast search spaces. While low-rank decomposition has been widely adopted for model compression, its performance is limited by the rank selection. Previous studies formulated rank selection as a one-shot neural architecture search (NAS) problem but suffered from high search costs due to full-dataset evaluations and limited optimization potential caused by the discrete rank representation. To address these issues, this study proposes the continuous efficient rank evolution strategy (CERES), which transforms rank selection into a continuous optimization problem to enable fine-grained search and utilizes a minibatch strategy for rapid evaluation. To mitigate the stochastic noise introduced by minibatch evaluation, CERES employs the covariance matrix adaptation evolution strategy (CMA-ES) as the core solver. Experimental results on ImageNet-1k demonstrate that CERES outperforms state-of-the-art techniques; specifically, CERES reduces search time by approximately 12 times while maintaining competitive accuracy.**

***Index Terms*—Evolutionary computation, low-rank decomposition, covariance matrix adaptation evolution strategy, batch optimization.**

I. Introduction

Deep learning models have achieved remarkable success across various domains. However, their massive parameter counts often exert a heavy computational burden on resource-constrained edge devices. This challenge is particularly acute in Vision Transformers (ViTs), which are characterized by high architectural complexity and substantial resource demands. To alleviate these computational burdens, low-rank decomposition has been widely adopted as a standard compression paradigm. Distinct from heuristic-based pruning, this approach utilizes truncated singular value decomposition (Truncated SVD) [1] to provide a theoretically grounded approximation. Specifically, this approach approximates weight matrices by retaining only the most significant top-k singular values, a process referred to as rank selection. Hence, the core challenge lies in determining the optimal rank k_i for each individual layer i. This results in a vast search space where one must balance the trade-off between minimizing the total parameter count and maintaining predictive performance.

Given the problem's inherent complexity, traditional methods often rely on heuristic rules or brute-force search, which are typically suboptimal or infeasible. Recently, evolutionary computation (EC) has shown considerable promise in automating this process. The state-of-the-art framework FLORA [2] successfully formulated rank selection as a one-shot neural architecture search (NAS) problem using genetic algorithm (GA). Despite its promise, FLORA operates under two major constraints. First, it relies on computationally expensive full-dataset evaluations to guide the search, resulting in prohibitively high search costs. Second, to manage the search space complexity, FLORA restricts rank choices to a discrete, coarse-grained set, which limits the optimization potential.

To overcome these inefficiencies, this study proposes a high-efficiency evolutionary framework, named continuous efficient rank evolution strategy (CERES). The proposed framework is built upon two key innovations. First, to accelerate the search, it introduces a minibatch optimization strategy that replaces computationally expensive full-dataset evaluations with rapid stochastic estimates. Second, it reformulates the rank selection problem within a continuous search space. The mapping is crucial, as it expands the optimization potential, allowing for precise, fine-grained adjustment of ranks that discrete methods often fail to capture. This shift to a rapid, continuous search paradigm necessitates a robust optimizer. To this end, we utilize the covariance matrix adaptation evolution strategy (CMA-ES) [3] to mitigate the stochastic noise introduced by minibatch estimation. CMA-ES capitalizes on its robust global search capability to exploit fine-grained correlations in continuous domains, rendering it superior to traditional GA in this context.

The major contributions of this paper are summarized as follows:

- CERES is presented to reformulate the rank selection as a continuous optimization problem. This mapping enables fine-grained compression ratios, facilitating more precise model approximation than coarse-grained discrete methods can achieve.
- A minibatch strategy is integrated with the CMA-ES solver to form the core optimizer of CERES. This combination efficiently addresses the continuous problem by replacing computationally expensive evaluations with rapid estimates while maintaining robustness against stochastic noise.
- The proposed framework is empirically validated on the ImageNet-1k dataset using DeiT-B architectures. The results demonstrate that CERES significantly reduces search costs, achieving an approximate $12\times$ speedup over the FLORA baseline while maintaining comparable model accuracy.

The remainder of this study is organized as follows. Section II provides a brief introduction to the related work. Section III describes the proposed method in detail. Section IV presents the experiment settings and results. Finally, conclusions are drawn in Section V.

II. Related Work

This section reviews the literature pertinent to the proposed framework. First, we discuss the current advancements in transformer compression, with a focus on low-rank approximation. Second, the formulation of rank selection is examined as a NAS problem. Finally, we explore efficient evolutionary optimization techniques, more specifically focusing on minibatch optimization strategies and the application of CMA-ES in noisy environments.

A. Transformer Compression and Low-Rank Approximation

With the widespread adoption of transformer-based models in both natural language processing (NLP) and computer vision, model compression has become a critical research area. Existing techniques can be broadly categorized into pruning [4], quantization [5], knowledge distillation [6], and low-rank decomposition [2, 7]. Among these, low-rank decomposition has gained prominence for its mathematical rigor in reducing redundancy by approximating weight matrices.

Early approaches typically applied a uniform compression ratio across all layers or relied on heuristic rules to determine ranks. However, recent studies have demonstrated that different layers exhibit varying sensitivities to information loss. Rank selection, which involves finding the specific optimal rank for each layer, has emerged as a key optimization challenge. For instance, Bolaco [8] addresses this in large language models (LLMs) by utilizing bayesian optimization to determine ranks based on output vector distributions. While effective for LLMs, such numerical analysis methods often require specific assumptions about layer importance and may not fully exploit the global interaction between layers in ViTs. These methods may fail to fully capture the complex, global interactions between layers in ViTs, creating a need for more holistic search-based strategies.

B. Neural-Architecture-Search for Rank Selection

To address the complexity of determining optimal model configurations, recent research has increasingly adopted NAS [9, 10]. Early NAS methods were hindered by excessive computational demands. However, the advent of one-shot NAS has mitigated this issue by enabling efficient architecture discovery through weight-sharing strategies [11]. In this paradigm, an over-parameterized supernet is trained once, and candidate sub-networks inherit weights directly from it, bypassing the need for training from scratch.

Recent studies have thus formulated rank selection as a specific NAS task. Distinct from traditional NAS which focuses on kernel sizes or topology, methods like FLORA [2] address the compression of dense layers in ViTs. Notably, FLORA streamlines the search by mapping the rank of each layer to a limited set of discrete compression rates. Despite its effectiveness, this discrete representation lacks the precision to capture fine-grained optimal solutions. Moreover, the process is hindered by the high computational cost associated with full-dataset validation.

C. Minibatch Optimization in Evolutionary Computation

The high cost of exact fitness evaluation in NAS [12] exerts a significant constraint on search efficiency. While gradient-based methods standardly use minibatch stochastic gradient descent (SGD) to approximate the loss function [13], traditional EC methods typically assume exact fitness evaluations. To bridge this gap, recent research has integrated minibatch strategies into EC. For instance, the limited evaluation evolutionary algorithms (LEEA) [14] evaluates each generation using only a subset of data.

However, evaluating individuals on random minibatches introduces stochastic noise, leading to the noisy fitness problem where high-quality solutions might be discarded due to a particularly difficult data batch. Techniques such as fitness inheritance have been proposed to mitigate this by smoothing estimates over time [15]. Despite these advances, applying simple GAs to high-variance minibatch evaluations remains challenging, necessitating optimizers that are inherently robust to uncertainty.

D. Evolutionary Optimization under Uncertainty

Addressing noisy optimization is a well-established subfield in evolutionary computation [16, 17]. Traditional discrete GAs are often sensitive to noise variance. By contrast, evolution strategies, particularly the CMA-ES, have demonstrated superior robustness in noisy environments [18, 19].

CMA-ES operates on a continuous search space and utilizes a population-based approach where the update of the covariance matrix aggregates information from multiple samples. This mechanism effectively averages out stochastic noise from minibatch evaluations. Moreover, unlike the discrete combinatorial search used in standard NAS frameworks [20], CMA-ES facilitates the exploration of continuous search spaces. This capability is particularly relevant for rank selection, as it allows the algorithm to treat rank configurations as continuous variables, enabling fine-grained optimization that discrete NAS methods like FLORA cannot achieve.

III. Methodology

This study presents a novel framework, CERES, which establishes a continuous search space for fine-grained rank selection and introduces a minibatch evaluation strategy to accelerate fitness estimation. Restated, this strategy transforms rank selection into a noisy optimization problem, enabling CMA-ES to leverage population statistics to filter stochastic noise and efficiently guide the search. The following sections provide a detailed explanation of the proposed approach.

A. Problem Formulation

Consider a pre-trained vision transformer model with L compressible layers. For a specific layer l, let $W_l \in \mathbb{R}^{m\times n}$ denote its weight matrix. To reduce redundancy, low-rank decomposition approximates W_l by factorizing it into the product of two lower-dimensional matrices. The truncated SVD is employed to achieve this approximation:

$$W_l \approx U_r \Sigma_r V_r^T = A_l B_l, \tag{1}$$

where $U_r \in \mathbb{R}^{m\times r}$ and $V_r \in \mathbb{R}^{n\times r}$ represent the matrices of the top-r left and right singular vectors, respectively, and $\Sigma_r \in \mathbb{R}^{r\times r}$ is the diagonal matrix of the top-r singular values ($r < \min(m, n)$). To implement this decomposition, the original matrix is replaced by two low-rank factors: $A_l = U_r\Sigma_r \in \mathbb{R}^{m\times r}$ and $B_l = V_r^T \in \mathbb{R}^{r\times n}$. This reduces the parameter count of the layer from mn to $r(m + n)$.

Since directly optimizing the discrete integer rank r is ill-suited for continuous evolutionary strategies, we propose a continuous relaxation of the search space. Instead of the integer rank, the decision variable is reformulated as the continuous compression ratio $c_l \in [0, 1]$, which represents the fraction of parameters removed from layer l. The relationship between the rank r_l and the compression ratio c_l is defined by

$$c_l = 1 - \frac{r_l(m + n)}{mn}. \tag{2}$$

The proposed framework adopts the continuous variable vector $\boldsymbol{c} = [c_1, c_2, \ldots, c_L]$ as the evolutionary representation for each candidate solution, where L is the total number of compressible layers. To map a continuous candidate solution back to a discrete rank for model instantiation, we apply the following transformation:

$$r_l = \text{Round}\left((1 - c_l)\frac{mn}{m + n}\right). \tag{3}$$

This formulation transforms the search space into a L-dimensional unit hypercube $\mathcal{C} = [0, 1]^L$, allowing the exploitation of fine-grained correlations between layers.

The goal is to identify the optimal configuration $\boldsymbol{c}^*$ that maximizes the model's validation accuracy while satisfying a global resource constraint, such as a target overall parameter count or compression rate. Formally, this objective is formulated as a constrained optimization problem, i.e.,

$$\boldsymbol{c}^* = \arg\max_{\boldsymbol{c}\in[0,1]^L} \mathcal{A}_{\text{val}}\left(f(\boldsymbol{x}; \Theta(\boldsymbol{c}))\right), \tag{4}$$

$$\text{s.t. } \mathcal{G}(\boldsymbol{c}) \geq \rho_{\text{target}} \tag{5}$$

where

- $\boldsymbol{x}$ represents the input data sampled from the validation set;
- $\Theta(\boldsymbol{c})$ denotes the model parameters reconstructed from the rank configuration derived from $\boldsymbol{c}$;
- $f(\cdot; \Theta(\boldsymbol{c}))$ denotes the inference function of the compressed model parameterized by $\Theta(\boldsymbol{c})$;
- $\mathcal{A}_{\text{val}}(\cdot)$ represents the accuracy evaluated on the validation set;
- $\mathcal{G}(\boldsymbol{c})$ calculates the global compression ratio of the candidate configuration;
- ρ_{target} is the predefined resource constraint.

To address the computational bottleneck associated with full-dataset validation, this study introduces a stochastic minibatch estimator $\hat{\mathcal{A}}_{\text{batch}}$ to provide a rapid approximation of the objective.

B. Efficient Minibatch Evaluation Strategy

Since evolutionary algorithms typically involve thousands of function evaluations, relying on full-dataset accuracy renders the search process infeasible. To overcome this inefficiency, CERES adopts a minibatch evaluation strategy that leverages stochastic estimation. This formulation enables rapid fitness approximation, significantly reducing the computational burden and facilitating practical exploration of the high-dimensional search space.

1) Minibatch Loss with Knowledge Distillation: To enable efficient continuous optimization, we must first substitute the discrete validation accuracy with a differentiable objective. Furthermore, since minibatch sampling introduces noise, the objective function must be robust enough to guide the search. To achieve this, we incorporate a knowledge distillation (KD) [6] term.

The maximization of accuracy is formulated as the minimization of an expected loss function. The uncompressed pre-trained model serves as the teacher, guiding the compressed student model. The comprehensive objective function is defined as a weighted sum of the original task loss and the distillation loss:

$$\mathcal{L}_{\text{total}}(\boldsymbol{c}) = (1 - \alpha_{\text{KD}})\mathcal{L}_{\text{orig}}(y, f(x; \Theta(\boldsymbol{c})))+ \alpha_{\text{KD}}\mathcal{L}_{\text{KD}}(f(x; \Theta(0)), f(x; \Theta(\boldsymbol{c}))), \tag{6}$$

where

- $f(x; \Theta(\boldsymbol{c}))$ represents the output logits of the compressed student model;
- $f(x; \Theta(0))$ represents the output logits of the uncompressed teacher model;
- y denotes the ground-truth label corresponding to the input x;
- $\mathcal{L}_{\text{orig}}$ is the standard Cross-Entropy loss;
- $\mathcal{L}_{\text{KD}}$ is the Kullback-Leibler divergence between the softened logits of the teacher and the compressed student;
- $\alpha_{\text{KD}} \in [0, 1]$ is the balancing coefficient that controls the strength of the distillation regularization.

Crucially, the inclusion of the distillation term ($\alpha_{\text{KD}} > 0$) acts as a regularizer that smooths the non-convex optimization landscape, mitigating the stochastic noise inherent in minibatch sampling and facilitating more stable convergence for the evolutionary solver.

2) Stochastic Minibatch Estimation and Noise: Even with the loss formulation, computing the expected value $\mathbb{E}_{(x,y)\in D}[\mathcal{L}_{\text{total}}]$ over the entire dataset remains computationally expensive. To achieve the necessary efficiency for evo-

lutionary search, we approximate the expectation using a randomly sampled minibatch D_{B}:

$$J(\boldsymbol{c}) = \frac{1}{|D_{\mathrm{B}}|} \sum_{(x,y)\in D_{\mathrm{B}}} \mathcal{L}_{\text{total}}(\boldsymbol{c}; x, y), \tag{7}$$

This minibatch strategy introduces a critical trade-off between computational efficiency and estimation precision. A larger batch size $|D_{\mathrm{B}}|$ yields a more accurate estimate but increases the computational cost per evaluation. Conversely, a smaller batch size drastically accelerates the search but introduces significant stochastic noise.

This stochasticity transforms the original problem into a noisy optimization problem. In such a landscape, a candidate solution might appear better simply due to a favorable data sample, rather than true architectural superiority. Therefore, ensuring stable convergence requires an optimization approach capable of distinguishing true performance improvements from stochastic fluctuations. While minibatch optimization offers massive speedups, it necessitates a solver that is inherently robust to stochastic variance, motivating the adoption of the evolution strategy component of CERES, as detailed in the subsequent section.

C. Evolutionary Optimization Framework

In light of the continuous nature of our formulated search space and the stochastic noise introduced by the high-efficiency minibatch evaluation, the choice of optimization algorithm is crucial. While GA has been widely used in NAS, this study employs CMA-ES as the core optimizer for its superior capability to exploit correlations in continuous domains and its inherent robustness against noise. Figure 1 illustrates the overall workflow of the proposed CERES framework.

At the beginning of each generation g, a population of candidate solutions is sampled from the current multivariate normal distribution defined by the mean vector, step-size, and covariance matrix. These continuous vectors are then mapped to the valid range $[0, 1]^L$ to represent feasible compression ratios. To ensure a fair comparison among these candidates, it evaluates the entire population using a single, identical minibatch $D_{\mathrm{B}}^{(g)}$ sampled for the current generation.

In addition to evaluating new candidates, we must address the shifting nature of the data distribution caused by minibatch sampling. Since the minibatch changes every generation ($D_{\mathrm{B}}^{(g)} \neq D_{\mathrm{B}}^{(g-1)}$), the fitness value of the best solution from the previous generation is no longer directly comparable to the current population. To prevent the algorithm from overfitting to noise, we perform elite re-evaluation. The previous elite solution is re-evaluated on the current minibatch alongside the new offspring. Finally, based on the fitness rankings obtained from this robust evaluation process, CMA-ES updates its distribution parameters to guide the search toward high-performance regions for the next generation.

D. Constraint Handling

While the evolutionary optimization framework described above effectively navigates the search space, the optimization problem is still subject to the resource constraint defined in (4). Given that CMA-ES is inherently designed for unconstrained optimization, it is necessary to directly incorporate this restriction into the fitness evaluation.

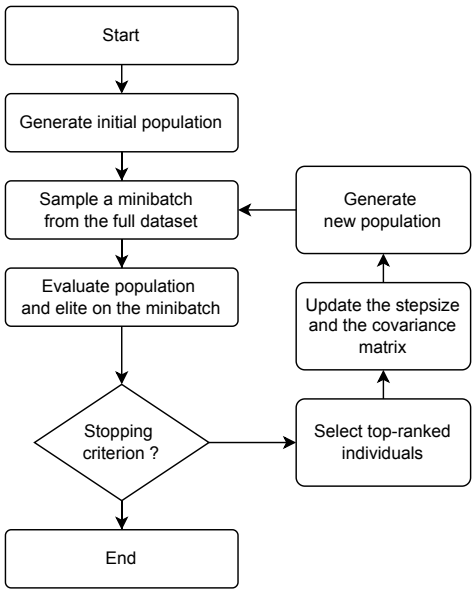

Figure 1. Workflow of the CERES framework. The algorithm integrates CMA-ES with a minibatch strategy. A shared batch $D_{\mathrm{B}}^{(g)}$ is sampled per generation to evaluate all candidates, ensuring fair comparison against stochastic noise and guiding the distribution update.

To address this issue, we transform the constrained problem into an unconstrained one via a penalty function approach. More specifically, we augment the fitness function with a regularization term that penalizes candidate solutions violating the target compression ratio. By employing an L1-Norm penalty for this purpose. This formulation provides a straightforward and effective mechanism to discourage infeasible solutions, ensuring that the evolutionary search is guided towards architectures that satisfy the deployment constraints. The final fitness function for a candidate $\boldsymbol{c}$ is thus formulated as

$$\mathcal{F}(\boldsymbol{c}) = J(\boldsymbol{c}) + \alpha_{reg} \cdot |\min(0, \mathcal{G}(\boldsymbol{c}) - \rho_{\text{target}})|, \tag{8}$$

where α_{reg} is a regularization coefficient.

IV. Experimental Results

This section first introduces the dataset and the parameter settings for the vision transformer architecture. Next, the experimental setup is described, including the reproduction of the baseline method, FLORA, and the configuration of the proposed CERES framework. Finally, the results regarding search efficiency and optimization robustness are presented.

All experiments were conducted on an Intel Core Ultra 7 265K CPU and an NVIDIA GeForce RTX 5090 GPU. The target architecture is DeiT-B [21], which targets five layers (Query, Key, Value, and the two feed-forward network layers) across 12 blocks, forming a 60-dimensional search space under an identical compression ratio constraint of $\rho_{\text{target}} = 57\%$. Each experimental condition consists of 30 independent trials.

Table I
PERFORMANCE COMPARISON ACROSS DIFFERENT BATCH SIZES. RESULTS DEMONSTRATE THE MEAN ACCURACY WITH STANDARD DEVIATION AND THE TOTAL SEARCH TIME IN GPU HOURS.

Batch size	Accuracy (%)	Time (GPU hours)
32	80.44 ± 0.61	0.50
128	81.01 ± 0.16	0.79
512	81.22 ± 0.09	2.67

Table II
PERFORMANCE COMPARISON ACROSS DIFFERENT DISTILLATION WEIGHT α_{KD}. RESULTS DEMONSTRATE THE MEAN ACCURACY WITH STANDARD DEVIATION.

Value of α_{KD}	Accuracy (%)
0.0	80.97 ± 0.13
0.5	81.01 ± 0.16
1.0	80.74 ± 0.14

To evaluate statistical significance, we performed comparisons using the Wilcoxon signed-rank test with Holm-Bonferroni p-value correction at a confidence level of $\alpha = 0.05$.

A. Experimental Settings

Prior to the comparative evaluation, we conducted a preliminary analysis to empirically determine the optimal settings for the batch size and the distillation weight α_{KD}. These values were subsequently fixed for all main experiments. First, for the batch optimization strategy, we investigated the impact of batch size on search accuracy and computational cost, with results detailed in Table I. The empirical results reveal a diminishing return in performance relative to computational time. Although the largest batch size yields the highest accuracy, it imposes a prohibitive computational burden. Conversely, the smallest batch size minimizes the time cost but results in a noticeable degradation in performance. Therefore, we determined that a batch size of 128 offers the optimal trade-off. This configuration achieves competitive accuracy comparable to the largest setting while requiring significantly less computational time, thereby balancing sample diversity with search efficiency.

Second, in Section III-B1 we introduced a knowledge distillation term weighted by α_{KD}. The results presented in Table II indicate that $\alpha_{\text{KD}} = 0.5$ yields the superior performance. It is worth noting that the performance differences among the tested non-zero values were not statistically significant, suggesting that CERES is relatively robust to this hyperparameter. However, to ensure consistent landscape smoothing and stability, we standardized $\alpha_{\text{KD}} = 0.5$ for all subsequent experiments.

To isolate the contribution of our evolutionary solver, we also introduce a comparative variant named CERGA. In this variant, the CMA-ES component is replaced by a standard GA, while retaining the same continuous search framework. The full list of parameter settings for CERES, CERGA, and the FLORA baseline is detailed in Table III.

B. Results and Discussions

1) Comparison with FLORA: First, we investigated the performance of the proposed CERES in comparison with the state-of-the-art method, FLORA. Table IV summarizes the quantitative results. As indicated in the table, CERES achieves a Top-1 validation accuracy of 81.01%, comparable to the FLORA baseline of 81.22%, confirming that our approach maintains model quality.

Notably, the results highlight a critical capability in constraint satisfaction. Under the target constraint of $\rho_{\text{target}} \geq 57\%$, CERES successfully meets the requirement with a compression ratio of 57.44% ± 0.19%. In contrast, FLORA yields a ratio of 56.35% ± 0.04%, failing to meet the strict target. This discrepancy likely stems from the coarse granularity of FLORA's discrete search space, which prohibits precise alignment with the boundary condition. Crucially, despite CERES compressing the model more aggressively to satisfy the constraint, it maintains an accuracy comparable to FLORA, demonstrating that our continuous optimization framework can adhere to strict resource budgets without sacrificing representational power.

Regarding computational efficiency, CERES demonstrates a substantial advantage. The results show that our method achieves a speedup factor of $11.97\times$ compared to FLORA. This observed speedup is particularly significant when contextualized with the fundamental differences in search space formulation. FLORA simplifies the optimization task by enforcing a shared rank for the Query, Key, and Value projections, resulting in only 3 learnable ranks per block and a total problem dimensionality of $3 \times 12 = 36$. Furthermore, it restricts rank selection to a coarse discrete grid, defined as $\boldsymbol{c} \in \{0.125 \cdot t \cdot R_{\text{max}} \mid t \in \{1, \ldots, 7\}\}^{36}$, where R_{max} denotes the maximum rank capacity. While this search space is vast ($7^{36} \approx 10^{30}$), it is inherently limited by its tied parameters and discrete steps.

In contrast, CERES framework relaxes these structural constraints. It treats all five layers in each block independently, leading to a higher-dimensional search space of $5 \times 12 = 60$. We project this problem into a fully continuous search space $\boldsymbol{c} \in [0, 1]^{60}$. This formulation offers infinite resolution and layer-wise independence, allowing the CMA-ES optimizer to fine-tune compression ratios with arbitrary precision. This precise control is reflected in the superior compression ratio observed in Table IV. Unlike FLORA, which is restricted to coarse discrete steps, CERES can exploit the continuous spectrum to locate efficient solutions that lie between fixed quantization points.

Exploring such an exponentially larger and continuous space would incur a higher computational cost compared to the simplified discrete baseline. However, the proposed batch optimization strategy effectively addresses this issue. By leveraging stochastic minibatch evaluation, CERES eliminates the bottleneck of full-dataset processing required by traditional methods. This allows our framework to enjoy the generality of a continuous search space while simultaneously outperforming

Table III
PARAMETER SETTINGS FOR CERES, CERGA, AND FLORA.

	FLORA	CERES (ours)	CERGA (ours)
Crossover	Uniform	Weighted intermediate	Two-point
Mutation	Uniform	Gaussian	Random resetting
Parent selection	10-tournament	Top $\mu = 16$	3-tournament
Survivor selection	(μ, λ)	(μ, λ)	(μ, λ)
Population size	50	64	64
Crossover rate	1.0	–	0.8
Mutation rate	0.2	$\sigma = 1/6$	0.2
#generations	20	500	500

Table IV
QUANTITATIVE COMPARISON WITH THE STATE-OF-THE-ART METHOD. RESULTS DEMONSTRATE THE TOP-1 VALIDATION ACCURACY, COMPRESSION RATIO, AND TOTAL SEARCH TIME.

Algorithm	Accuracy (%)	Compression Ratio (%)	Time (GPU hours)	Speedup
FLORA	81.22 ± 0.28	56.35 ± 0.04	9.47	1.0×
CERES (ours)	81.01 ± 0.16	57.44 ± 0.19	0.79	11.97×
CERGA (ours)	77.89 ± 0.15	57.59 ± 0.75	0.82	11.57×

the discrete baseline in terms of search speed.

2) Robustness Analysis: A critical observation from our experiments was the distinct performance gap between different optimization strategies. Specifically, we noted that the GA failed to maintain stability under stochastic evaluation, resulting in degraded model accuracy. To strictly validate the necessity of employing CMA-ES, we performed a comparative ablation study against the GA-based variant (CERGA). The quantitative results in Table IV confirm this disparity. As indicated, CERES achieves a mean validation accuracy of 81.01%, significantly surpassing the 77.89% obtained by CERGA.

More critically, the stability of the two algorithms differs substantially. CERES maintains a low standard deviation of $\pm 0.16\%$, whereas CERGA exhibits a much higher variance of $\pm 1.47\%$. This contrast is visually corroborated in Figure 2, which depicts the evolutionary trajectory of the validation accuracy. While CERES demonstrates a consistent upward trend enclosed within a narrow error band, the CERGA trajectory is surrounded by a wide shaded region, indicating severe fluctuations and unreliability across independent runs.

The instability and inferior performance observed in CERGA can be attributed to the interaction between its discrete genetic operators and the stochastic noise inherent in minibatch evaluation. Standard GA relies on point-wise selection mechanisms, which are susceptible to transient sampling bias. In this scenario, individuals may receive inflated fitness scores due to favorable data sampling, often referred to as lucky minibatches, rather than genuine architectural quality. This erroneous selection pressure causes the population to drift away from the global optimum, as evidenced by the substantial variance previously illustrated in Figure 2. In contrast, CMA-ES updates a multivariate normal distribution based on the aggregate statistics of the population. This statistical approach effectively acts as a low-pass noise filter, smoothing out the high-frequency variance introduced by stochastic minibatch

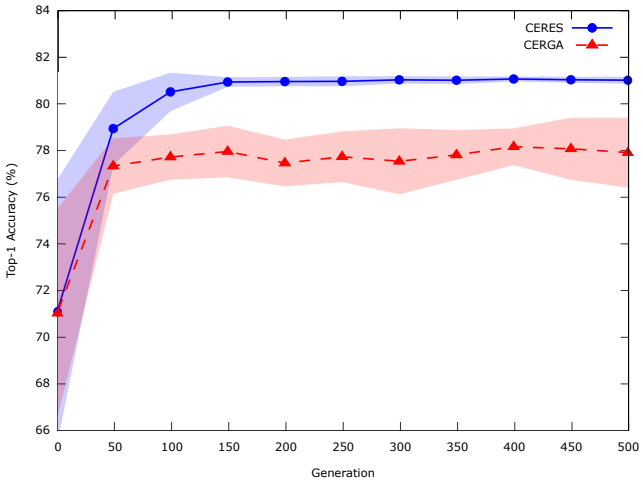

Figure 2. Progress of Top-1 validation accuracy against the number of generations for CERES and CERGA. The CERES demonstrates superior convergence efficiency and higher stability compared to the GA variant.

sampling. Hence, CERES is capable of precise and robust convergence even in a highly noisy optimization environment, validating the suitability of the covariance matrix adaptation mechanism for our batch-based framework.

V. CONCLUSIONS

This study proposed CERES, a novel evolutionary framework designed to automate the rank selection process for vision transformer compression. By reformulating rank selection as a continuous optimization problem and introducing a stochastic minibatch evaluation strategy, CERES effectively addresses the computational bottlenecks and search space limitations inherent in prior discrete NAS-based methods.

The extensive experiments demonstrated that employing the CMA-ES is essential for navigating the noisy landscape

introduced by rapid minibatch estimation. While standard GA struggled with stability under stochastic noise, CMA-ES successfully leveraged population statistics to ensure robust convergence. Empirical results on ImageNet show that CERES achieves a competitive trade-off between model size and accuracy while delivering an approximate $12\times$ speedup compared to the state-of-the-art FLORA framework. Collectively, these findings establish CERES as a highly efficient and scalable solution for low-rank model compression, bridging the gap between evolutionary optimization and resource-constrained deep learning applications.

Looking forward, we plan to extend CERES from a constrained optimization setup to a multi-objective optimization framework. While the current approach treats the compression rate as a hard constraint, formulating it as a simultaneous objective alongside accuracy would allow for the discovery of a Pareto front of efficient architectures. This evolution could provide greater flexibility for deploying models across diverse resource environments. Furthermore, applying this continuous strategy to larger foundation models remains a promising direction for future research.

REFERENCES

[1] P. C. Hansen, "Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank," *SIAM Journal on Scientific and Statistical Computing*, vol. 11, no. 3, pp. 503–518, 1990.

[2] C.-C. Chang, Y.-Y. Sung, S. Yu, N.-C. Huang, D. Marculescu, and K.-C. Wu, "Flora: Fine-grained low-rank architecture search for vision transformer," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 2482–2491.

[3] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[4] N. Tyche, A. Taylor, J. Evans, and M. Reid, "Improving neural network efficiency through advanced pruning techniques," *Authorea Preprints*, 2024.

[5] A. Polino, R. Pascanu, and D.-A. Alistarh, "Model compression via distillation and quantization," in *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[6] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[7] M. Thoma, J. Villasante, E. Aghajanzadeh, S. B. Sampath, P. Mori, M. Groetzinger, D. Dylkin, M.-R. Vemparala, N. Fasfous, A. Frickenstein *et al.*, "Flar-svd: Fast and latency-aware singular value decomposition for model compression," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1898–1907.

[8] Y. Ji, Y. Xiang, J. Li, W. Chen, Z. Liu, K. Chen, and M. Zhang, "Feature-based low-rank compression of large language models via bayesian optimization," *CoRR*, 2024.

[9] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, "A comprehensive survey of neural architecture search: Challenges and solutions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–34, 2021.

[10] K. T. Chitty-Venkata, M. Emani, V. Vishwanath, and A. K. Somani, "Neural architecture search for transformers: A survey," *IEEE Access*, vol. 10, pp. 108 374–108 412, 2022.

[11] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Smash: one-shot model architecture search through hypernetworks," *arXiv preprint arXiv:1708.05344*, 2017.

[12] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan, "A survey on evolutionary neural architecture search," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 2, pp. 550–570, 2021.

[13] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[14] G. Morse and K. O. Stanley, "Simple evolutionary optimization can rival stochastic gradient descent in neural networks," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, 2016, pp. 477–484.

[15] R. E. Smith, B. A. Dike, and S. Stegmann, "Fitness inheritance in genetic algorithms," in *Proceedings of the 1995 ACM symposium on Applied computing*, 1995, pp. 345–350.

[16] D. V. Arnold, *Noisy Optimization with Evolution Strategies*. Springer Science & Business Media, 2002, vol. 8.

[17] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[18] K. Lenc, E. Elsen, T. Schaul, and K. Simonyan, "Non-differentiable supervised learning with evolution strategies and hybrid methods," *arXiv preprint arXiv:1906.03139*, 2019.

[19] N. Sinha and K.-W. Chen, "Neural architecture search using covariance matrix adaptation evolution strategy," *Evolutionary Computation*, vol. 32, no. 2, pp. 177–204, 2024.

[20] Z. Yang, Y. Wang, X. Chen, B. Shi, C. Xu, C. Xu, Q. Tian, and C. Xu, "Cars: Continuous evolution for efficient neural architecture search," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1829–1838.

[21] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the International Conference on Machine Learning*, 2021, pp. 10 347–10 357.