Neural Architecture Transfer 2: A Paradigm for Improving Efficiency in Multi-Objective Neural Architecture Search

Simone Sarti, Eugenio Lomurno, and Matteo Matteucci

Department of Electronics, Information, and Bioengineering
Politecnico di Milano, Milan, Italy
{simone.sarti, eugenio.lomurno, matteo.matteucci}@polimi.it

Abstract. The advent of deep learning has had a significant impact on various sectors of modern society, with artificial neural networks becoming the leading models for tackling a wide range of challenges. The innovation of Neural Architecture Search (NAS) methods, which facilitate the automated creation of optimal neural networks, marks a significant step forward in this field. However, the large computational resources and time required for NAS processes are significant limitations. To address these challenges, Once-For-All (OFA) and its advanced version, Once-For-All-2 (OFAv2), were introduced to develop a single, comprehensive super-network capable of efficiently deriving specific sub-networks without the need for retraining, thereby maintaining stellar performance under varying constraints. Building on this, Neural Architecture Transfer (NAT) was developed to improve the efficiency of extracting such sub-networks from the overarching super-network. This study introduces Neural Architecture Transfer 2 (NAT2), an evolution of NAT that refines the multi-objective search mechanisms within dynamic super-networks to further improve the performance-complexity trade-off for the searched architectures. Leveraging the advances of OFAv2, NAT2 introduces significant qualitative improvements in the sub-networks that can be extracted by incorporating novel policies for network initialisation, pre-processing, and archive updates, as well as a fine-tuning based post-processing pipeline. The empirical evidence presented here highlights the effectiveness of NAT2 over its predecessor, particularly in the development of high-performance architectures with a reduced number of parameters and multiply-accumulate operations.

Keywords: Neural Architecture Transfer 2 · Neural Architecture Search · Super-Network · Sub-Network · Multi-Objective Optimisation

1 Introduction

Deep learning has revolutionised many fields through the use of artificial neural networks, which excel at identifying complex patterns without manual feature engineering. These networks outperform traditional methods in various tasks due

to their advanced layered architecture, which facilitates superior feature extraction. Despite these advances, the increasing size of models poses challenges [8]. Neural Architecture Search (NAS) is emerging as a solution to automate the discovery of optimal neural architectures, minimising the need for domain expertise while catering to specific datasets and tasks [18]. NAS has evolved to balance computational efficiency with model performance, with an emphasis on reducing the complexity of the search process, energy consumption and consequently CO_2 emissions. It seeks an optimal balance between model accuracy and computational requirements, including parameters and operations, especially on memory-constrained devices [7].

The Once-For-All (OFA) method marked a breakthrough by minimising computational requirements through a versatile super-network that allows the extraction of efficient sub-networks tailored to different constraints, without compromising performance [1]. This approach was further enhanced by Once-For-All-2 (OFAv2), which expanded the search space using state-of-the-art neural network design techniques, thereby enhancing the capabilities of the super-network [15]. Focusing on the extraction of sub-networks, the Neural Architecture Transfer (NAT) algorithm was developed to optimise this process through knowledge transfer and adaptation from pre-trained super-networks, combining transfer learning with multi-objective evolutionary search [11].

This paper introduces Neural Architecture Transfer 2 (NAT2), which advances NAT by optimising multi-objective search algorithms for dynamic super-network architectures. NAT2 uses OFAv2-generated super-networks and incorporates architectural enhancements such as parallel blocks, dense skip connections, and early exits as search space. Improvements include new initialisation, pre-processing and update policies, as well as a novel coding scheme and improved prediction models. An optional post-processing tuning stage is also introduced to further refine model performance with minimal additional parameters and MACs. NAT2 not only outperforms NAT in terms of accuracy, but also achieves these gains with fewer parameters and MACs, underscoring its goal of creating architectures that are exceptionally lightweight in terms of parameters and computational resources. Figure 1 gives an overview of the NAT2 methodology.

The manuscript proceeds as follows: Section 2 reviews NAS and significant work in image classification, detailing their design motivations. Section 3 explains the NAT2 methodology, highlighting the novel contributions. Experimental setups, results and comparisons are presented in section 4. Section 5 concludes the paper by summarising its main contributions.

2 Related Works

Neural Architecture Search (NAS) remains a vibrant area within deep learning, bridging machine learning techniques with optimisation to automate the design of complex neural networks. Despite its acclaim, NAS suffers from a lack of standardised methodologies due to its diverse techniques. Elsken *et al.* classifies NAS algorithms based on three main components: the search space, the

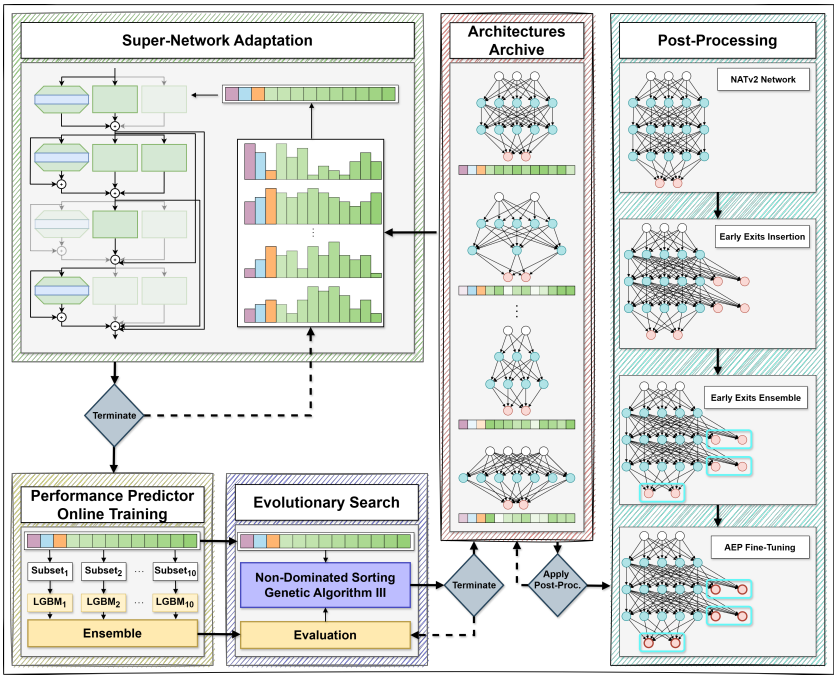

Fig. 1. The NAT2 summary diagram. The proposed algorithm designs customised architectures from a very large search space of possible state-of-the-art configurations. Multi-objective optimisation extends the work of NAT with new encoding and super-networks management techniques. New predictors provide accurate estimates for efficient evolutionary search. Once the optimal sub-network has been extracted, it can be further refined by an additional post-processing step for fine-tuning early exits neural networks.

search strategy, and the performance evaluation strategy, laying a foundation for understanding the multifaceted nature of NAS [2].

Early NAS efforts, such as NASNet, demonstrated the potential to create models that outperform human-designed counterparts in tasks such as image classification by employing a cell-based search approach, albeit at considerable computational cost [18]. To mitigate these challenges, PNAS introduced a sequential model-based optimisation strategy that uses a predictor to efficiently guide the architecture search process, thereby reducing computational load and time [6]. Subsequent developments led to the POPNAS series, which refined the cell-based methodology with predictors to estimate training times, allowing a shift towards multi-objective optimisation. This balanced search efficiency with architectural quality, effectively training networks along the Pareto front to optimise both search time and model performance [9, 4, 3].

DARTS further improved search efficiency by introducing a continuous search space, represented as a super-network. This allowed simultaneous optimisation of model weights and structure by gradient descent, culminating in the extraction of a subgraph as the final architecture [7]. Lyu *et al.* introduced a multi-objective

evolutionary algorithm with a probability stack (MOEA-PS) that focuses on optimising both accuracy and time efficiency in the design of deep neural networks. MOEA-PS incorporates a novel representation and genetic operation control mechanism to efficiently generate architectures [12]. Ma *et al.* proposed a novel classifier-based Pareto evolution approach to address the rank disorder issue in multi-objective evolutionary NAS. By transforming the NAS process into a classification task, the authors simplify the search for optimal architectures. The approach leverages an online classifier for predicting dominance relationships, coupled with adaptive clustering for reference architecture selection and an α-domination method to balance sample distribution [13].

2.1 Once-For-All

Once-For-All (OFA) represents a paradigm shift in Neural Architecture Search (NAS) by introducing a super-network capable of adapting to different architectural configurations without the need for retraining. This innovation addresses the computational inefficiencies of traditional NAS methods and significantly reduces the carbon footprint associated with extensive model training processes. OFA employs a novel Progressive Shrinking (PS) strategy, a comprehensive method that goes beyond simple pruning by dynamically adjusting model dimensions. This approach not only diversifies the architectural space, but also ensures optimal performance on a variety of hardware platforms with different constraints [1]. The core of OFA's methodology lies in its efficient training regime, which uses PS to sequentially refine the super-network. Starting with the most extensive network configuration, it gradually introduces constraints, allowing a wide range of sub-networks to be trained. This strategy is advantageous because it ensures that smaller networks inherit the most important features from their larger counterparts, thereby maintaining high levels of accuracy. The effectiveness of the PS algorithm is evident in its application across different hardware, where OFA consistently outperforms state-of-the-art NAS methods, offering significant improvements in latency and accuracy with significantly lower environmental impact. Significantly, OFA's approach to model deployment revolutionises NAS by separating the training phase from the architecture search, thereby eliminating the training overhead for model specialisation. This separation enables rapid deployment across multiple scenarios without additional computational cost. The flexibility and efficiency of the architecture in training and deployment underscores OFA's pivotal role in advancing NAS towards more sustainable and adaptive solutions [1].

2.2 Once-For-All-2

Once-For-All-2 (OFAv2) advances its predecessor by integrating architectural improvements and a refined training methodology to construct a super-network capable of efficient deployment across diverse hardware configurations. This evolution maintains environmental sustainability while increasing performance [15]. In particular, OFAv2 incorporates early exits, parallel blocks and dense skip links into the OFAMobileNetV3 architecture. These features not only increase

the adaptability and performance of the super-network, but also facilitate the training of more diverse and efficient sub-networks. The Extended Progressive Shrinking (EPS) algorithm introduces Elastic Level and Elastic Exit steps tailored to the new architectural elements. Elastic Level allows the dynamic selection of parallel blocks within levels, thereby increasing architectural diversity. Elastic Exit, on the other hand, allows the training of sub-networks to be terminated at different stages, optimising efficiency and performance. Together, these steps enable OFAv2 to improve accuracy over OFA while maintaining its flexibility and environmental benefits. In addition, OFAv2's novel approach to teacher network extraction dynamically updates the teacher network after each EPS step, ensuring that the most relevant and up-to-date knowledge is transferred to subsequent training steps. This strategy, in contrast to OFA's static teacher network, provides a more nuanced and effective knowledge distillation process [15].

2.3 Neural Architecture Transfer

Neural Architecture Transfer (NAT) is presented as a NAS technique with an innovative focus on the rapid generation of task-specific models through a strategic combination of online transfer learning and multi-objective evolutionary search [11]. At the heart of NAT's efficiency is the use of a pre-trained super-network, based on the OFAMobileNetV3 search space, which enables skilful exploration of architectural designs. This methodology facilitates rapid adaptation of architectures to different hardware specifications or objectives without the need for re-training, overcoming the computational and time constraints of traditional NAS methodologies. NAT's unique approach is to selectively tune the sub-networks of the super-network that are expected to align with the efficient trade-off frontier for the targeted data set, rather than indiscriminately tuning all possible sub-networks. This focused adaptation not only conserves computational resources, but also ensures that the generated architectures are highly optimised for the given objectives. Demonstrated on a wide range of image classification tasks, including those with limited data availability, NAT has proven its ability to efficiently deliver models that meet and frequently exceed the state-of-the-art in mobile environments, while exploiting the comprehensive and versatile search space provided by OFAMobileNetV3 [11].

3 Method

NAT2, the proposed advanced version of the Neural Architecture Transfer (NAT) algorithm, incorporates the Once-For-All-2 (OFAv2) technique for generating the initial super-network search space to generate architectures with high performance and lower complexity. It includes modifications to the original algorithm, sub-network sampling, and performance prediction methods. It introduces a preprocessing step for initialising a variety of efficient architectures, and a two-stage post-processing step for fine-tuning these architectures. Components not mentioned remain unchanged from the original version.

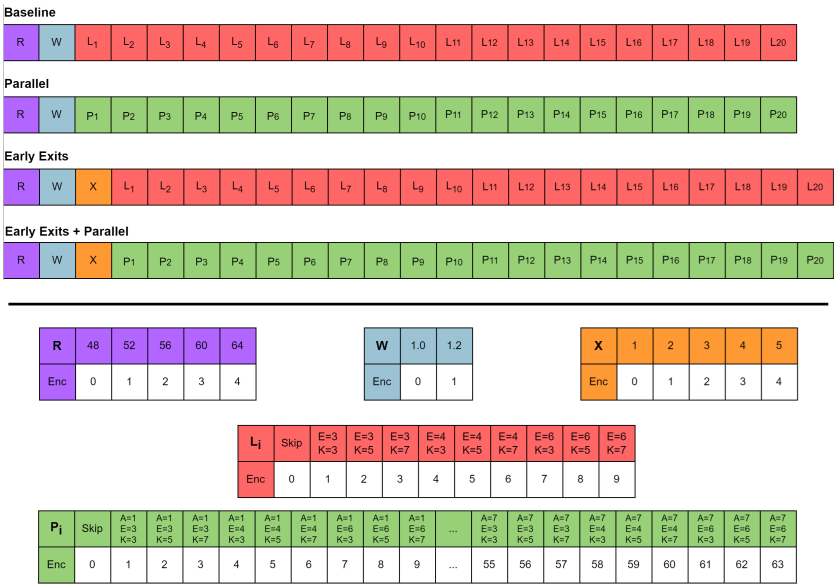

Fig. 2. The encodings representing the possible sub-networks within different types of super-networks have the following structure. R encodes the value corresponding to the size of the input images. W encodes the value of the width multiplier, which determines the width of the network architecture. X encodes information about the selected exit, specifically for super-networks that support early exits. L_i encodes the configuration of the i^{th} IRB/IB block for non-parallel networks. P_i encodes the configuration of the i^{th} level, i.e. the set of parallel blocks, for parallel networks.

3.1 Expanded Search Space

NAT2 introduces an advanced new encoding method to enhance evolutionary search within OFAv2 super-networks, refining the encoding technique used by NAT in OFAMobileNetV3. The original encoding, while fundamental, required refinement to cover a wider range of architectural possibilities. NAT encoding uses integer encoded strings of 22 elements, shown in Figure 2 as the "Baseline". This compact notation conveys specific details: the first value represents the resolution of the input image R, and the second specifies the width multiplier W, which is essential for adjusting filter sizes in the OFA framework. NAT2 continues the approach of generating two super-networks with width multipliers of 1.0 and 1.2, maintaining the core coding principles of NAT.

In the NAT framework, the 20 encoded values denote combinations of kernel size K and expansion ratio E for each of the 20 internal Inverted Residual Blocks (IRB) or Inverted Blocks (IB). NAT2 extends this encoding to parallel blocks within super-networks by converting these pairs to triplets by adding a new term A, which ranges from 1 to 7 to cover all permutations of parallel block activation states. In addition, a special value of 0 indicates the absence of the i^{th} block or level, effectively reducing the stage depth.Thus, in this extended framework, each of the 20 P_i values can signify up to 64 unique states, enriching the search space

without increasing the size of the encoding. This adaptation is illustrated in the second row of Figure 2, marked "Parallel".

In order to include early exits within super-networks, thus facilitating the extraction of sub-networks that can make inferences at intermediate stages, NAT2 introduces an additional variable X into the encoding strings, positioned third. This inclusion is shown in the third row of Figure 2, marked "Early Exits". The value of X corresponds to the index of the selected exit point, providing precise details of the selected early exit in the super-network structure. NAT2's extended encoding, shown in the fourth row of Figure 2 under "Early Exits + Parallel", combines the adjustments for parallel blocks and early exits. This method greatly expands the search space relative to the original NAT, with the minor concession of adding an extra value to the encoding sequence.

3.2 Archive Initialization and Update

NAT2 adopts a novel approach to managing the optimal sub-networks archive that affects two key phases of the NAT algorithm: archive initiation and expansion. During the initiation phase, NAT2 adopts a different sampling strategy from NAT, which samples architectures uniformly across the search space, inadvertently favouring networks with four levels of depth. This bias in NAT arises from the coding of skippable IRB blocks, where the use of 0 as a skipping mechanism disproportionately represents shallower levels. The complexity is compounded by the introduction of parallel blocks, which increases the encoding variations for each network level to 64. To address this, NAT2 prioritises uniformity in the sampling of sub-networks, not indiscriminately across the search space, but with a focus on level and network depth, as well as block configurations, including both parallel and non-parallel forms. This strategy encourages a more diverse range of architectures within the initial archive, thereby increasing the predictor's ability to generalise across a large training dataset.

NAT2 adopts a sub-network replacement strategy, eschewing the incremental addition of new sub-networks. Instead, NAT2 fills the archive to capacity from the beginning with a predefined maximum number of architectures. During each iteration, inferior architectures are replaced by superior ones from the evolutionary search, ensuring a constant archive size. This technique aims to improve the quality of the architectures in the archive, which subsequently improves the performance of the prediction model. This improvement is particularly noticeable in the early stages, providing a richer dataset for training the predictor, allowing for more accurate sub-network evaluations in NAT2.

In addition, NAT2 introduces a pre-processing phase during archive initialisation, which involves sampling a set of architectures, A_s, ten times the intended size of the archive, fixed at 300. After evaluating and comparing these sampled sub-networks, only the top $A_s - 2$ networks, including the maximum and minimum networks, make up the initial archive. This procedure of selecting high quality architectures from the start promotes better performance in subsequent sub-networks. However, this preprocessing step initially increases the execution time, a singular cost that depends on the volume of architectures initially sampled.

3.3 Performance Predictor

NAT2's evolutionary search process produces a large number of sub-networks, and evaluating the performance of each one individually, despite weight sharing, is computationally prohibitive. To overcome this, NAT2 uses a performance prediction model. This regression-based model is trained online using the encodings of the sub-networks in the archive and their top-1 accuracy as training data. This training allows the predictor to estimate the performance of unseen architectural encodings, making the evaluation of many sub-networks computationally feasible.

In NAT2, the range of potential prediction models has been significantly expanded. Initially, models such as Gaussian Process (GP), Radial Basis Function (RBF), Multilayer Perceptron (MLP), Classification and Regression Tree (CART) and Radial Basis Function Ensemble (RBFE) were considered. Further investigation led to the inclusion of Support Vector Regressor (SVR), Ridge Regressor, K-Nearest Neighbours Regressor (KNN) and Bayesian Ridge Regressor.

In addition, models such as End-to-End Random Forest-based Performance Predictor (E2EPP) [17], Light Gradient Boosting Machine (LGBM) [5] and Catboost [14] have been added to the list of candidate predictors based on their documented success. This wide range of machine learning models allows NAT2 to thoroughly investigate different regression mechanisms, highlighting the critical role of the predictor in the effectiveness of the algorithm.

3.4 Training Networks with Early Exits

The training regime for super-networks with early exits within NAT2 has been refined through the Anticipate Ensemble and Prune (AEP) methodology, addressing early exit orchestration alongside OFAv2 enhancements [16]. It primarily applies the AEP technique to multi-exit networks, using a weighted ensemble strategy for exits. This approach uses different weighting schemes to equitably adjust the influence of each exit, thus improving training results [16]. In addition, NAT2 assimilates ENS-KD, a knowledge distillation strategy from OFAv2 based on the AEP concept. Unlike conventional approaches that transfer knowledge to a student network only from the last layer of the teacher, ENS-KD uses data from all exits of the teacher network. By applying the weighting and aggregation of the AEP, it enables a more efficient knowledge distillation, resulting in an enhanced performance of the student network.

During the initial training phases of NAT2 super-networks with early exits, the maximal networks extracted from the OFAv2 super-network, particularly those with width multipliers of 1.0 and 1.2, undergo AEP training following the DESC weighting strategy [16]. This is possible because the maximal network in an OFAv2 super-network with early exits includes all exits from the super-network. For the adaptation phase, where the super-network is fine-tuned by sequentially activating sub-networks, NAT2 uses a training algorithm similar to the final phase of the Extended Progressive Shrinking (EPS) algorithm [15]. However, whereas EPS gradually reveals elastic parameters and their values, NAT2 immediately makes all elastic parameter values available for sampling.

Table 1. The details of the datasets used in this work in terms of number of classes and splits.

Dataset	Classes	Train size	Validation size	Test size
Tiny ImageNet	200	85000	15000	10000
CIFAR10	10	45000	5000	10000
CIFAR100	100	45000	5000	10000

In the adaptation phase, NAT2 differs from its predecessor by replacing standard knowledge distillation with the ENS-KD technique for early exit super-networks [15]. To align with the EPS training used in OFAv2, the sub-networks activated during this phase in NAT2 are configured as single exit networks.

3.5 Post-Processing

In NAT2's multi-objective optimisation, the most effective sub-networks excel across different objectives, but may not maximise their classification potential. To improve accuracy, two different fine-tuning post-processing methods are used. The first applies to single-exit super-networks and the second to early-exit super-networks, with both methods consisting of two steps. First, the optimal training time, e, for each sub-network is determined by fine-tuning on the target dataset and assessing the performance of the validation set. Subsequently, further fine-tuning for e epochs with both training and validation sets leads to an evaluation of the test classification performance. The difference between these methods lies in the fine-tuning approach. The first method simply fine-tunes single-exit networks generated by NAT2. The second, using the AEP technique, integrates all exits above the selected exit for combined fine-tuning [16], typically achieving higher accuracy with a marginal parameter and MACs increase.

4 Results and Discussion

The experiments aimed to demonstrate the efficiency of NAT2 in producing architectures with improved or equivalent performance compared to its predecessor, while reducing parameters and MACs. Both NAT and NAT2 models were pre-trained on the Tiny ImageNet dataset, following the parameter configurations from the original NAT study [11]. Architecture generation and validation was performed on the CIFAR10, CIFAR100 and Tiny ImageNet datasets, which characteristics related to splits and number of classes are summarized in Table 1.

At the end of each search, the top four architectures were selected for both NAT and NAT2. In the case of the search designed to maximise accuracy as a single objective, only the best performing architecture was returned. All the found architectures were trained using SGD with a momentum of 0.9 and a weight decay of $3 \cdot 10^{-4}$, starting with a learning rate of $2.5 \cdot 10^{-3}$, adjusted by a cosine annealing scheduler, and a batch size of 256. The warm-up periods used identical hyperparameters except for an initial learning rate of $7.5 \cdot 10^{-3}$.

During the post-processing phase, the architectures were trained with an initial learning rate equal to 10^{-4} and AdamW optimizer with weight decay set to

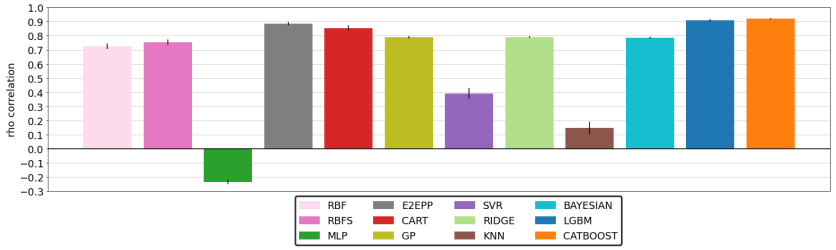

Fig. 3. The rho correlation achieved by the proposed accuracy predictors. The metric is evaluated via 10-fold cross-validation and then averaged.

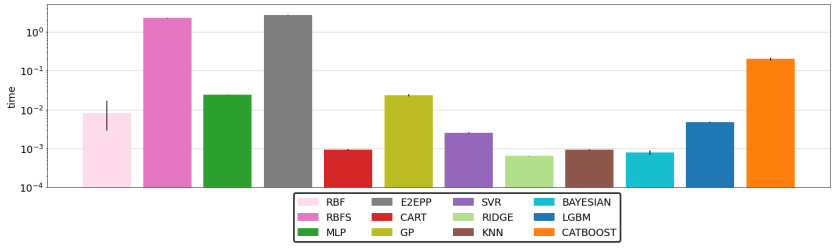

Fig. 4. The inference time achieved by the proposed accuracy predictors. The metric is evaluated via 10-fold cross-validation and then averaged.

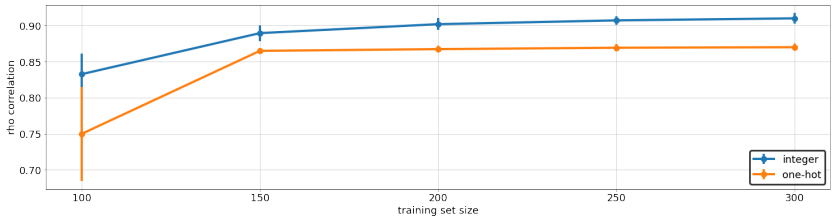

Fig. 5. The rho correlation values achieved by the LGBM accuracy predictor for different training set sizes and encodings.

$5x10^{-4}$ [10]. The batch size for these experiments was set to 64, and the networks were trained for a maximum of 150 epochs using a cosine annealing learning rate scheduler. Early stopping was used, with a patience value of 30 epochs based on validation loss. For the AEP strategy for early exit architectures, a uniform weight was used to balance the contribution of exits, as this gave better results on average. All models were implemented using PyTorch 1.12.1 and experiments were run on an NVIDIA Quadro RTX 6000 GPU.

4.1 Performance Predictors Analysis

In the first ablation study, predictor models were evaluated with a fixed training set size of 300, using the encodings from Section 3 for the input features. The performance of the predictor models was assessed using correlation values derived from the analysis of sub-networks of OFAv2 and OFA configurations. This study aimed to identify the most accurate predictor of sub-network accuracy, ap-

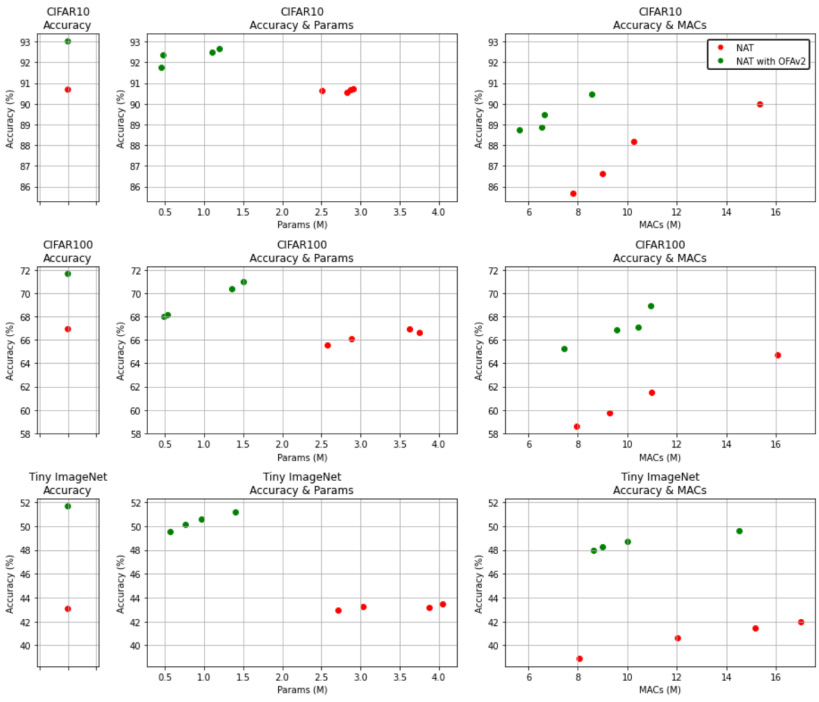

Fig. 6. The results of the study of the effectiveness of the OFAv2 super-networks compared to the OFA super-network within the NAT algorithm, based on the proposed datasets and optimisation strategies. For both sets of experiments, the encodings proposed in Section 3 were used.

plicable to both NAT and NAT2 evaluations. Through 10-fold cross-validation, models were evaluated and their performances averaged; the best performing ensemble was selected as the predictor. The comparison, shown in Figure 3, revealed CatBoost and LGBM as top performers with rho correlations above 0.9, indicating a strong relationship between encoding accuracy and network heterogeneity capture. The time efficiency analyses in Figure 4 highlighted the CART, RIDGE and BAYESIAN models as the fastest, albeit with lower performance. The balance between rho correlation and fitting time favoured LGBM over the slightly better but slower CatBoost. Further evaluations varied training set sizes and encoding methods, with integer encoding outperforming one-hot in performance and stability, as shown in Figure 5. These results, obtained with LGBM, suggest that there is an optimal number of samples beyond which improvements in rho correlation plateau. The switch from NAT's growing archive, with initial size of 100, to NAT2's fixed-size archive is validated by improved early search accuracy estimation, highlighting the effectiveness of the method.

4.2 OFAv2 Super-Networks Analysis

The second ablation study evaluates the effectiveness of the NAT algorithm, augmented with new coding methods and performance predictors. This preliminary analysis measures the improvements made by our modifications over the original model, focusing on performance shifts resulting from initial super-network modifications, comparing OFA-derived super-networks with those derived from OFAv2.

The study is summarised in Figure 6, which shows results across different datasets and optimisation objectives. The CIFAR10 results demonstrate improvements across all configurations with the OFAv2-derived super-network. Accuracy optimisation shows a 2% increase in accuracy with OFAv2 compared to OFA. Multi-objective search results highlight not only increased accuracy, but also significant reductions in parameters and MACs. For example, architectures optimised for fewer parameters show a 1% increase in accuracy and a fivefold reduction in parameters with OFAv2 compared to OFA.

Results from CIFAR100 and Tiny ImageNet, more complex datasets, support these findings. Accuracy-focused optimisations on these datasets show approximately 5% and 9% improvements with OFAv2. Results from multi-objective optimisations confirm the increasing effectiveness of the method with increasing problem complexity. These results also highlight the effectiveness of the new encoding method in capturing complexity and improving model performance.

4.3 Final Results

The concluding experiments aim to juxtapose the NAT baseline model with its advanced version proposed in this paper, NAT2, evaluating both with and without the application of the post-processing step. The results of these experiments are presented in Table 2, which includes the evaluation on the CIFAR10, CIFAR100 and Tiny ImageNet datasets. The results are reported in terms of top-1 accuracy, number of parameters and number of multiply-accumulate (MAC) operations. For each multi-objective search, the best architecture in terms of accuracy and the best architecture in terms of the secondary objective are displayed.

In the CIFAR10 dataset, NAT2 with Post-Processing (NAT2 + PP) stands out with an accuracy of 93.17%, surpassing other configurations. However, this model is the most resource-intensive, with 12.42 million parameters and 77.82 million MACs. Striking a balance between accuracy and resource efficiency, NAT2's "Accuracy Params" strategy yields a model with just 0.27M parameters and 91.46% accuracy, outperforming all NAT-based models. Similarly, the "Accuracy MACs" optimisation results in a model of unparalleled efficiency, achieving 89.67% accuracy with only 6.35M MACs and 0.19M parameters, an advantageous option for highly constrained devices. In various optimisation scenarios, NAT2 consistently outperforms the original NAT, demonstrating the beneficial impact of post-processing without significantly compromising model efficiency or accuracy.

Table 2. The results extracted from the final set of experiments. For each dataset, the first column on the left shows the best sub-networks found, grouped by research objective. For each dataset and metric, the best result is highlighted in **bold**. For each dataset, metric and objective, the best result is underlined. For each multi-objective optimisation, the best model found for accuracy, and the best model for the second objective of the optimisation are reported. Each NAT2 experiment is also presented in its post-processed form called NAT2 + PP.

	Objective	Model	Accuracy	Params (M)	MACs (M)
CIFAR10	Accuracy	NAT	90.68	6.75	59.97
		NAT2	93.06	8.74	65.74
		NAT2 + PP	**93.17**	12.42	77.82
	Accuracy & Params	NAT	90.73	2.91	30.03
		NAT	90.65	2.51	26.74
		NAT2	92.69	1.30	34.39
		NAT2	91.46	0.27	12.52
		NAT2 + PP	93.06	1.56	37.75
		NAT2 + PP	92.00	0.47	24.28
	Accuracy & MACs	NAT	89.98	2.40	15.33
		NAT	85.66	2.14	7.82
		NAT2	91.77	1.09	20.11
		NAT2	89.67	**0.19**	**6.35**
		NAT2 + PP	92.29	1.35	22.75
		NAT2 + PP	90.23	0.23	6.92
CIFAR100	Accuracy	NAT	66.93	6.26	55.72
		NAT2	71.88	9.75	56.60
		NAT2 + PP	**73.39**	11.13	74.28
	Accuracy & Params	NAT	66.83	3.62	31.44
		NAT	65.56	2.57	26.69
		NAT2	70.68	1.36	34.26
		NAT2	69.50	0.86	21.62
		NAT2 + PP	72.03	1.70	31.79
		NAT2 + PP	70.54	1.03	23.38
	Accuracy & MACs	NAT	64.76	2.70	16.05
		NAT	58.61	2.26	7.94
		NAT2	69.31	1.26	21.34
		NAT2	66.29	**0.21**	**8.14**
		NAT2 + PP	71.02	1.59	24.04
		NAT2 + PP	67.90	0.26	8.93
Tiny ImageNet	Accuracy	NAT	43.06	8.10	61.67
		NAT2	53.59	1.66	43.43
		NAT2 + PP	**54.82**	2.06	46.94
	Accuracy & Params	NAT	43.45	4.05	46.80
		NAT	42.99	2.71	28.00
		NAT2	51.16	1.44	39.19
		NAT2	39.92	**0.10**	**5.43**
		NAT2 + PP	54.31	1.85	41.70
		NAT2 + PP	45.03	**0.10**	**5.43**
	Accuracy & MACs	NAT	42.00	2.86	17.01
		NAT	38.89	2.39	8.06
		NAT2	51.05	1.46	28.41
		NAT2	47.24	0.25	5.95
		NAT2 + PP	53.91	1.87	31.97
		NAT2 + PP	48.96	0.32	6.55

Moving to CIFAR100, similar performance trends emerge. The NAT2 + PP configuration achieves the highest accuracy of 73.39%, demonstrating its ability to handle the complexity of the dataset, well above the 66.93% baseline accuracy set by NAT. NAT2's refined optimisation "Accuracy Params" produced a model with 0.86M parameters with an accuracy of 69.50%, a significant improvement over NAT-derived models within the same optimisation criteria. Interestingly, even in a multi-objective optimisation context, NAT2 outperforms the best NAT models in accuracy while significantly reducing both parameters and MACs. In particular, the "Accuracy MACs" optimisation model demonstrates NAT2's superior efficiency, achieving 66.29% accuracy with the fewest MACs (8.14M) and minimum parameters (0.21M), highlighting NAT2's enhanced capability for efficient model optimisation.

For the Tiny ImageNet dataset, NAT2 + PP again achieves the highest accuracy at 54.82%, demonstrating the improved classification capabilities of the model. Using efficiency-based optimisations, NAT2 produces a model with only 0.10M parameters that achieves an accuracy of 39.92%. Post-processing this model further improves its accuracy to 45.03% without increasing resource requirements, demonstrating the effectiveness of the post-processing step. Not only does this model outperform all NAT-derived models, but it does so with exceptionally minimal resource usage.

Overall, NAT2 + PP delivers consistently superior accuracy across all datasets evaluated. By achieving high accuracy with significantly fewer parameters, NAT2 demonstrates its superior efficiency. Furthermore, NAT2's ability to achieve optimal trade-offs between accuracy and computational requirements positions it as particularly well suited to model searches for memory-constrained devices. The inclusion of post-processing invariably benefits model performance, allowing the development of architectures that are both lighter and faster than those generated by NAT, thereby achieving significantly improved accuracy.

5 Conclusions

In this paper, we introduced Neural Architecture Transfer 2 (NAT2), an advanced Neural Architecture Search (NAS) technique designed to develop architectures that are not only high-performing but also markedly efficient in terms of parameters and computational operations. The experimental evidence demonstrates that NAT2 significantly outperforms its predecessor in both model accuracy and efficiency. Additionally, it has been established that the application of the proposed post-processing invariably enhances model accuracy, albeit with an increase in model complexity. Future directions for advancing NAS models encompass exploring more complex and expansive search spaces, as well as integrating attention mechanisms. These developments aim to foster solutions that are viable for real-world application, even on devices subject to stringent constraints, without compromising on performance.

Acknowledgment

This project has been supported by AI-SPRINT: AI in Secure Privacy-pReserving computINg conTinuum (European Union H2020 grant agreement No. 101016577) and FAIR: Future Artificial Intelligence Research (NextGenerationEU, PNRR-PE-AI scheme, M4C2, investment 1.3, line on Artificial Intelligence).

References

1. H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.
2. T. Elsken, J. H. Metzen, and F. Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.
3. A. Falanti, E. Lomurno, D. Ardagna, and M. Matteucci. Popnasv3: A pareto-optimal neural architecture search solution for image and time series classification. *Applied Soft Computing*, page 110555, 2023.
4. A. Falanti, E. Lomurno, S. Samele, D. Ardagna, and M. Matteucci. Popnasv2: An efficient multi-objective neural architecture search technique. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.
5. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.
6. C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.
7. H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.
8. Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.
9. E. Lomurno, S. Samele, M. Matteucci, and D. Ardagna. Pareto-optimal progressive neural architecture search. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1726–1734, 2021.
10. I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
11. Z. Lu, G. Sreekumar, E. Goodman, W. Banzhaf, K. Deb, and V. N. Boddeti. Neural architecture transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):2971–2989, 2021.
12. B. Lyu, S. Wen, K. Shi, and T. Huang. Multiobjective reinforcement learning-based neural architecture search for efficient portrait parsing. *IEEE Transactions on Cybernetics*, 53(2):1158–1169, 2021.
13. L. Ma, N. Li, G. Yu, X. Geng, S. Cheng, X. Wang, M. Huang, and Y. Jin. Pareto-wise ranking classifier for multi-objective evolutionary neural architecture search. *IEEE Transactions on Evolutionary Computation*, 2023.
14. L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.
15. S. Sarti, E. Lomurno, A. Falanti, and M. Matteucci. Enhancing once-for-all: A study on parallel blocks, skip connections and early exits. *arXiv preprint arXiv:2302.01888*, 2023.

16. S. Sarti, E. Lomurno, and M. Matteucci. Anticipate, ensemble and prune: Improving convolutional neural networks via aggregated early exits. *arXiv preprint arXiv:2301.12168*, 2023.
17. Y. Sun, H. Wang, B. Xue, Y. Jin, G. G. Yen, and M. Zhang. Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Transactions on Evolutionary Computation*, 24(2):350–364, 2019.
18. B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition, 2018.