NAS-Bench-360: Benchmarking Neural Architecture Search on Diverse Tasks

Renbo Tu[*]
University of Toronto
renbo.tu@mail.utoronto.ca

Nicholas Roberts[*]
University of Wisconsin
nick11roberts@cs.wisc.edu

Mikhail Khodak
Carnegie Mellon University
khodak@cmu.edu

Junhong Shen
Carnegie Mellon University
junhongs@andrew.cmu.edu

Frederic Sala
University of Wisconsin
fsala@wisc.edu

Ameet Talwalkar
Carnegie Mellon University
talwalkar@cmu.edu

Abstract

Most existing neural architecture search (NAS) benchmarks and algorithms prioritize well-studied tasks, e.g. image classification on CIFAR or ImageNet. This makes the performance of NAS approaches in more diverse areas poorly understood. In this paper, we present **NAS-Bench-360**, a benchmark suite to evaluate methods on domains beyond those traditionally studied in architecture search, and use it to address the following question: *do state-of-the-art NAS methods perform well on diverse tasks?* To construct the benchmark, we curate ten tasks spanning a diverse array of application domains, dataset sizes, problem dimensionalities, and learning objectives. Each new task is carefully chosen to interoperate with modern convolutional neural network (CNN) search methods while being far-afield from their original development domain. To speed up and reduce the cost of NAS research, for two of the tasks we release the precomputed performance of 15,625 architectures comprising a standard CNN search space. Experimentally, we show the need for more robust NAS evaluation of the kind NAS-Bench-360 enables by showing that several modern NAS procedures perform inconsistently across the ten tasks, with many catastrophically poor results. We also demonstrate how our benchmark and its associated precomputed results will enable future scientific discoveries by testing whether several recent hypotheses promoted in the NAS literature hold on diverse tasks. NAS-Bench-360 is hosted at `https://nb360.ml.cmu.edu/`.

1 Introduction

Neural architecture search (NAS) aims to automate the design of deep neural networks, ensuring performance on par with hand-crafted architectures while reducing human labor devoted to tedious architecture tuning [17]. With the growing number of application areas of ML, and thus of use-cases for automating it, NAS has experienced an intense amount of study in well-established machine learning domains, with significant progress in search space design [55, 36, 5], search efficiency [40], and search algorithms [47, 32, 46]. Notably, the field has largely been dominated by methods designed for and evaluated on benchmarks in computer vision [36, 48, 14], yet the use of NAS techniques may

[*]Equal contribution

Preprint. Under review.

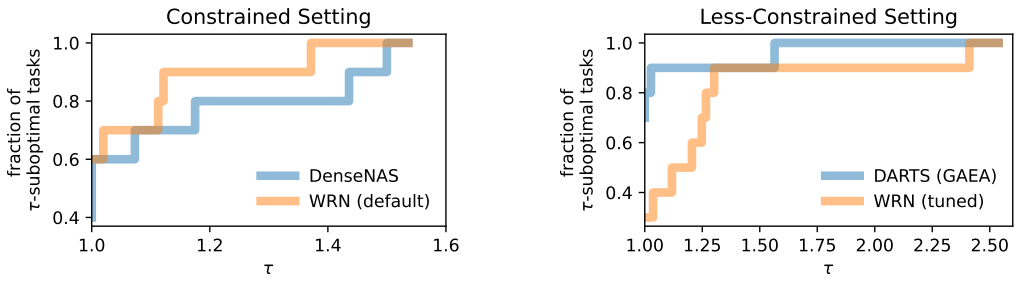

Figure 1: Performance profiles on NAS-Bench-360 comparing NAS methods (blue) to a fixed CNN (orange), specifically a Wide ResNet (WRN) [49]. Resource-constrained practitioners might be better off not using NAS (left), while less constrained practitioners can still benefit (right). The y-axis is the fraction of tasks on which error is within a factor τ of the optimal method, i.e. higher is better.

be especially impactful in under-explored or under-resourced domains where less is known about useful architecture design patterns. There have been a few recent efforts to diversify these benchmarks to settings such as vision-based transfer learning [15] and speech and language processing [37, 29]; however, evaluating NAS methods on such well-studied tasks using traditional CNN search spaces does not give a good indication of their utility on more far-afield applications, which have often necessitated the design of custom neural operations [9, 34].

We make progress towards studying NAS on more diverse tasks by introducing a suite of benchmark datasets drawn from various data domains that we collectively call **NAS-Bench-360**. This benchmark consists of an organized setup of ten suitable datasets that represent diverse application domains, dataset sizes, problem dimensionalities, and learning objectives. We also include a standard image classification task as a baseline point of comparison, as many new methods continue to be designed for that setting. Note that the core component of NAS-Bench-360 is *not* a typical NAS benchmark, which often involves precomputing all architectures in some fixed search space. In contrast, our contribution is explicitly intended to be agnostic of the search space being used, as different search spaces may work well for different tasks. Thus NAS-Bench-360 is a task-oriented NAS benchmark with the intended use-case of evaluating NAS method and search space pairs on a wide variety of domains. However, to aid research, three of our tasks—for two of which we contribute the precompute—do come with trained architectures from the NAS-Bench-201 search space [14].

Experimentally, we demonstrate the usefulness of NAS-Bench-360 by performing a set of analyses evaluating whether the success of NAS in computer vision is indicative of strong performance on the much broader set of problems to which NAS can be applied. Specifically, we report performance comparisons between NAS methods, investigate the validity of existing NAS hypotheses made solely on computer vision tasks, and extend an existing analysis of zero-cost proxies already-enabled by our benchmark [45]. From these analyses, we arrive at the following conclusions:

- Resource-constrained practitioners may be better of using a fixed CNN rather than NAS (Figure 1).
- NAS-Bench-201 analyses on computer vision tasks do not generalize to diverse tasks.
- Zero-cost proxies perform inconsistently on diverse tasks, corroborating earlier findings [45].

We have released all datasets, experiment code, precomputed models, seeds, and environments used in our experiments.[1] Releasing our code, random seeds, and environments in the form of Docker containers assures reproducibility of all experimental results presented in this work and encourages the same level of reproducibility for future research performed using NAS-Bench-360.

2 Related Work

Benchmarks have been critical to the development of NAS in recent years. This includes standard evaluation datasets and protocols, of which the most popular are the CIFAR-10 and ImageNet routines used by DARTS [36]. Another important type of benchmark has been tabular benchmarks such as NAS-Bench-101 [48], NAS-Bench-201 [14], NAS-Bench-1Shot1 [50], and TransNAS-Bench-

[1] `https://github.com/rtu715/NAS-Bench-360`

101 [16]; these benchmarks exhaustively evaluate all architectures in their search spaces, which is made computationally feasible by defining simple searched cells. Consequently, they are less expressive than the DARTS cell [36], often regarded as the most powerful search space in the cell-based regime. Notably, the full NAS-Bench-360 benchmark is *not* intended to be a tabular benchmark, i.e. we do *not* evaluate every architecture from a fixed search space on all ten of our tasks; instead, the focus is on the organization of a suite of tasks for assessing both NAS algorithms and search spaces, which would necessarily be restricted by fixing a search space for a tabular benchmark. Pre-computing on an expansive search space such as DARTS, with 10^{18} possible architectures, is computationally intractable. Architectures found on lesser search spaces are most likely suboptimal: a vanilla Wide ResNet (WRN) outperforms all networks in the NAS-Bench-201 search space on CIFAR-100. Nonetheless, we find that including precompute results for all of NAS-Bench-201 on two of our tasks is useful in evaluating various claims in the NAS literature centered on computer vision tasks.

While NAS methods and benchmarks have generally been focused on computer vision, recent work such as AutoML-Zero [41] and XD-operations [42] has started moving towards a more generically applicable set of tools for AutoML. However, even more recent benchmarks that do go beyond the most popular vision datasets have continued to focus on well-studied tasks, including vision-based transfer learning [15], speech recognition [37], and natural language processing [29]. We aim to go beyond such areas to evaluate the potential of NAS to automate the application of ML in truly under-explored domains. One analogous work to ours in the field of meta-learning is the Meta-Dataset benchmark of few-shot tasks [43], which similarly aimed to establish a wide-ranging set of evaluations for that field. For our inclusion of diverse tasks, we title our benchmark NAS-Bench-360 to resemble the idea of a 360-degree camera that covers all possible directions.

3 NAS-Bench-360: A Suite of Diverse and Practical Tasks

In this section, we introduce the NAS setting targeted by our benchmark, our motivation for organizing a new set of diverse tasks as a NAS evaluation suite, and our task-selection methodology. We report evaluations of specific algorithms on this new benchmark in the next section.

3.1 Neural Architecture Search: Problem Formulation and Baselines

For completeness and clarity, we first formally discuss the architecture search problem itself, starting with the extended hypothesis class formulation [32]. Here the goal is to use a dataset of points $x \in \mathcal{X}$ to find parameters $\mathbf{w} \in \mathcal{W}$ and $a \in \mathcal{A}$ of a parameterized function $f_{\mathbf{w},a} : \mathcal{X} \mapsto \mathbb{R}_{\geq 0}$ that minimize the expectation $\mathbb{E}_{x\sim\mathcal{D}} f_{\mathbf{w},a}(x)$ for some test distribution $\mathcal{D}$ over $\mathcal{X}$; here $\mathcal{X}$ is the input space, $\mathcal{W}$ is the space of model weights, and $\mathcal{A}$ is the set of architectures. For generality, we do not require the training points to be drawn from $\mathcal{D}$ to allow for domain adaptation, as is the case for one of our tasks, and we do not require the loss to be supervised. Note also that the goal here does not depend on computational or memory efficiency, which we do not focus on in our evaluations; our restriction is only that the entire pipeline can be run on an NVIDIA V100 GPU.

Notably, this formulation makes no distinction between the model weights $\mathbf{w}$ and architectures a, treating both as parameters of a larger model. Indeed, the goal of NAS may be seen as similar to model design, except now we include the design of an (often discrete) *architecture space* $\mathcal{A}$ such that it is easy to find an architecture $a \in \mathcal{A}$ and model weights $\mathbf{w} \in \mathcal{W}$ whose test loss $\mathbb{E}_{\mathcal{D}} f_{\mathbf{w},a}$ is low using a search algorithm. This can be done in a one-shot manner—simultaneously optimizing a and $\mathbf{w}$—or using the standard approach of first finding an architecture a and then keeping it fixed while training model weights $\mathbf{w}$ using a pre-specified algorithm such as stochastic gradient descent (SGD). This formulation divides NAS algorithms into two camps: one-shot, weight-sharing methods and non-weight-sharing ones such as random search, which operate by repeatedly sampling architectures and evaluating them. The formulation also includes non-NAS methods by allowing the architecture search space to be a singleton. When the sole architecture is a standard and common network such as WRN [49], this yields a natural baseline with an algorithm searching for training hyperparameters, not architectures. For our empirical investigation, we compare the performance of state-of-the-art NAS approaches against that of the three baselines: WRN, PerceiverIO [26], and XGBoost [6].

Table 1: Task metadata for NAS-Bench-360. Metrics are standardized such that lower is better.

Task name	Size	Dim.	Type	Learning objective	Metric	New to NAS?
CIFAR-100 [30]	60K	2D	Point	Classify natural images into 100 classes	0-1 error	no, widely used
Spherical [9]	60K	2D	Point	Classify spherically projected images into 100 classes	0-1 error	✓
NinaPro [4]	3956	2D	Point	Classify sEMG signals into 18 classes of hand gestures	0-1 error	✓
FSD50K [20]	51K	2D	Point (multi-label)	Classify sound events in log-mel spectrograms with 200 labels	$1-\text{mAP}$	✓
Darcy Flow [34]	1100	2D	Dense	Predict the final state of a fluid from its initial conditions	relative ℓ_2	no, used in [42]
PSICOV [3]	3606	2D	Dense	Predict pairwise distances between residuals from pairwise sequence features	MAE_8	no, used in [42]
Cosmic [51]	5250	2D	Dense	Predict probabilistic maps to identify cosmic rays in telescope images	FNR	✓
ECG [8]	330K	1D	Point	Detecting atrial cardiac disease from ECG recordings	$1-\text{F1}$	✓
Satellite [39]	1M	1D	Point	Classify satellite image pixel time series into 24 land cover types	0-1 error	✓
DeepSEA [10]	250K	1D	Point (multi-label)	Predicting chromatin and binding states of RNA sequences	$1-\text{AUROC}$	no, used in [52, 53]

3.2 Task Selection: Motivation and Methodology

Curating a diverse, practical set of tasks for the study of NAS is our primary motivation behind this work. We observe that past NAS benchmarks focused on creating larger search spaces and more sophisticated search methods for neural networks. However, the utility of these search spaces and methods are only evaluated on canonical computer vision datasets. On a broader range of problems, whether these new methods can improve upon simple baselines remains an open question. This calls for the introduction of new datasets lest NAS research overfits to the biases of CIFAR-10 and ImageNet. By identifying these possible biases, future directions in NAS research can be better primed to suit the needs of practitioners and to increase the deployment of NAS.

Summarized in Table 1, NAS-Bench-360 consists of problems that are conducive to processing by convolutional neural networks, which includes a trove of applications associated with spatial and temporal data, spanning single and multiple dimensions. Most current NAS methods are not implemented to search for other types of architectures to process tabular data and graph data. Therefore, we have set this scope for our investigation. During the selection of tasks, diversity is our primary consideration. We define the following axes of diversity to govern our task-filtering process: the first is problem dimensionality, including both 2D with matrix inputs and 1D with sequence inputs; the second is dataset size, for which our selection spans the scale from 1,000 to 1,000,000; the third is problem type , divisible into tasks requiring a singular prediction (point prediction) and multiple predictions (dense prediction); fourth and finally, diversity is achieved through selecting tasks from various learning objectives from applications of deep learning, where introducing NAS could improve upon the performance of handcrafted neural networks.

In lieu of providing raw data, we perform data pre-processing locally and store the processed data on a public Amazon Web Services S3 data bucket with download links available on our website. Our data treatment largely follows the procedure defined by the researchers who provided them. This enhances reproducibility by ensuring the uniformity of input data for different pipelines. Additional information about the datasets, pre-processing, and augmentation steps are described in the Appendix.

Table 2: Performance of NAS and baselines across NAS-Bench-360. Methods are divided into efficient methods (e.g. DenseNAS and fixed WRN) that take 1-10 GPU-hours, more expensive methods (e.g. DARTS and tuned WRN) that take 10-100+ GPU-hours, and specialized methods (Auto-DL and AMBER). All results are averages of three random seeds, and lower is better for all metrics.

Search space	Search algorithm	CIFAR-100	Spherical	Darcy Flow	PSICOV	Cosmic
WRN	default	**23.35±0.05**	85.77±0.71	0.073±0.001	3.84±0.05	51.76±2.09
DenseNAS	random	25.49±0.41	71.23±1.65	0.071±0.006	3.70±0.06	70.42±6.07
DenseNAS	original	25.98±0.38	72.99±0.95	0.100±0.010	3.84±0.15	79.52±2.20
Perceiver IO	default	70.04±0.44	82.57±0.19	0.240±0.010	8.06±0.06	100.0±0.00
XGBoost	default	84.83±4.15	96.92±0.02	0.085±0.000	n/a*	46.26±0.09
WRN	ASHA	23.39±0.01	75.46±0.40	0.066±0.00	3.84±0.05	37.53±10.2
DARTS	GAEA	24.02±1.92	**48.23±2.87**	**0.026±0.001**	**2.94±0.13**	**31.15±3.48**
Auto-DL	DARTS	n/a	n/a	0.049±0.005	6.73±0.73	99.79±0.02

Search space	Search algorithm	NinaPro	FSD50K	ECG	Satellite	DeepSEA
WRN	default	**6.78±0.26**	0.92±0.001	0.43±0.01	15.49±0.03	0.40±0.001
DenseNAS	random	8.45±0.56	**0.60±0.001**	0.42±0.01	13.91±0.13	0.40±0.001
DenseNAS	original	10.17±1.31	0.64±0.002	0.40±0.01	13.81±0.69	0.40±0.001
Perceiver IO	default	22.22±1.80	0.72±0.002	0.66±0.01	15.93±0.08	0.38±0.004
XGBoost	default	21.90±0.70	0.98±0.002	0.56±0.00	36.36±0.02	0.50±0.000
WRN	ASHA	7.34±0.76	0.91±0.030	0.43±0.01	15.84±0.52	0.41±0.002
DARTS	GAEA	17.67±1.39	0.94±0.020	0.34±0.01	**12.51±0.24**	0.36±0.020
AMBER	ENAS	n/a	n/a	**0.33±0.02**	12.97±0.07	**0.32±0.010**

* did not fit on a single V100 GPU.

Table 3: Median rank and performance improvement over WRN across NAS-Bench-360.

Search space Search algorithm	WRN default	DenseNAS original	DenseNAS random	WRN ASHA	DARTS GAEA	Auto-DL DARTS	AMBER ENAS
Median rank	4.0	4.0	4.0	3.5	1.5	6.0†	1.0†
% better than WRN*	0.0%	2.53%	0.0%	0.0%	20.1%	-75.3%†	20.0%†

* relative improvement over the default (untuned) WRN baseline
† metric computed only on the subset of three tasks on which the method was evaluated

4 Experimental design

Having detailed our construction of NAS-Bench-360, in this section we will establish the experimental setup for our analyses in the following section, which demonstrates the usefulness of NAS-Bench-360 for evaluating NAS methods on diverse tasks. We first specify the NAS methods and baselines we compare, followed by the details of the experimental setup and intended use of the benchmark. Finally, we provide details of the precomputed NAS-Bench-201 search space for two representative diverse tasks from NAS-Bench-360: NinaPro and Darcy Flow.

4.1 Baselines and Search Procedures

Our initial experiments follow two practitioners with different resource settings: one with enough compute to tune a WRN (less-constrained) and another who can only train it once with the default hyperparameters (constrained). Given these two scenarios, we compare against NAS methods that each practitioner would be able to run. In both cases, we focus on two well-known search paradigms: cell-based NAS (using DARTS [36]) and macro NAS (using DenseNAS [18]). We further compare these approaches to two customized NAS methods: Auto-DeepLab [35] for 2D dense prediction and AMBER [53] for 1D prediction, as well as general-purpose baselines: Perceiver IO [26] and XGBoost [6]. Additional details are provided in the Appendix.

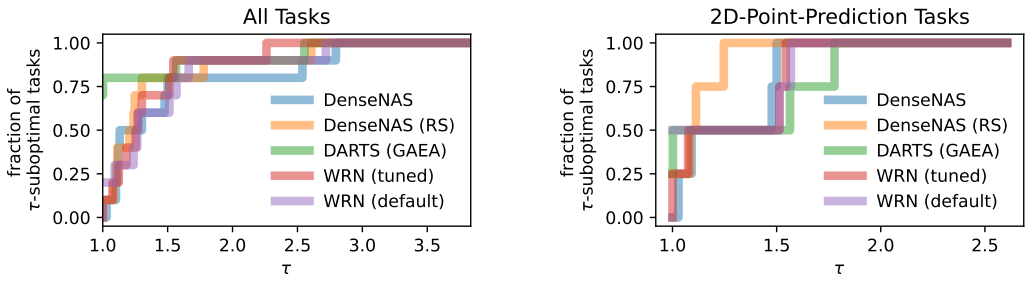

Figure 2: In our investigation of modern methods on NAS-Bench-360, we find that methods like GAEA DARTS can be strong in aggregate, as shown in the performance profiles on all tasks (left), but worse on salient subsets such as 2D point tasks (right). The y-axis is the fraction of tasks on which error is within a factor τ of the optimal method, i.e. higher is better.

4.2 Experimental Setup

Below we discuss the main reporting details of our empirical evaluation.

- **Hyperparameter tuning:** As detailed in the Appendix, we use the same hyperparameter ranges across all tasks to tune WRN. We use ASHA [31] to search over these hyperparameters and give it a budget on each task that matches the total search and retraining budget of DARTS (GAEA).
- **Aggregation metrics:** To aggregate results across tasks, we use the median rank of each method and its performance improvement over WRN for direct comparison via a singe number, as demonstrated in Table 3. However, since these metrics can be sensitive to small differences in performance, we also employ performance profiles [13] to mitigate that effect while still accounting for outliers. As described in Figure 1, these curves denote for each τ the fraction of tasks on which a method is no worse than a τ-factor from the optimal. Concretely, we plot $\rho_s(\tau) = \frac{1}{|\mathcal{P}|}\left|\left\{p \in \mathcal{P} : \frac{\text{error}_{p,s}}{\min_{s\in\mathcal{S}} \text{error}_{p,s}} \leq \tau\right\}\right|$ given some method $s \in \mathcal{S}$ on tasks $\mathcal{P}$.
- **Software and hardware:** We adopt the free, open-source software *Determined*[2] for experiment management, hyperparameter tuning, and cloud deployment. All experiments are performed on a single p3.2xlarge instance with an NVIDIA V100 GPU. Costs in GPU-hours are in the appendix.

4.3 Precomputing NAS-Bench-201 on NinaPro and Darcy Flow

The intended goal of NAS-Bench-360 is to evaluate the performance of *NAS search method and search space pairs* on diverse tasks, which precludes the precomputation of all architectures in general due to the lack of a single fixed search space. A complete lack of precomputed architectures would be perhaps limiting for many NAS researchers, who rely on precomputed NAS benchmarks when developing new search methods. In an effort to address this potential limitation, we precompute all architectures in the NAS-Bench-201 [14] search space on two representative tasks in NAS-Bench-360: NinaPro and Darcy Flow. We follow the same experimental procedure as in the original NAS-Bench-201 benchmark [14] to generate the precompute results, except where they vary the number of models trained for each architecture between one and three, we fix the number of trials per architecture to one. Note that since NAS-Bench-201 already includes precompute for CIFAR-100, a dataset we include in NAS-Bench-360.

5 Analysis

We conclude our presentation of NAS-Bench-360 with three sets of analyses. The first, a performance analysis of NAS methods and fixed baselines across diverse tasks, reveals new insights about the capabilities and robustness of current NAS methods and demonstrates how our benchmark can enable critical next steps in NAS research. In our second analysis, we evaluate claims from the NAS literature originally made using computer vision tasks, and show that they do not generalize to diverse tasks; this demonstrates how NAS research can benefit from our contribution in the future. Finally, we extend an existing analysis of zero-cost proxy methods on diverse tasks that already uses NAS-Bench-360 [45].

[2] `https://github.com/determined-ai/determined`

5.1 Performance across diverse tasks using NAS-Bench-360

As discussed in Section 4.2, we start by considering two practitioners faced with a choice of spending their limited compute on a (possibly tuned) fixed-architecture CNN or trying to find a better architecture using NAS. With this study, we investigate whether modern NAS methods perform well beyond the tasks for which they were designed.

1. A surface-level analysis suggests that under light resource constraints, modern NAS in the form of DARTS (GAEA) is quite robust to a wide variety of tasks: Table 3 shows it is the highest-ranked domain-independent method and attains the most significant improvement over the fixed WRN baseline. The performance profile in Figure 2 (left) also seems favorable, although it is overtaken by tuned WRN at a higher τ-suboptimality. However, a closer look at 2D point tasks in Figure 2 (right) reveals that DARTS is quite poor there, despite its design domain being image classification; in particular, it performs very poorly on NinaPro and FSD50K. Furthermore, on tasks where it performs well, it can still lag behind expert architectures; for example, on Darcy Flow, networks that use FNO [34] or XD-operations [42] do much better. Overall, our results suggest that this practitioner can apply NAS and expect to see some improvement, but also risks catastrophically poor performance (e.g. FSD50K) or not getting truly state-of-the-art results (e.g. Darcy Flow).
2. Under stronger budget constraints, our experiments strongly suggest that a practitioner should simply apply the default WRN to their problem rather than undergo the additional complexity of using DenseNAS, as the latter attains little-to-no improvement over the former in Table 3 and has a usually-worse performance profiles Figure 2. On the other hand, DenseNAS performs well on FSD50K—it outperforms all methods even while DARTS (GAEA) fails.

These first experiments suggest that the modern NAS methods are not always robust to diverse tasks, especially under resource-constrained settings. We believe that NAS-Bench-360's main roles as a future benchmark include developing an understanding of the multi-domain performance of existing approaches and guiding research into better NAS methods. While the latter is beyond the scope of this paper, our additional experiments demonstrate how NAS-Bench-360 facilitates the former.

Notably, several of our results address the question of the relative importance of search space vs. search algorithm. For example, Table 3 shows that on DenseNAS, random search is nearly identical to the more sophisticated weight-sharing scheme of the original paper; the two algorithms' performance profiles are also difficult to distinguish in Figure 2. Furthermore, AMBER—a 1D NAS method whose search space includes larger-kernel convolutions for handling such tasks—does better than GAEA even though it uses an older search algorithm (ENAS). These both suggest that search space design, including the use of a wider variety of operations, may be at least as crucial for success as the search algorithm. This point is reinforced by example tasks such as Darcy Flow, where architectures with more exotic operations substantially outperform our best results, as discussed earlier.

NAS-Bench-360 also reveals failure points of several methods, not just of general ones that usually perform quite well such as DARTS (GAEA) but also the objective-specific approach Auto-DL, which despite being designed for dense prediction tasks does poorly on all those considered here. Understanding when and why these performance drops happen is critical to developing a more robust NAS that is useful not just on average but in more challenging settings.

5.2 Do past NAS-Bench-201 analyses generalize to NAS-Bench-360?

Existing NAS-benches have been widely used for analyses such as (1) comparing performances of different architectures across tasks, (2) quickly evaluating search methods, and (3) investigating design choices that impact performance. In this section we show via the NAS-Bench-201 search space that the conclusions of past analyses cannot be assumed to hold on tasks beyond computer vision.

5.2.1 Architecture transferability

We start by using the precomputed results outlined in Section 4.3 to show in Figure 3 the rank of each architecture across different datasets, indexed on the x-axis by its rank on CIFAR-100. This reveals that while architecture rankings are highly correlated on image classification datasets—as pointed out by the authors of the original benchmark [14]—the rankings become uncorrelated when evaluated on a more diversified set of tasks. Therefore, NAS evaluations should be done across domains to verify true generalizability, and NAS-Bench-360 is especially useful for this purpose.

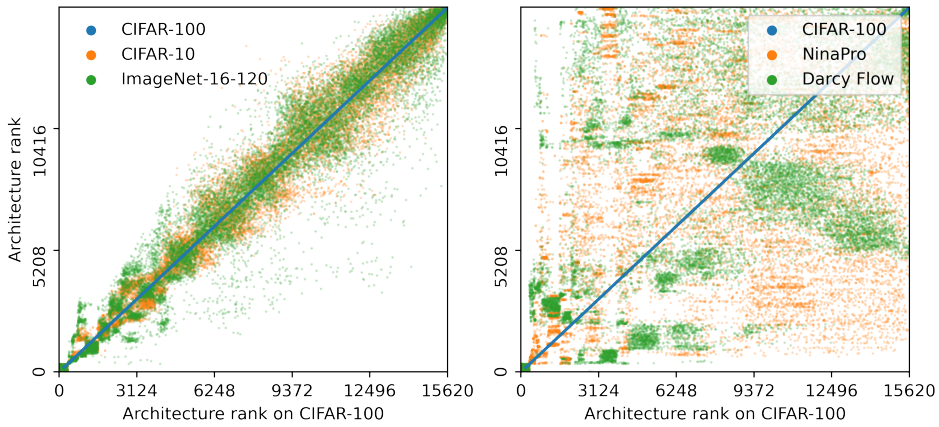

Figure 3: Architecture rankings between computer vision tasks correlate on NAS-Bench-201 [14] (left, sorted by performance on CIFAR-100) but are uncorrelated between CIFAR-100 and two NAS-Bench-360 tasks, NinaPro and Darcy Flow (right).

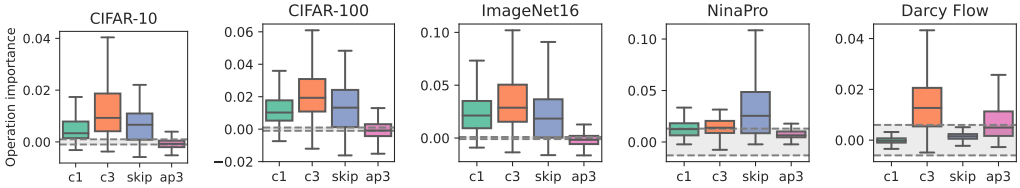

Figure 4: Different operations are important for different tasks. While prior work [44] shows that the operation importance distributions are stable across computer vision tasks—as shown by the high similarity of the three plots on the left—we find that they differ significantly for NinaPro and Darcy Flow.

5.2.2 Operation redundancy

Our final analysis using the NAS-Bench-201 search space is to investigate the conclusions of a more recent study on the redundancy of operations [44]. We find that the operation redundancy phenomenon they outline is task-dependent and does not generalize to tasks beyond the three vision tasks—CIFAR-10, CIFAR-100, and ImageNet16-120—that they study. To conduct our study we follow their procedure to obtain "operation importance" distributions for each operation in the NAS-Bench-201 search space for NinaPro and Darcy Flow; additionally, we reproduce their results on CIFAR-10, CIFAR-100, and ImageNet16-120. *Operation importance* measures the incremental effect of each operation choice in the NAS-Bench-201 search space—1x1 convolutions (c1), 3x3 convolutions (c3), skip connections (skip), and 3x3 average pooling (ap3)—on performance [44]. The original analysis found that the operation importance distributions are roughly similar across the original NAS-Bench-201 computer vision datasets, which we confirm and show in Figure 4. However, we found that the operation importance distributions were drastically different for NinaPro and Darcy Flow, which we also show in Figure 4. Not only are their distributions different from those of the computer vision tasks in the original analysis, but the operation importance distribution for NinaPro differs significantly from that of Darcy Flow. This tells us that *different operations are more useful for different tasks*, and using NAS-Bench-360, we find that we cannot conclude that any of these operations are universally redundant or useful in a given search space across tasks. In other words, using NAS-Bench-360, we find that the original claim that "existing search spaces contain a high degree of redundancy" [44] does not hold when considering diverse tasks beyond computer vision.

5.3 Zero-cost proxies on diverse tasks

We conclude with an analysis of TE-NAS [7], a zero-cost proxy inspired by neural tangent kernel (NTK) analysis, on four NAS-Bench-360 tasks. Zero-cost proxies [38, 1] are the subject of a recent direction in NAS research that aims to construct quick-to-evaluate measures of architecture performance

Table 4: Performance comparison of TE-NAS and GAEA using the DARTS search space on CIFAR-100, Spherical, NinaPro, and Darcy Flow. Lower is better for all metrics.

	CIFAR-100	Spherical	NinaPro	Darcy Flow
TE-NAS	24.32	56.87	**9.71**	**0.012**
GAEA	**24.02**	**48.23**	17.67	0.026

without doing any training. Recently, [45] evaluated several zero-cost proxies on tasks from NAS-Bench-360 (Spherical, NinaPro, and Darcy Flow), as well as on TransNAS-Bench-101 [15]. One major weakness of zero-cost proxies that they point out is that zero-cost proxies are not much more computationally efficient than weight-sharing methods, as the total compute cost is still dominated by the evaluation of the searched architecture [45]. For example, this renders TE-NAS in the DARTS search space comparable to GAEA DARTS in terms of computational efficiency. The authors of [45] also point out that the performance of different zero-cost proxies vary considerably across diverse datasets, even subject to the same search space. Performance may be strong on some tasks, but weak on others.

To expand such study of zero-cost proxies, we look at one that [45] do not consider—TE-NAS—and evaluate its performance on the DARTS space using four NAS-Bench-360 tasks: CIFAR-100, Spherical, NinaPro, and Darcy Flow. The results of this evaluation are shown in Table 4. Unlike many other zero-cost-proxies [38], the fact that TE-NAS is constructed from a domain-agnostic NTK analysis rather than experiments makes it a potential candidate for good performance on diverse tasks. However, Table 5 shows that performance does vary considerably across tasks, as observed for other proxies by [45]. In-particular, TE-NAS performs okay on NinaPro and beats all methods in Table 2 on Darcy Flow—where its performance approaches that of the expert-designed FNO [34]—but does poorly on Spherical. This evaluation adds evidence to existing scientific findings already enabled by NAS-Bench-360 [45] and provides additional evidence for the need to evaluate all NAS methods, including zero-cost proxies, on diverse tasks.

6 Conclusion

NAS-Bench-360 is a new performance benchmark consisting of ten diverse tasks derived from various fields of research and practice. It is designed for reproducible research on an academic budget that will guide the development of NAS methods and other automated approaches towards more robust performance across different domains. In initial results, we have demonstrated both the need for such a benchmark and the utility of NAS-Bench-360 specifically for developing new search spaces and algorithms. We also provide precompute architectures from the NAS-Bench-201 search space on two of the ten tasks. While the precomputed architectures on these two tasks are useful for analysis on their own, adding more precomputed search spaces and tasks is an area of further improvement. We welcome researchers to use the NAS-Bench-360 tasks to develop new procedures for automating ML.

Acknowledgments

We thank Maria-Florina Balcan for providing useful feedback. We also thank Hewlett Packard Enterprise for compute resources and the Determined AI open-source community for its support. This work was supported in part by DARPA FA875017C0141, the National Science Foundation grants IIS1705121, IIS1838017, IIS2046613 and IIS-2112471, an Amazon Web Services Award, a Facebook Faculty Research Award, funding from Booz Allen Hamilton Inc., a Block Center Grant, a Two Sigma Fellowship Award, and a Facebook PhD Fellowship Award. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of these funding agencies.

References

[1] Mohamed S. Abdelfattah, Abhinav Mehrotra, Lukasz Dudziak, and Nicholas D. Lane. Zero-cost proxies for lightweight NAS. In *Procedings of the 9th International Conference on Learning Representations*, 2021.

[2] Badri Adhikari. Deepcon: protein contact prediction using dilated convolutional neural networks with dropout. *Bioinformatics*, 36(2):470–477, 2020.

[3] Badri Adhikari. A fully open-source framework for deep learning protein real-valued distances. *Scientific reports*, 10(1):1–10, 2020.

[4] Manfredo Atzori, Arjan Gijsberts, Simone Heynen, Anne-Gabrielle Mittaz Hager, Olivier Deriaz, Patrick Van Der Smagt, Claudio Castellini, Barbara Caputo, and Henning Müller. Building the ninapro database: A resource for the biorobotics community. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 1258–1265. IEEE, 2012.

[5] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[6] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[7] Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four {gpu} hours: A theoretically inspired perspective. In *International Conference on Learning Representations*, 2021.

[8] Gari D Clifford, Chengyu Liu, Benjamin Moody, H Lehman Li-wei, Ikaro Silva, Qiao Li, AE Johnson, and Roger G Mark. Af classification from a short single lead ecg recording: The physionet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.

[9] Taco S. Cohen, Mario Geiger, Jonas Koehler, and Max Welling. Spherical CNNs. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[10] ENCODE Project Consortium et al. The encode (encyclopedia of dna elements) project. *Science*, 306(5696):636–640, 2004.

[11] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):760–771, 2019.

[12] Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.

[13] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

[14] Xuanyi Dong and Yi Yang. NAS-Bench-201: Extending the scope of reproducible neural architecture search. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[15] Yawen Duan, Xin Chen, Hang Xu, Zewei Chen, Xiaodan Liang, Tong Zhang, and Zhenguo Li. TransNAS-Bench-101: Improving transferability and generalizability of cross-task neural architecture search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[16] Yawen Duan, Xin Chen, Hang Xu, Zewei Chen, Li Xiaodan, Tong Zhang, and Zhenguo Li. Transnas-bench-101: Improving transferability and generalizability of cross-task neural architecture search. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5247–5256, 2021.

[17] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

[18] Jiemin Fang, Yuzhu Sun, Qian Zhang, Yuan Li, Wenyu Liu, and Xinggang Wang. Densely connected search space for more flexible neural architecture search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[19] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. Fsd50k: an open dataset of human-labeled sound events. *arXiv preprint arXiv:2010.00475*, 2020.

[20] Eduardo Fonseca, Jordi Pons Puig, Xavier Favory, Frederic Font Corbera, Dmitry Bogdanov, Andres Ferraro, Sergio Oramas, Alastair Porter, and Xavier Serra. Freesound datasets: a platform for the creation of open audio datasets. In *Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93.* International Society for Music Information Retrieval (ISMIR), 2017.

[21] John S Garofolo. Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium, 1993*, 1993.

[22] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE, 2017.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[24] Shenda Hong, Yanbo Xu, Alind Khare, Satria Priambada, Kevin Maher, Alaa Aljiffry, Jimeng Sun, and Alexey Tumanov. Holmes: health online model ensemble serving for deep learning models in intensive care units. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1614–1624, 2020.

[25] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[26] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022.

[27] David Josephs, Carson Drake, Andy Heroy, and John Santerre. semg gesture recognition with a simple model of attention. In *Machine Learning for Health*, pages 126–138. PMLR, 2020.

[28] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Kathryn Tunyasuvunakool, Olaf Ronneberger, Russ Bates, Augustin Žídek, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Anna Potapenko, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Martin Steinegger, Michalina Pacholska, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. High accuracy protein structure prediction using deep learning. *In Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book)*, 2020.

[29] Nikita Klyuchnikov, Ilya Trofimov, Ekaterina Artemova, Mikhail Salnikov, Maxim Fedorov, and Evgeny Burnaev. NAS-Bench-NLP: Neural architecture search benchmark for natural language processing. arXiv, 2020.

[30] Alex Krizhevksy. Learning multiple layers of features from tiny images. Technical report, 2009.

[31] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A system for massively parallel hyperparameter tuning. *arXiv preprint arXiv:1810.05934*, 2018.

[32] Liam Li, Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Geometry-aware gradient algorithms for neural architecture search. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[33] Lisha Li, Kevin G Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. In *ICLR (Poster)*, 2017.

[34] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[35] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 82–92, 2019.

[36] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[37] Abhinav Mehrotra, Alberto Gil, C. P. Ramos, Sourav Bhattacharya, Łukasz Dudziak, Ravichander Vipperla, Thomas Chau, Samin Ishtiaq, Mohamed S. Abdelfattah, and Nicholas D. Lane. NAS-Bench-ASR: Reproducible neural architecture search for speech recognition. In *Proceedings of the 8th International Conference on Learning Representations*, 2021.

[38] Joseph Mellor, Jack Turner, Amos Storkey, and Elliot J. Crowley. Neural architecture search without training. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[39] François Petitjean, Jordi Inglada, and Pierre Gançarski. Satellite image time series analysis under time warping. *IEEE transactions on geoscience and remote sensing*, 50(8):3081–3095, 2012.

[40] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[41] Esteban Real, Chen Liang, David R. So, and Quoc V. Le. AutoML-Zero: Evolving machine learning algorithms from scratch. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[42] Nicholas Roberts, Mikhail Khodak, Tri Dao, Liam Li, Chris Ré, and Ameet Talwalkar. Rethinking neural operations for diverse tasks. arXiv, 2021.

[43] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[44] Xingchen Wan, Binxin Ru, Pedro M Esperança, and Zhenguo Li. On redundancy and diversity in cell-based neural architecture search. In *International Conference on Learning Representations*, 2022.

[45] Colin White, Mikhail Khodak, Renbo Tu, Shital Shah, Sébastien Bubeck, and Debadeepta Dey. A deeper look at zero-cost proxies for lightweight nas. In *ICLR Blog Track*, 2022. https://iclr-blog-track.github.io/2022/03/25/zero-cost-proxies/.

[46] Colin White, Willie Neiswanger, and Yash Savani. BANANAS: Bayesian optimization with neural architectures for neural architecture search. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021.

[47] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. PC-DARTS: Partial channel connections for memory-efficient architecture search. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[48] Chris Ying, Aaron Klein, Eric Christiansen, Esteban Real, Kevin Murphy, and Frank Hutter. NAS-Bench-101: Towards reproducible neural architecture search. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[49] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016.

[50] Arber Zela, Julien Siems, and Frank Hutter. NAS-Bench-1Shot1: Benchmarking and dissecting one-shot neural architecture search. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[51] Keming Zhang and Joshua S Bloom. deepcr: Cosmic ray rejection with deep learning. *The Astrophysical Journal*, 889(1):24, 2020.

[52] Zijun Zhang, Evan M Cofer, and Olga G Troyanskaya. Ambient: accelerated convolutional neural network architecture search for regulatory genomics. *bioRxiv*, 2021.

[53] Zijun Zhang, Christopher Y Park, Chandra L Theesfeld, and Olga G Troyanskaya. An automated framework for efficiently designing deep convolutional neural networks in genomics. *Nature Machine Intelligence*, 3(5):392–400, 2021.

[54] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

[55] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

A Dataset Descriptions

CIFAR-100: Standard image classification As a starting point of comparison to existing benchmarks, we include the **CIFAR-100** task [30], which contains RGB images from natural settings to be classified into 100 fine-grained categories. CIFAR-100 is preferred over CIFAR-10 because it is more challenging and suffers less from over-fitting in previous research.

Spherical: Classifying spherically projected CIFAR-100 images To test NAS methods applied to natural-image-like data, we consider the task of classifying spherical projections of the CIFAR-100 images, which we call **Spherical**. In addition to scientific interest, spherical image data is also present in various applications, such as omnidirectional vision in robotics and weather modeling in meteorology, as sensors usually produce distorted image signals in real-life settings. To create Spherical CIFAR, we project the planar signals of the CIFAR images to the northern hemisphere and add a random rotation to produce spherical signals for each channel following the procedure specified in [9]. The resulting images are 60×60 pixels with RGB channels.

NinaPro: Classifying electromyography signals **NinaPro** moves away from the image domain to classify hand gestures indicated by electromyography signals. For this, we use a subset of the NinaPro DB5 dataset [4] in which two Myo armbands collect EMG signals from 10 test individuals who hold 18 different hand gestures to be classified. These armbands leverage data from muscle movement, which is collected using electrodes in the form of wave signals. Each wave signal is then sampled using a wavelength and frequency prescribed in [11] to produce 2D signals.

FSD50K: Labeling sound events **FSD50K** [19] is derived from the larger Freesound dataset [20] of Youtube videos with 51,000 clips totaling more than 100 hours of sound. These clips are manually labeled and equally distributed in 200 classes from the AudioSet ontology [22]. Each clip could receive multiple labels. Unlike TIMIT [21], FSD50K does not focus exclusively on sounds of spoken language but includes sound events from physical sources and production mechanisms. The mean average precision (mAP) is used to evaluate classification results.

Darcy Flow: Solving partial differential equations (PDEs) Our first regression task, **Darcy Flow**, focuses on learning a map from the initial conditions of a PDE to the solution at a later timestep. This application aims to replace traditional solvers with learned neural networks, which can output a result in a single forward pass. The input is a 2d grid specifying the initial conditions of a fluid, and the output is a 2d grid specifying the fluid state at a later time, with the ground truth being the result computed by a traditional solver. This we use the same Darcy Flow dataset that was used in [34]. We report the mean square error (MSE or ℓ_2).

PSICOV: Protein distance prediction **PSICOV** studies the use of neural networks in the protein folding prediction pipeline, which has recently received significant attention to the success of methods like AlphaFold [28]. While the dataset and method they use are too large-scale for our purposes, we consider a smaller set of protein structures to tackle the specific problem of inter-residual distance predictions outlined in [3]. 2D large-scale features are extracted from protein sequences, resulting in input feature maps with a massive number of channels. Correspondingly, the labels are pairwise-distance matrices with the same spatial dimension. The evaluation metric is mean absolute error (MAE or ℓ_1) computed on distances below 8 Å, referred to as MAE_8.

Cosmic: Identifying cosmic ray contamination Images from space-based facilities are prone to corruption by charged particles collectively referred to as "cosmic rays." Cosmic rays on images should be identified and masked before the images are used for further analysis [51]. The **Cosmic** task uses imaging data of local resolved galaxies collected from the Hubble Space Telescope. Inputs and outputs are same-size 2D matrices, with the output predicting whether each pixel in the input is an artifact of cosmic rays. We report the false-negative rate (FNR) of identification results.

ECG: Detecting heart disease Electrocardiograms are frequently used in medicine to diagnose sinus rhythm irregularities. The **ECG** task is based on the 2017 PhysioNet Challenge [8], with 9 to 60-second ECG recordings sampled at 300 Hz and labeled using four classes: normal, disease, other, or noisy rhythms. Recordings are processed using a fixed sliding window of 1,000 ms and stride of 500 ms. We report the F1-score according to the challenge's guidelines.

Satellite: Satellite image time series analysis Satellite image time series (SITS) are becoming more widely available in earth monitoring applications. Our dataset comes from Formosat-2 satellite images acquired over Toulouse, France [39]. Available in multiple channels, SITS track the land cover changes over several years as each pixel in the image represents a geographical region. The goal of the **Satellite** task is to generate land cover maps for geo-surveying. Specifically, a series of pixels in a given color channel which constitutes a time series to be classified into 24 land cover types.

DeepSEA: Predicting functional effects from genetic sequences Predicting chromatin effects of genetic sequence alterations is a significant challenge in the field to understand genetic diseases. **DeepSEA** [54], provides a compendium of genomic profiles from the Encyclopedia of DNA Elements (ENCODE) project [10] to train a predictive model estimating the behavior of chromatin proteins, divided into 919 categories. Due to computation constraints, we subsample 36 of these categories as per [52] and further take 5% of the training data for prediction. We report the area under the receiver operating characteristic (AUROC) following the previous work.

B Baselines

Wide ResNet with Hyperparameter Tuning Architectures based on ResNet [23] are a common first choice by practitioners faced with a new domain [19, 3]; it is thus a natural source of fixed-architecture baselines for our study. We use the Wide ResNet variant [49] with 16 layers and a widen factor of 4, and apply its original training routine directly for the constrained practitioner. For the other practitioner, we wrap the training procedure with a hyperparameter tuner, ASHA [31], an asynchronous version of Hyperband [33]. Given a range for each hyperparameter, ASHA uniformly samples configurations and uses brackets of elimination: at each round, each configuration is trained for some epochs, before the algorithm selects the best-performing portion based on validation metrics. This procedure is useful for finding suitable hyperparameters in an easy-to-use, automated fashion.

Cell-based search using DARTS The first NAS paradigm we consider is cell-based NAS. These methods first search for a genotype, a cell containing neural operations. During evaluation, an architecture is constructed by replicating the searched cell and stacking them together. The most popular search space for this approach is DARTS [36], which assigns one of eight operations to six edges in two types of cells: "normal" cells preserve the shape of the input while "reduction" cells downsample it. For dense tasks, we do not use the reduction cell to prevent introducing a bottleneck. For 1D tasks, all convolutions in the cell are converted from 2D to 1D. Finally, to adhere to standard ML practices, we do *not* adapt the standard DARTS pipeline from the original paper. As this search space has been heavily studied, we use as a search routine a recent approach, GAEA PC-DARTS (GAEA), that achieves some of the best-known results on CIFAR-10 and ImageNet [32]. This NAS approach, due to its heavy retraining routine, is compared to the tuned WRN baseline of the less-resource-constrained practitioner.

Macro NAS using DenseNAS The second NAS paradigm we consider is macro NAS. Instead of building from a fixed cell, it requires the specification of a supernet with different inter-connected blocks. These blocks and connections are then pruned to construct an architecture. For our benchmark, we choose a recent search space, DenseNAS [18], which has near state-of-the-art results on ImageNet. DenseNAS searches for architectures with densely-connected, customizable routing blocks to emulate DenseNet [25]. In our experiments, we use the ResNet-based search space, DenseNAS-R1, which contains all neural operations of the WRN backbone. For 2D tasks, we adapt two super networks from the one used for ImageNet as inputs to the search algorithm. The super network for dense tasks maintains the same spatial dimensions without downsampling to avoid bottlenecks, and we lower the learning rate for evaluating architectures to prevent divergence. For transferring to 1D tasks, all network operations are switched from 2D to 1D. Other training and evaluation procedures are identical to those in the original paper and uniform across all tasks. DenseNAS is quick to search and evaluate, making it comparable to the fixed WRN baseline.

We apply another search method to the fixed DenseNAS space to study the relative importance of search algorithms. Random search is implemented through randomly sampling architectures from the DenseNAS space and validating them after a brief training period of 10 epochs before evaluating the best performer. To ensure fairness of comparison, we allot equal GPU hours to random search and

regular DenseNAS search, additionally applying a soft constraint that random architecture model sizes should not surpass DenseNAS searched architecture sizes significantly.

Domain-specific NAS Baselines: Auto-DL and AMBER To study the importance of search spaces, we further handpick two domain-specific NAS approaches applicable only to a subset of the tasks. Using an encoder-decoder architecture, Auto-DeepLab (Auto-DL) [35] is designed for dense prediction, e.g., semantic segmentation. While the decoder is fixed, Auto-DL searches for an encoder via first-order DARTS. We evaluate Auto-DL on Darcy Flow, PSICOV, and Cosmic tasks.

AMBER [53] aims to automate neural network design for 1D genomic data. This framework establishes a 12-layer network backbone and parametrizes a long-short term memory network (LSTM) as a controller to search for suitable 1D operations and residual connections, following the ENAS [40] optimization protocol. At each step, the controller samples architectures to compute reward based on area under the receiver operating characteristics (AUROC) before outputting the highest-reward architecture. We evaluate AMBER on the ECG, Satellite, and DeepSEA tasks.

General-purpose baselines: Perceiver IO and XGBoost As the overarching theme of NAS-Bench-360 is evaluate NAS methods on a wide variety of diverse tasks and when to even use NAS methods over fixed baselines, general-purpose non-NAS methods are obvious points of comparison. We evaluate two such baselines on NAS-Bench-360: the recent transformer-based Perceiver IO [26], and the popular non-deep learning baseline, XGBoost [6]. Perceiver IO is a general-purpose transformer architecture that is designed to handle arbitrary input and output dimensionalities with minimal changes to its encoder and decoder networks—as such, we evaluate Perceiver IO on all 10 NAS-Bench-360 tasks. Similarly, the popular gradient-boosting method, XGBoost, is applicable to a wide variety of tasks and learning objectives, including single-output and multi-output classification and regression problems, which covers all 10 tasks in NAS-Bench-360. For efficiency and comparison to deep learning methods, we employ the GPU-based implementation of histogram gradient-boosting in XGBoost.

C Comparison of NAS with Expert Architectures

We create a more challenging baseline for NAS by evaluating hand-designed architectures for each specific task. Hand-crafted networks are selected according to best-effort search. The full evaluation results of NAS methods vs. non-NAS baselines can be found in Table 6. Figure 6 illustrates a comparison between best-performing NAS methods vs. best non-NAS methods. Surprisingly, GAEA PC-DARTS beats all the baselines on a portion of the tasks.

Here is a brief summary of these expert models and their citations:

1. DenseNet-BC (CIFAR-100): a more sophisticated version of ResNet, achieving state-of-the-art performance on vision classification [25].
2. S2CNN (Spherical): a spherical CNN contains special operations designed for spherical signals, state-of-the-art on spherically-projected MNIST [9].
3. Fourier Neural Operator (FNO) Network (Darcy Flow): via parametrization in Fourier space, FNO can efficiently learn a family of partial differential equations and their mapping to solutions [34].
4. DEEPCON (PSICOV): a dilated-convolution neural network combined with dropout to optimize for protein distance prediction [2].
5. deepCR-mask (Cosmic): a modified version of UNet retaining data dimension to keep pixels at the borders to suit astronomy applications, state-of-the-art on this task [51].
6. Attention-based model (NinaPro): a lightweight feed-forward neural network adopting attention modules in place of convolutions [27].
7. VGG-like (FSD50K): a smaller VGG network with output features combining both global max pooling and average pooling for audio [19].
8. ResNet-1D (ECG): ResNet with 1D convolution, using a larger kernel size of 16 and a stride of 2 for all convolutions. The architecture is state-of-the-art on several time-series prediction tasks in medicine [24].

Table 5: Experiment training runtimes of NAS-Bench-360 (GPU hours)

Task	GAEA	DenseNAS	WRN	AMBER / Auto-DeepLab
CIFAR-100	9.5	2.5	2	n/a
Spherical	16.5	2.5	2	n/a
Darcy Flow	6.5	0.5	0.5	5.5
PSICOV	18	24	18.5	19
Cosmic	21.5	2.5	4	17.5
NinaPro	0.5	0.2	0.2	n/a
FSD50K	37	4.5	4	n/a
ECG	140	6.5	5	27
Satellite	28	3	4.5	26
DeepSEA	39.5	2	1.5	28

9. ROCKET (Satellite): a simple linear classifier with random convolution kernel as a feature extractor, achieving state-of-the-art performance on UCR time-series prediction tasks [12].
10. DeepSEA model (DeepSEA): the original 1D convolution model accompanying the dataset, state-of-the-art on DeepSEA itself [54].

D Experiment Details

D.1 Hyperparameter Tuning and Backbone

We use a wide residual network with 16 layers and a widening factor of 4 (WRN-16-4) for all tasks.

For tuning hyperparameters, we use ASHA's default elimination schedule and search over 7 to 256 randomly sampled hyperparameter configurations matching GAEA PC-DART's runtime. The maximum epochs that a single configuration could be trained is equal to that of Wide ResNet's default, 200.

We have selected the following hyperparameter ranges for tuning the Wide ResNet backbone:

- $\log_{10}$(learning rate): Unif[-4, -1]
- momentum: Unif{0.0, 0.3, 0.6, 0.9}
- $\log_{10}$(weight decay): Unif[-5, -2]
- dropout: Unif{0.0, 0.3, 0.6}
- batch size: 128 (all point tasks except FSD50K), 4 (Darcy Flow), 8 (PSICOV, Cosmic), 256(FSD50K, ECG, Deepsea), 4096 (Satellite)

D.2 Reference Runtimes

Using a Nvidia V100 GPU, we have recorded the following runtimes for each experiment in this benchmark in Table 5. Overall, GAEA PC-DARTS is more costly than backbone with hyperparameter optimization, which is more costly than DenseNAS. The protein tasks requires heavy computation since the data is not static but generated during training.

D.3 Model sizes and FLOPS statistics

Full information of model parameter counts and FLOPs can be found in Table 7 and Table 8.

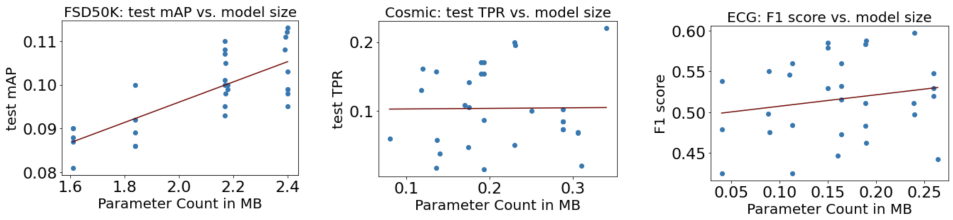

Figure 5: Performances v. Model sizes for three sample tasks.

D.4 Adjustments for Dense Prediction Tasks

On the wide ResNet backbone, we add an adaptive averaging pooling operation to upsample the features back to their original dimensions before output. On the DARTS space, we prevent downsampling and keep spatial dimensions unchanged by disabling reduction cells and replacing them with normal cells. On DenseNAS, we configure the super-network to contain only blocks with the original spatial dimensions.

D.5 Adjustments for 1D Prediction Tasks

The WRN-1D does not have a convolution stem and uses larger kernel sizes of 8, 5, 3 in each convolution block. We substitute 2D operations with 1D operations within the DARTS and DenseNAS search spaces.

D.6 Random Seeds

For main experiments, we fix the random seed to be 0, 1, 2 for each of the 3 trials respectively.

For AMBER experiments, we completed three trials as the package did not offer the option of setting random seeds.

D.7 Correlation between Performance and Model Size

We plot performances of 30 random architectures from the DenseNAS search space across three tasks in Figure 5. From our random search experiment, larger models searched by NAS are not always better-performing. We study the Pearson correlation coefficient between test performance vs. model size in number of parameters for three tasks: FSD50K, Cosmic, and ECG. On Cosmic and ECG, the correlation is very weak ($r = 0.01$ and $r = 0.19$ respectively). On FSD50K, a stronger correlation ($r = 0.79$) is observed but performance varies significantly even for architectures of the same size.

E Supplementary Materials

E.1 Data License

- CIFAR-100: CC BY 4.0 (on `https://www.tensorflow.org/datasets/catalog/cifar100`)
- Spherical CIFAR-100: CC BY-SA
- NinaPro: CC BY-ND
- FSD50k: CC BY 4.0
- Darcy Flow: MIT
- DeepCov, PSICOV: GPL
- Cosmic: Open License (https://registry.opendata.aws/hst/)
- ECG: ODC-BY 1.0
- Satellite: GPL 3.0
- Deepsea: CC BY 4.0

E.2 Data Preprocessing Details

CIFAR-100: while the 10,000 testing images are kept aside only for evaluating architectures, the 50,000 training images are randomly partitioned into 40,000 for architecture search and 10,000 for validation. On all of the 50,000 training images, we apply standard CIFAR augmentations including random crops and horizontal flipping, and finally normalize them using a pre-calculated mean and standard deviation of this set. On the 10,000 testing images, we only apply normalization with the same constants.

Spherical: with the same split ratios CIFAR-100, the generated spherical image data is directly used for training and evaluation without data augmentation and pre-processing.

NinaPro: Containing less than 4,000 samples, the data is comprised of single-channel signals with an irregular shape of 16*52 pixels. This task also differs from CIFAR for its class imbalance, as over 65% of all gestures are the neutral position. We split the data using the same ratio as CIFAR, resulting in 2638 samples for training, and 659 samples for validation and testing each. No additional pre-processing is performed.

FSD50K: The raw sound files are first resampled at a frequency of 22,050 Hz and transformed into 96-band, log-mel spectrograms, which is a representation of the sound's power spectrum. Small overlapping audio chunks of 1 second are obtained from these larger clips, resulting in an input size of 101*96 (101 frames of 96-band spectrograms). As data augmentation, background noise of the same frequency is also mixed into 75% of the training data. We split 4,170 clips into the validation set and 10,231 clips into the test set following the original paper. During training, we train on one randomly-sampled chunk, instead of all chunks, from each clip.

Darcy Flow: we use scripts provided by [34] to generate the PDEs and their solutions, for a total of 900 data points for training, 100 for validation, and 100 for testing. All input data is normalized with constants calculated on the training set before fed into the neural network and de-normalized following an encode-decode scheme. The solutions, or labels, for the training set are also encoded and decoded this way. The test labels are not processed.

PSICOV: we adopt the chosen subset of DeepCov proteins in [3], consisting of 3,456 proteins each with 128*128 feature maps across 57 channels. 100 proteins from this set are used for validation and the rest for training. Test data for final evaluation is gathered from another set of 150 proteins, PSICOV. Since these produce feature maps that are larger (512*512), we run the prediction network over all of its non-overlapping 128*128 patches.

Cosmic: we use data from a specific filter, F435W, of the space telescope, representing the 3605–4882 Å spectral range. Image stamps of 256*256 pixels are taken from large images. The dataset contains 4,347 stamps for training, and 420 for test, and 483 for validation to match the test set size.

ECG: from the sliding window approach, 12,186 single lead recordings are converted into more than 330,000 recording segments comprised of 261,740 for training, 33,281 for validation, and 33,494 for test. Each segment is of the shape 1*1,000, representing one channel of 1,000-long temporal sequence.

Satellite: each satellite time-series is single-channel of length 46 (1*46). After applying standard normalization, we divide the one million entries to 800,000 for training, 100,000 for validation, and 100,000 for test. Zero-padding to 48-length sequences is required for DenseNAS' downsampling network.

DeepSEA: the genome sequences are 1,000-base pair (bp) long and represented as a 1000×4 binary matrix, as each bp is represented as an one-hot encoding corresponding to either A,C,T,G at that location. Total training set size is 71,753. Validation and test sizes that are not subsampled are 2,490 and 149,400 respectively.

Table 6: Performance of NAS and baselines across NAS-Bench-360 compared to expert architectures. All results are averages of three random seeds, and lower is better for all metrics.

Search space	Search algorithm	CIFAR-100	Spherical	Darcy Flow	PSICOV	Cosmic
WRN	default	23.35±0.05	85.77±0.71	0.073±0.001	3.84±0.05	51.76±2.09
DenseNAS	random	25.49±0.41	71.23±1.65	0.071±0.006	3.70±0.06	70.42±6.07
DenseNAS	original	25.98±0.38	72.99±0.95	0.100±0.010	3.84±0.15	79.52±2.20
Perceiver IO	default	70.04±0.44	82.57±0.19	0.240±0.010	8.06±0.06	100.0±0.00
XGBoost	default	84.83±4.15	96.92±0.02	0.085±0.000	n/a*	46.26±0.09
WRN	ASHA	23.39±0.01	75.46±0.40	0.066±0.00	3.84±0.05	37.53±10.2
DARTS	GAEA	24.02±1.92	**48.23±2.87**	0.026±0.001	**2.94±0.13**	31.15±3.48
Auto-DL	DARTS	n/a	n/a	0.049±0.005	6.73±0.73	99.79±0.02
Expert	default	**19.39±0.20**	67.41±0.76	**0.008±0.001**	3.35±0.14	**25.29±1.44**

Search space	Search algorithm	NinaPro	FSD50K	ECG	Satellite	DeepSEA
WRN	default	**6.78±0.26**	0.92±0.001	0.43±0.01	15.49±0.03	0.40±0.001
DenseNAS	random	8.45±0.56	**0.60±0.001**	0.42±0.01	13.91±0.13	0.40±0.001
DenseNAS	original	10.17±1.31	0.64±0.002	0.40±0.01	13.81±0.69	0.40±0.001
Perceiver IO	default	22.22±1.80	0.72±0.002	0.66±0.01	15.93±0.08	0.38±0.004
XGBoost	default	21.90±0.70	0.98±0.002	0.56±0.00	36.36±0.02	0.50±0.000
WRN	ASHA	7.34±0.76	0.91±0.030	0.43±0.01	15.84±0.52	0.41±0.002
DARTS	GAEA	17.67±1.39	0.94±0.020	0.34±0.01	**12.51±0.24**	0.36±0.020
AMBER	ENAS	n/a	n/a	0.33±0.02	12.97±0.07	0.32±0.010
Expert	default	8.73±0.90	0.62±0.004	**0.28±0.00**	19.80±0.00	**0.30±0.024**

* did not fit on a single V100 GPU.

Table 7: Paramater counts of searched and baseline models for all tasks of NAS-Bench-360. Searched model sizes are reported as mean±standard deviation of three random seeds. Results are reported in millions (M). Architectures with the best performance are bolded.

Search space	Search algorithm	CIFAR-100	Spherical	Darcy Flow	PSICOV	Cosmic
DenseNAS	random	1.74±0.12	2.23±0.47	1.00±0.18	1.21±0.16	0.25±0.06
DenseNAS	original	2.03±0.53	1.84±0.15	0.38±0.13	0.93±0.36	0.15±0.16
DARTS	GAEA	4.92±0.28	**1.67±0.14**	0.63±0.08	**0.53±0.05**	0.43±0.15
Auto-DL	DARTS	n/a	n/a	22.98±3.49	6.50±1.84	7.61±2.14
WRN	default	2.77	2.77	2.75	2.76	2.75
Expert	default	**3.08**	0.16	**1.19**	0.60	**0.10**

Search space	Search algorithm	NinaPro	FSD50K	ECG	Satellite	DeepSEA
DenseNAS	random	6.80±0.46	**2.40±0.00**	0.18±0.05	0.79±0.16	0.25±0.04
DenseNAS	original	6.69±0.53	1.45±0.00	0.11±0.05	1.08±0.63	0.19±0.00
DARTS	GAEA	3.35±0.48	0.81±0.11	3.31±0.07	**3.35±0.35**	2.91±0.47
AMBER	ENAS	n/a	n/a	6.61±0.33	6.22±1.36	8.44±1.47
WRN	default	**2.75**	2.80	0.50	0.51	0.51
Expert	default	1.36	0.35	**16.5**	0.48	**60.9**

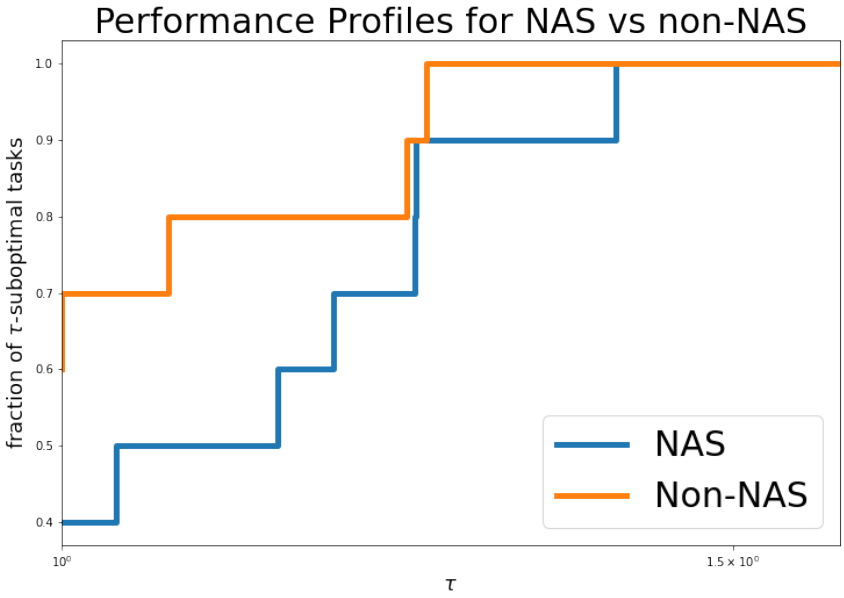

Figure 6: Performance profiles on all tasks for best-performing NAS vs. Non-NAS. The y-value indicates the fraction of tasks on which a plotted method's error is within a multiplicative factor τ of the lowest error achieved by all plotted methods..

Table 8: FLOPS of searched and baseline models for all tasks of NAS-Bench-360. Searched model FLOPS are reported as mean±standard deviation of three random seeds. Results are reported in GFLOPS. Architectures with the best performance are bolded.

Search space	Search algorithm	CIFAR-100	Spherical	Darcy Flow	PSICOV	Cosmic
DenseNAS	random	0.46±0.07	0.91±0.07	14.42±2.58	39.80±5.09	8.42±2.11
DenseNAS	original	0.44±0.53	1.84±0.15	5.43±1.82	30.51±11.90	5.00±5.30
DARTS	GAEA	1.42±0.09	**1.91±0.65**	9.33±1.13	**17.74±1.68**	14.27±4.90
Auto-DL	DARTS	n/a	n/a	2.54±1.20	3.43±1.27	2.44±0.26
WRN	default	0.78	2.78	39.72	90.58	90.06
Expert	default	**1.18**	n/a	**n/a**	0.01	**1.96**

Search space	Search algorithm	NinaPro	FSD50K	ECG	Satellite	DeepSEA
DenseNAS	random	1.02±0.06	**0.40±0.00**	0.11±0.02	0.02±0.01	0.15±0.02
DenseNAS	original	0.97±0.14	0.80±0.00	0.16±0.03	0.02±0.01	0.10±0.00
DARTS	GAEA	0.89±0.12	2.57±0.47	2.28±0.05	**0.11±0.07**	2.01±0.33
AMBER	ENAS	n/a	n/a	0.03±0.01	0.03±0.01	0.04±0.01
WRN	default	**0.64**	7.56	1.02	0.04	1.02
Expert	default	0.02	0.66	**0.70**	0.01	**0.12**

*some expert models contain non-standard modules without FLOPS count.

F Ethics and Responsible Use

Within our array of tasks, only NinaPro, ECG, and DeepSEA contain human-derived data. Our chosen subset of NinaPro contains only muscle movement data without any exposure of personal information. The original experiments to acquire NinaPro data are approved by the ethics commission of the canton of Valais, Switzerland [4]. The ECG data derives from an open challenge and is provided by the medical device company AliveCor, under the GPL license allowing it for public use. The DeepSEA data derived from ENCODE is part of an international collaborative effort, which is overseen and

funded by the National Human Genome Research Institute (NHGRI). For other datasets, we have listed the data licenses in the appendix. While we do not view the specific datasets in NAS-Bench-360 as potential candidates for misuse, the broader goal of applying NAS to new domains comes with inherent risks that may require mitigation on an application-by-application basis.