

## 深層学習モデルの構造多様性を考慮した Neural Architecture Search Neural Architecture Search for Structural Diversity in Deep Learning Models

針谷 亘輝<sup>1)</sup> 西原 慧<sup>1)</sup> 中田 雅也<sup>1)</sup>  
Nobuki Hariya Kei Nishihara Masaya Nakata

### 1 はじめに

一般に、深層学習の性能は、深層学習モデルの構造に大きく依存することが知られているが、最適な深層学習モデルを手動で設計することは困難である [1, 2, 3]。この課題に対処するために、深層学習モデルの構造を自動で最適化する Neural Architecture Search (NAS) [4] の研究が盛んに行われている。一般的な NAS は、深層学習モデルを構成する畳み込み層やプーリング層などの適用順序を最適化する方法が主流である。特に、強力なブラックボックス最適化手法である進化計算を用いた NAS (Evolutionary NAS: ENAS) は、画像分類や物体検出において、NAS を用いて自動設計した深層学習モデルを用いることで、分類・検出精度が大幅に改善することが示されている [5, 6]。本稿では、CNN [3] を対象とした ENAS に焦点を当てる。

近年の ENAS は、深層学習モデルの最適化に要する時間を削減することを目的に、バイズ最適化などの高効率なサンプリング手法を用いた方法が数多く提案されている [4, 7]。この方法は、サロゲート ENAS とも呼ばれる [8]。この背景として、1つの深層学習モデルの性能評価(分類精度の算出)に長時間を要することに加え、一般的な NAS では最適化に要する試行数(深層学習モデルを学習し分類精度を算出する回数)が多くなることが挙げられる。例えば、ENAS の一種である LargeEvo [9] では、1台の GPU を用いて理論上 3150 日の計算時間を必要とする。

具体的にサロゲート ENAS は次の 3つの手順に従う。まず、ガウス過程などの機械学習を用いて、評価済みの深層学習モデルを学習データとし、それらの分類精度を推定するサロゲートを構築する。次に、サロゲートの出力から計算されるサンプリング基準、例えば期待改善度(Expected Improvement)などを用いることで、分類精度の改善が見込める深層学習モデルを選択する。最後に、選択した深層学習モデルを GPU を用いて実際に評価する。したがって、ENAS の性能を改善するためには、サロゲートの高精度化と適切なサンプリング基準の設計が肝要となる [10]。

サンプリング基準の設計は、サロゲートにおける探索と活用のバランスを決定することに相当する。分類精度の推定値の改善を重視する基準は、深層学習モデルの分類精度が早期に改善する。また、多様な深層学習モデルを選択する基準、例えば、推定値の分散(不確実性)などを考慮する場合は、収束速度は低下するが初期収束の

リスクを軽減でき、サロゲートの推定精度も改善する効果を生む。一方で、最新の ENAS は、分類精度を重視するサンプリング基準を用いている。具体的には、最新の ENAS である BANANAS [7] は、分類精度の推定値、ならびに、その推定値の分散をもとに Independent Thompson Sampling (ITS) を用いて、実評価する深層学習モデルを選択する。しかし、この方法は、実際には分類精度の推定値に重点を置いたサンプリングとなることが報告されている [10]。NPENAS [11] では、ニューラルネットワークを用いたサロゲートを構築し、分類精度の推定値が最も高い深層学習モデルをサンプリング基準とする。

しかしながら、分類精度の推定値を特に重視するサンプリング基準を用いると、類似する構造を持つ深層学習モデルが選択される可能性が高くなる。そして、類似する深層学習モデルは互いに似た分類精度となる傾向があるため [12, 13]、初期収束の問題が発生し分類精度の改善率が低下する。この問題は、特に大規模な深層学習モデルを最適化する際により顕著になる。

そこで本稿では、深層学習モデルの構造的多様性を考慮した適応的サンプリング基準を提案し、この基準に基づく ENAS を導入する。具体的には、新たに生成された深層学習モデルの中でもモデル構造が、異なるものを優先的に選択することで、探索と活用のバランスに優れたサンプリング基準を実現する。モデル構造の違いを検出するために、深層学習モデルを実数値ベクトルに変換する Graph2vec[14] を用いる。そして、変換された実数値ベクトル間でコサイン類似度をもとめることで、構造が異なる深層学習モデルを検出する。加えて、サロゲートの推定精度に応じて、分類精度の推定値と構造的多様性の重要度を適応的に変更することで、深層学習モデルの分類精度の早期改善も同時に達成する。この適応的サンプリング基準を NPENAS に組み込むことで、構造的多様性を考慮した ENAS を構築する。本稿の貢献は以下の通りである。

- 著者らの知る限り、深層学習モデルの構造的多様性を考慮したサンプリング基準を初めて導入する。なお、BANANAS で用いる不確実性は、分類精度の推定値に関するものであり、モデル構造の違いに関する指標とはならないことに留意されたい。
- サンプリングに用いる指標(分類精度の推定値と構造的多様性の指標)を適応的に変更することで、分類精度の収束速度を改善しつつ初期収束の問題も回避できることを明らかにする。特に、提案手法は、ベンチマーク問題である NASBench-101 [6]、NASBench-201 [15] と DARTS 探索空間 [16] において

1) 横浜国立大学 理工学府 Graduate School of Engineering Science, Yokohama National University

最先端の性能を達成することを示す。

本稿の構成は次の通りである。第 2 章では要素技術として NPENAS について概説する。第 3 章では提案手法のアルゴリズムを説明する。第 4 章ではベンチマーク問題を用いて提案手法の有効性を評価する。第 5 章では、構造的多様性の指標、ならびに、適応的サンプリング基準の有効性を検証する。最後に、第 6 章で本稿の結論を述べる。

## 2 NPENAS

NPENAS は、ベンチマーク問題において最先端性能を導出した ENAS の 1 種である。NPENAS の特徴は、サロゲートにグラフニューラルネットワークの 1 種である Graph Isomorphism Network (GIN) [17] を採用することにある。NPENAS では、GIN を用いたサロゲートを Neural Predictor と呼ぶ。BANANAS などの多くの既存手法では、深層学習モデルを符号ベクトルに変換し、これを入力とするサロゲートを構築する。この方法に比べ、Neural Predictor は深層学習モデルの構造をサロゲートに直接入力できるため、サロゲートの高精度化が達成できる。

Algorithm 1 に NPENAS の疑似アルゴリズムを示す。まず、 $n_0$  個の深層学習モデルをランダムに生成し、各モデルを所与の学習データで訓練したのち検証データで分類精度を評価する。これらの深層学習モデルをサロゲートの学習データとして  $D$  に格納する。次に、メインループとして、まず、Neural Predictor  $G$  を構築する。Neural Predictor に含まれる重みパラメータをランダムに初期化した後、学習データ  $D$  を用いて  $G$  を訓練する。次に、 $D$  から選択した  $\mu_{num}$  個の深層学習モデルに one-to-many 突然変異戦略を適用することで、 $\mu_{num}$  個の深層学習モデルを新たに生成する。次に、生成した深層学習モデルを  $G$  に入力することで検証誤差の推定値を得る。そして、検証誤差が小さい順に  $t$  個の深層学習モデルを選択し、所与の学習データで訓練したのち検証データで分類精度を評価する。これらの  $t$  個のモデルを  $D$  に追加する。以上の操作を評価回数  $n$  が  $n_{total}$  に達するまで繰り返す。

## 3 提案手法

構造的多様性を考慮した適応的サンプリング基準を NPENAS に組み込むことで提案手法を構築する。本章では、まず提案手法の全体アルゴリズムを述べた後に、構造的多様性の算出方法、ならびに、これを用いた適応的サンプリング基準について詳述する。

### 3.1 全体アルゴリズム

提案手法では、サロゲートの推定精度に応じて、NPENAS で用いる検証誤差を用いたサンプリング基準と、新たに導入する構造的多様性を用いた基準を適応的に選択する。構造的多様性を用いた基準では、Graph2vec を用いて深層学習モデルを実数値ベクトルに

### Algorithm 1 NPENAS

- 1: **Input:** initial population size  $n_0$ , the number of total training samples  $n_{total}$ , the number of candidate models  $\mu_{num}$ , and the number of offspring to be sampled  $t$
- 2: Randomly generate  $n_0$  initial models  $\{s_i\}_{i=1}^{n_0}$ , and evaluate their validation error  $\{y_i\}_{i=1}^{n_0}$
- 3: Initialize a dataset with initial models,  $D = \{(s_i, y_i)\}_{i=1}^{n_0}$
- 4:  $n = n_0$
- 5: **while**  $n < n_{total}$  **do**
- 6:   Randomly initialize the parameters of neural predictor  $G$
- 7:   Train the neural predictor  $G$  with dataset  $D$
- 8:   Generate  $\mu_{num}$  candidate models from  $D$  by using one-to-many mutation strategy, forming a candidate set  $M = \{\hat{s}_i\}_{i=1}^{\mu_{num}}$ .
- 9:   Calculate the estimated validation error of each candidate model  $\hat{s}_i$  in  $M$  by  $G$ .
- 10:   Select  $t$  candidate models having the  $t$  minimum estimated validation errors, and evaluate their actual validation errors
- 11:   Append  $t$  evaluated candidate models to  $D$
- 12:    $n \leftarrow n + t$
- 13: **Return** the best model having the minimum validation error

変換することで、同モデルの構造上の違いを検出する。例えば、提案手法は図 1 で表される深層学習モデル (a) と同モデル (b) は構造上類似するが (青色四角内の構造だけが異なる)、深層学習モデル (a) と同モデル (c) は構造上大きく異なる。Algorithm 2 に提案手法のアルゴリズムを示す。なお、*diversity\_sampling* は、使用するサンプリング基準を決定する変数であり、*false* であれば検証誤差を用いた基準、*true* であれば構造的多様性を用いた基準を適用する。

NPENAS の初期化に加えて、*diversity\_sampling = false* と初期化する。次に、ランダムに生成した構造を持つ  $K$  の深層学習モデルを生成し、これらを用いて Graph2vec  $h$  を訓練する。なお、Graph2vec は深層学習モデルを実数値ベクトルに変換する処理のみで用いるため、初期化時にのみ一度構築するだけで良い。次に、one-to-many 突然変異を使用して  $\mu_{num}$  個の深層学習モデルを新たに生成する操作まで NPENAS と同じ処理を行う。*diversity\_sampling* が *false* であれば、NPENAS と同様に検証誤差が小さい順に  $t$  個の深層学習モデルを選択し、所与の学習データで訓練したのちテストデータで検証誤差を評価する。*diversity\_sampling* が *true* であれば、後述する構造的多様性に関する指標を計算し、多様性指標が大きい順に  $d$  個の深層学習モデルを選択し、訓練ならびに検証誤差を評価する。加えて、残り  $t - d$  個の深層学習モデルは、検証誤差が小さい順に  $t$  個を選択する。そして、評価した合計  $t$  個のモデルを  $D$  に追加する。以上の操作を評価回数  $n$  が  $n_{total}$  に達するまで繰り返す。

### 3.2 構造的多様性の算出方法

構造的多様性を考慮したサンプリングでは、学習データ  $D$  内に格納された既に評価済みの深層学習モデルと、最も構造が異なるモデルを選択するように設計する。このために、新たに生成された深層学習モデル  $\mu \in M$  に対する多様性  $v$  を Graph2vec モデル  $h$  とコサイン類似度の分散を用いて定義する。具体的には、 $D$  の全深層学習

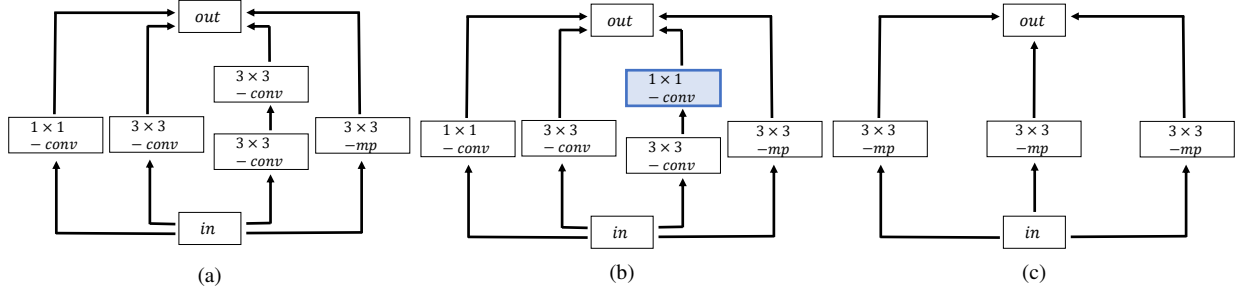


図 1 深層学習モデルの構造的多様性の例

モデル  $g$  を用いて  $v$  を次式で計算する。

$$v = \frac{1}{|D|} \sum_{g \in D} (\cos(h(g), h(\mu)) - \bar{v})^2 \quad (1)$$

$$\bar{v} = \frac{1}{|D|} \sum_{g \in D} \cos(h(g), h(\mu)) \quad (2)$$

ただし、 $h(g), h(\mu)$  は、 $h$  を用いて変換したモデル  $g, h$  の実数値ベクトルである。

前節で概説したように、 $M$  内の全ての深層学習モデルについて多様性  $v$  を求め、大きい順に  $d$  個の深層学習モデルを選択し実際に評価する。

### 3.3 適応的なサンプリング基準の変更

推定精度が高く信頼性の高い Neural Predictor が構築できたとき、NPENAS のように検証誤差の推定値が小さい深層学習モデルを優先的に選択することが好ましい。一方で、推定精度が低い場合は、構造的多様性を考慮したサンプリング基準を用いて、異なる構造を持つ深層学習モデル（学習データ）を選択することで、初期収束の問題を回避するとともに、Neural Predictor の推定精度の改善も見込める。このことから、Neural Predictor の推定精度に応じて、検証誤差を用いたサンプリング基準と新たに導入する構造的多様性を用いた基準を適応的に選択する。

まず、所定のサンプリング基準によって選択された  $t-d$  個の深層学習モデル  $\mu_1^*, \dots, \mu_{t-d}^*$  に対して、それらを実際に評価し検証誤差  $\epsilon_1^*, \dots, \epsilon_{t-d}^*$  が計算された状態を考える。これらのデータ  $\{(\mu_i^*, \epsilon_i^*)\}_{i=1}^{t-d}$  を Neural Predictor の推定精度を評価するため検証データとする。各モデル  $\mu_i^*$  を Neural Predictor に入力することで、検証誤差の推定値  $\hat{\epsilon}_i^*$  を求め、 $\{(\mu_i^*, \hat{\epsilon}_i^*)\}_{i=1}^{t-d}$  を得る。そして、 $\{(\mu_i^*, \epsilon_i^*)\}_{i=1}^{t-d}$ 、ならびに、 $\{(\mu_i^*, \hat{\epsilon}_i^*)\}_{i=1}^{t-d}$  に対し、ケンドールの順位相関係数  $K_\tau$  [18] を計算する。なお、 $K_\tau$  が大きい値であるほど、Neural Predictor の推定精度が高いことを意味する。また、 $K_\tau \leq 0$  であれば逆相関を意味し、Neural Predictor の信頼性が極めて低い。以上より、下式に示すように、 $K_\tau > 0$  であれば、*diversity sampling* を *true* とし、検証誤差を用いたサンプリング基準を用いる。そのほかの場合は、構造的多様性を用いたサンプリング基準を用いる。

### Algorithm 2 提案手法

- 1: **Input:** initial population size  $n_0$ , the number of total training samples  $n_{total}$ , the number of candidate models  $\mu_{num}$ , the number of offspring to be sampled  $t$ , and the number of offspring for diversity-based sampling  $d$
- 2: Randomly generate  $n_0$  initial models  $\{s_i\}_{i=1}^{n_0}$ , and evaluate their validation error  $\{y_i\}_{i=1}^{n_0}$
- 3: Initialize a dataset with initial models,  $D = \{(s_i, y_i)\}_{i=1}^{n_0}$
- 4: Generate a training dataset for Graph2vec  $h$  by randomly generating  $K$  models, and train  $h$
- 5:  $n = n_0$
- 6: *diversity sampling = false*
- 7: **while**  $n < n_{total}$  **do**
- 8: Randomly initialize the parameters of neural predictor  $G$
- 9: Train the neural predictor  $G$  with dataset  $D$
- 10: Generate  $\mu_{num}$  candidate models from  $D$  by using one-to-many mutation strategy, forming a candidate set  $M = \{\hat{s}_i\}_{i=1}^{\mu_{num}}$ .
- 11: Calculate the estimated validation error of each candidate model  $\hat{s}_i$  in  $M$  by  $G$ .
- 12: **if** *diversity sampling = true* **then**
- 13: Calculate the diversity score of each candidate model  $\hat{s}_i$  in  $M$  by Eq. 2.
- 14: Select  $d$  candidate models having  $d$  maximum diversity scores from  $M$ , and evaluate their actual validation errors
- 15: Select  $t-d$  candidate models having the  $t-d$  minimum estimated validation errors from  $M$ , and evaluate their actual validation errors
- 16: **else**
- 17: Select  $t$  candidate models having the  $t$  minimum estimated validation errors from  $M$ , and evaluate their actual validation errors
- 18: Append  $t$  evaluated candidate models to  $D$
- 19: Update *diversity sampling* by Eq. (3)
- 20:  $n \leftarrow n + t$
- 21: **Return** the best model having the minimum validation error

$$sampling\_strategy = \begin{cases} true & \text{if } K_\tau > 0, \\ false & \text{otherwise.} \end{cases} \quad (3)$$

## 4 実験

提案手法の有効性を評価するために、3つのベンチマーク問題を用いた実験を行う。具体的には、NASBench-101 [6]、NASBench-201 [15]、DARTS [16]を用いる。それぞれのベンチマーク問題では、CIFAR-10 [19]のデータセットをもとに深層学習モデルの最適化を行う。加えて、NASBench-201では、CIFAR-10に加え、CIFAR-100 [19]、ImageNet [20]を用いた実験を行う。

表1 NASBench-101 (CIFAR-10) におけるアルゴリズムの比較

Method	Val Err (%)	Test Err (%)
RS	5.681	6.337
REA	5.539	6.140
BANANAS	5.474	6.079
NPENAS	5.274	5.884
提案手法	5.283	5.883

#### 4.1 比較手法と評価方法

NASBench-101 と NASBench201 では、ランダムサーチ (RS) [21]、正則化進化アルゴリズム (REA) [22]、BANANAS[7]、NPENAS を比較手法とする。DARTS では、NPENAS のみ比較する。提案手法のパラメータは、 $t = 10, d = 8$  とする。Graph2vec は、NASBench-101 と NASBench-201 ではグラフ構造を 8 次元のベクトル、DARTS では 16 次元のベクトルへと変換するように学習する。Graph2vec の訓練エポック数は 40 とする。その他の設定は NPENAS の原著に従う。提案手法と比較手法について、最大評価回数  $n_{total}$  は NASBench-101、NASBench-201、DARTS でそれぞれ 150、100、100 とする。

評価方法として、検証データならびにテストデータの分類誤差で評価する。NASBench-101 と NASBench201 のベンチマーク問題では独立した 400 試行、DARTS では 3 試行とする。DARTS で発見した深層学習モデルの再学習は、乱数的な重みの初期化を考慮し、各試行に対して 2 回ずつ行う。すべてのベンチマーク問題について、最大評価回数時の平均値を報告する。

#### 4.2 実験結果

● **NASBench-101** まず、NASBench-101 における検証データとテストデータの分類誤差を表 1 に示す。また、図 2 に評価回数の変化に対するテストデータでの分類誤差の平均の推移を示す。また、同図のエラーバーは 30 パーセントと 70 パーセントを示す。表 1 より、提案手法はテストデータにおいては最良値を導出しているが NPENAS と僅差である。また、図 2 より、提案手法は NPENAS とほぼ同じ収束速度となり、抜本的な改善が見られなかった。

● **NASBench-201** NASBench-201 における検証データとテストデータの分類誤差を表 2、テストデータにおける分類誤差の推移を図 3-5 に示す。CIFAR-100 においては、提案手法の性能は NPENAS に劣る。しかし、CIFAR-10、ImageNet において提案手法は検証データとテストデータともに最良の分類誤差を達成しており顕著な性能改善が見られる。加えて、図 3, 5 に示す通り、評価回数が 60 回以降で優れた分類誤差を導出し、収束速度に優れることがわかる。

● **DARTS** 最後に、DARTS におけるテストデータの分類誤差を表 3 に示す。なお、DARTS における構造探索には膨大な時間を要するため、評価回数毎に対するテストデータを計算していない。また、同様の理由で NPENAS

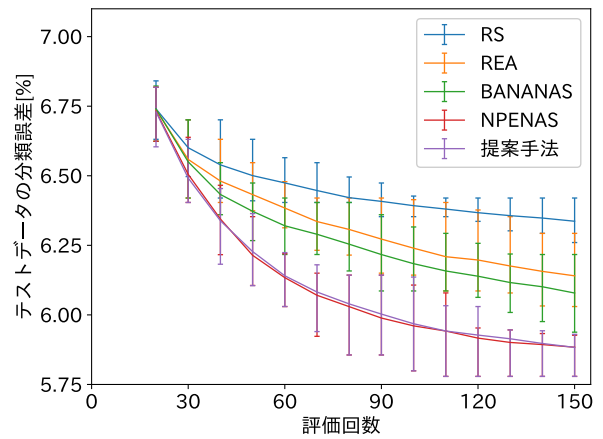


図 2 NASBench-101 の評価回数に対する性能の遷移

表 2 NASBench-201 におけるアルゴリズムの比較

a) CIFAR-10		
Methods	Val Err (%)	Test Err (%)
RS	8.961	6.210
REA	8.582	5.881
BANANAS	8.672	5.916
NPENAS	8.478	5.712
提案手法	8.442	5.684

b) CIFAR-100		
Methods	Val Err (%)	Test Err (%)
RS	28.54	28.52
REA	27.15	27.18
BANANAS	27.41	27.45
NPENAS	26.53	26.52
提案手法	26.54	26.54

c) ImageNet		
Methods	Val Err (%)	Test Err (%)
RS	54.61	54.60
REA	53.83	53.86
BANANAS	53.76	53.82
NPENAS	53.37	53.56
提案手法	53.36	53.55

と提案手法のみ実験した。一方で、公表されている他の NAS 手法の分類誤差と計算時間 (GPU Days) も表 3 に示している。表より、分類誤差の最良値は AmoebaNet-B、次点で BANANAS、次に提案手法となる。提案手法の分類誤差および計算時間は、NPENAS よりも優れている。

#### 5 分析

ここでは、構造的多様性を考慮した新たなサンプリング基準、ならびに、その適応的な調整メカニズムの効果について分析を行う。

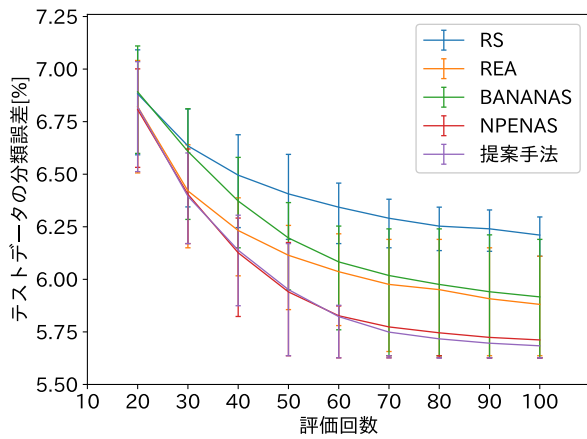


図 3 NASBench-201 の CIFAR-10 を用いた評価回数に対する性能の遷移の比較

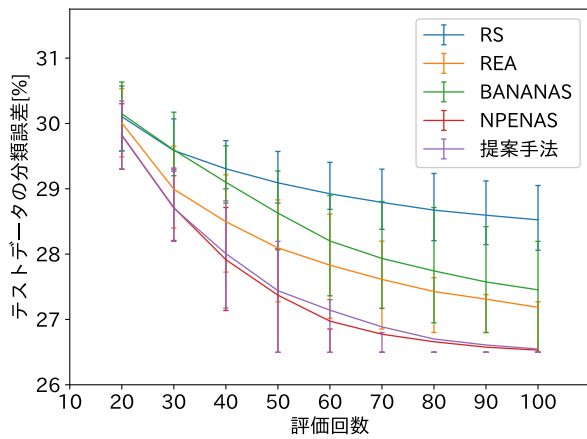


図 4 NASBench-201 の CIFAR-100 を用いた評価回数に対する性能の遷移の比較

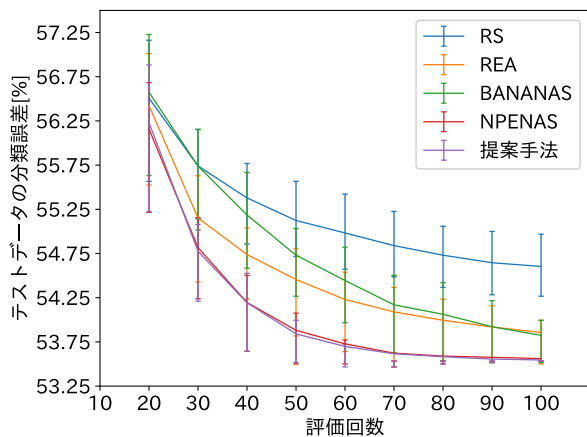


図 5 NASBench-201 の ImageNet を用いた評価回数に対する性能の遷移の比較

### 5.1 構造的多様性によるサンプリング基準の効果

提案手法では、多様な構造をもつ深層学習モデルを選択することで、Neural Predictor の推定精度の改善を意図していた。ここでは、1) Neural Predictor の推定精度、2) 選択された  $t$  個の深層学習モデルの多様性指標  $v$ 、なら

表 3 DARTS における性能比較

Methods	Test Err (%)	GPU Days
RS [16]	3.29	2.7
ENAS [23]	2.89	0.5
AmoebaNet-A [22]	3.34	3150
AmoebaNet-B [22]	2.55	3150
BANANAS [7]	2.64	11.8
NPENAS	2.85	0.63
提案手法	2.78	0.54

びに、検証誤差/構造的多様性のサンプリング基準の選択割合について分析する。これらの観点から、NPENAS、提案手法を NASBench-101 を用いて比較する。

まず、各 ENAS のサロゲートの推定精度を比較するために、最適化とは無関係な深層学習モデルを 50 個生成し、これらの検証誤差の真値とサロゲートによる推定値間のケンドールの順位と相関係数を求める。ここでは、NASBench-101 を用いる。また、深層学習モデルの多様性指標については、提案手法に加えて NPENAS でも Graph2vec モデルを構築し、選択した  $t$  個の深層学習モデルの構造多様性  $v$  の平均値を比較する。

図 6 a), b), c) に各世代における推定精度 (ケンドールの相関値) の推移、構造的多様性の平均、ならびに、サンプリング基準の使用割合を示す。同図 a) より、提案手法は、探索回数が増加し学習データが増えるにつれて推定精度が改善している。また、提案手法の構造的多様性の平均値は、NPENAS よりも高く、異なる構造を持つ深層学習モデルを多く選択していることがわかる。最後に、同図 c) より、提案手法は、探索初期で検証誤差/構造的多様性のサンプリング基準をバランスよく選択しているが、探索回数が増加するにつれて (Neural Predictor の推定精度が改善するにつれて) 徐々に検証誤差に基づく基準を用いている。

以上より、提案手法は、異なる構造を持つ深層学習モデルを多く選択することで、Neural Predictor の推定精度を改善に成功していることが判る。

### 5.2 適応的サンプリングの効果

最後に、適応的サンプリングの効果を検証するために、常に構造的多様性を考慮したサンプリング基準を用いる提案手法を導入しその性能を比較する。具体的には、*diversity sampling* を常に *true* と固定し、かつ、 $d = 10$  とした提案手法を構築し、提案手法  $_{ds}$  と表記する。NASBench-101 を用いて、1) テストデータにおける分類誤差、2) Neural Predictor の推定精度、ならびに 3) 選択された  $t$  個の深層学習モデルの多様性指標  $v$  を比較する。なお、2) と 3) については、前節と同じ方法で算出する。

分析結果を図 7 に示す。まず、提案手法に比べ、提案手法  $_{ds}$  の分類誤差は著しく低下しており、構造的多様性を考慮したサンプリング基準のみを用いても NPENAS の性能改善に寄与しないことがわかる。一方で、提案手法  $_{ds}$  が構築した Neural Predictor の推定精度は大幅に向上している。また、図 7 c) より、提案手法

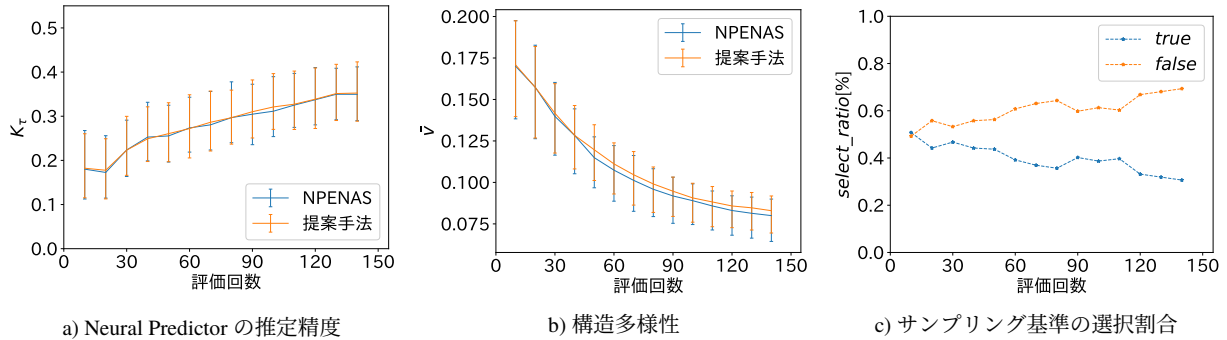


図6 NASBench-101 における分析結果

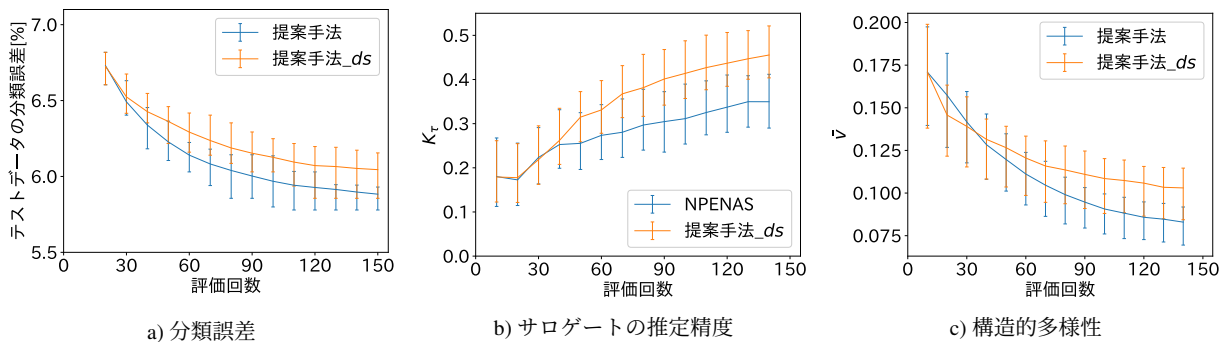


図7 NASBench-101 における分析結果 (適応的サンプルング基準の効果)

$ds$  の構造的な多様性の平均値は高く、異なる構造を持つ深層学習モデルを多く選択していることがわかる。

提案手法  $ds$  が構築した Neural Predictor の推定精度が改善した理由としては、多様な深層学習モデルを学習データとしたことで、探索範囲をより広範に推定できるサロゲートが構築できたことが理由として考えられる。一方で、サロゲートを用いた進化的最適化では、広範な領域を扱うグローバルサロゲートは、局所領域に対する推定精度に劣り、結果として最適化性能が低下することが知られている [24]。したがって、適応的なサンプルング基準によって、探索と活用のバランスをとることで、適度に局所化されたサロゲートを構築できたことが提案手法の性能改善の要因と考えられる。

## 6 結論

本稿では、深層学習モデルの構造的な多様性を考慮した適応的なサンプルング基準に基づく NPENAS を提案した。提案手法は、類似する深層学習モデルは類似する性能となる知見を活用し、これを回避するために、モデルの構造的な多様性に着目したサンプルング基準を設計した。実験では、NASBench-101、NASBench-201(CIFAR-10, ImageNet) ならびに DARTS 探索空間を用いたベンチマーク問題において、最先端の ENAS 手法よりも優れた性能を導出することを示した。分析では、構造的な多様性を考慮したサンプルング基準がサロゲートの推定精度の改善に対し貢献することを示した。

今後の予定として、ハイパーパラメータ  $K_\tau$  や  $d$  の解析を行う。

## 参考文献

- [1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [4] S. Liu, H. Zhang, and Y. Jin, "A survey on surrogate-assisted efficient neural architecture search," *arXiv preprint arXiv:2206.01520*, 2022.
- [5] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.
- [6] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter, "Nas-bench-101: Towards reproducible neural architecture search," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7105–7114.
- [7] C. White, W. Neiswanger, and Y. Savani, "Bananas: Bayesian optimization with neural architectures for neural architecture search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10293–10301.
- [8] X. Gao, B. Xiao, D. Tao, and X. Li, "A survey of graph edit distance," *Pattern Analysis and applications*, vol. 13, pp. 113–129, 2010.
- [9] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2902–2911.

- [10] L. Fan and H. Wang, "Surrogate-assisted evolutionary neural architecture search with network embedding," *Complex & Intelligent Systems*, pp. 1–19, 2022.
- [11] C. Wei, C. Niu, Y. Tang, Y. Wang, H. Hu, and J. Liang, "Npenas: Neural predictor guided evolution for neural architecture search," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [12] A. Zela, A. Klein, S. Falkner, and F. Hutter, "Towards automated deep learning: Efficient joint neural architecture and hyperparameter search," *arXiv preprint arXiv:1807.06906*, 2018.
- [13] J. You, J. Leskovec, K. He, and S. Xie, "Graph structure of neural networks," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 10 881–10 891. [Online]. Available: <https://proceedings.mlr.press/v119/you20b.html>
- [14] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "graph2vec: Learning distributed representations of graphs," *arXiv preprint arXiv:1707.05005*, 2017.
- [15] X. Dong and Y. Yang, "Nas-bench-201: Extending the scope of reproducible neural architecture search," *arXiv preprint arXiv:2001.00326*, 2020.
- [16] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.
- [17] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.
- [18] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.
- [19] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [21] L. Li and A. Talwalkar, "Random search and reproducibility for neural architecture search," 2019.
- [22] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4780–4789.
- [23] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *International conference on machine learning*. PMLR, 2018, pp. 4095–4104.
- [24] T. Sonoda and M. Nakata, "Multiple classifiers-assisted evolutionary algorithm based on decomposition for high-dimensional multiobjective problems," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 6, pp. 1581–1595, 2022.