

Approach to Efficient Neural Architecture Search on the Latent Space of VGAE Models with Multiple Datasets

逸見 一喜^{†,††} 谷垣 勇輝^{††} 大西 正輝^{††,†}

[†] 筑波大学 〒305-8577 茨城県つくば市天王台 1-1-1

^{††} 産業技術総合研究所 〒305-8560 茨城県つくば市梅園 1-1-1

E-mail: [†]kazuki.one.jack.work@gmail.com, ^{††}tanigaki.yuki@aist.go.jp, ^{†††}tonishi@ni.aist.go.jp

あらまし Neural Architecture Search (NAS) は、データや目的に応じてニューラルネットワークのアーキテクチャを自動で最適化する AutoML 手法の一つである。しかし、NAS は膨大な探索空間により高い計算コストが必要であり、新たなデータセットに適したアーキテクチャを 0 から探索することは非効率である。そこで本研究では、既存データセットに適したアーキテクチャを潜在空間にマッピングすることで、未知のデータセットに適したアーキテクチャを生成するフレームワークを提案する。また、従来の NAS は主に単一データセットの精度を最適化するが、実世界では異なる特性を持つ複数のデータセットに対して適応するアーキテクチャが重要であるため、複数のデータセットに適したアーキテクチャのトレードオフを特徴量空間に落としこむ方法も検討する。NAS のアーキテクチャがグラフ構造で表現できることから、潜在特徴量のマッピングには Variational Graph Auto-Encoders に基づく Encoder-Decoder 変換モデルを用いる。実験の結果、提案手法は従来法に比べて高い探索効率と優れた性能を示した。

キーワード Neural Architecture Search, AutoML, 多目的最適化, Graph Neural Network

1 序 論

近年、データセットやタスクに合わせて最適なニューラルネットワークのアーキテクチャを自動的に探索する Neural Architecture Search (NAS) は AutoML (自動機械学習) の領域において重要な技術となっている。NAS が登場する以前はアーキテクチャの構築は基本的にハンドクラフトな方法で行われており、性能向上のためには試行錯誤を繰り返す必要があった。特に、ハンドクラフトなアーキテクチャの構築は深層学習の専門的な知識が必要であり、膨大な時間が必要であった。しかし、NAS の登場によりアーキテクチャ探索の試行錯誤を自動的に行うことで、深層学習の専門的な知識を有していなくても高性能なネットワークアーキテクチャが容易に構築可能となった。一方で NAS はネットワークアーキテクチャの探索にかかる時間が大きいという欠点がある。

NAS を用いた最適なアーキテクチャの探索は大量の計算リソースと時間を必要とする。よって、新たなデータセットが与えられるたびに適したアーキテクチャを 0 から探索することは非効率である。今後予想されるデータセットの増加に対応するためには、未知のデータセットに最適なアーキテクチャを直接推測する技術が必要となる。そこで、私たちは既存のデータセットに適したアーキテクチャを潜在特徴量空間にマッピングすることで、NAS を行わずに未知のデータセットに適したアーキテクチャを生成するフレームワーク Variational Graph Auto-Encoders Neural Architecture Search (VGAE-NAS) を提案する。潜在特徴量のマッピングには、VGAE に基づいた Encoder-Decoder 変換モデルを用いる。VGAE は Encoder と Decoder を用いてグラフの特徴量抽出を行う手法であり、グラ

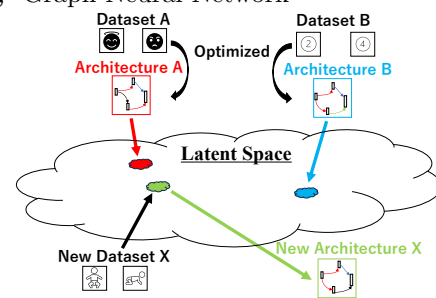


図 1: 本研究のコンセプト、データセット A, B を基に、それぞれアーキテクチャ A, B を最適化し構築する。最適化されたアーキテクチャ A, B は潜在特徴量空間にマッピングされる。新しいデータセット X が生成されたとき、そのデータセットの特徴を考慮した最適なアーキテクチャ X が潜在空間から瞬時に生成される。

フ構造で表現される NAS のアーキテクチャのマッピングに適している。入力するアーキテクチャと再構成されたアーキテクチャの誤差を減らすことで、潜在空間の潜在特徴量がアーキテクチャの特徴を表現する。マッピング後の潜在特徴量空間から VGAE の Decoder により直接アーキテクチャを生成することで、NAS を実行せずとも新たなデータセットに対して有望なアーキテクチャを生成することができる。新たなデータセットに対して 0 からアーキテクチャを探索する計算コストを大幅に削減できるため、NAS におけるアーキテクチャ情報の転移は非常に重要である。提案手法は多様なデータセットに対する最適なアーキテクチャの構築を目指した手法である。また、従来の NAS は主に単一データセットの精度を最適化するが、実世界では異なる特性を持つ複数のデータセットに対して適応する

アーキテクチャが重要であるため、複数のデータセットに適したアーキテクチャのトレードオフを特徴量空間に落としこむ方法も検討する。本研究の概要を図 1 に示す。

本研究の構成は以下の通りである。第 2 節はベースとなる NAS や GNN、関連手法について説明し、第 3 節では提案手法の詳細について述べる。第 4 節は提案手法を用いた実験設定とその結果について考察を行う。そして第 5 節にて本研究の結論を述べる。

2 関連研究

本節では、本研究で扱う NAS 手法や GNN について述べる。

2.1 Neural Architecture Search (NAS)

NAS は AutoML (自動機械学習) の一分野であり、深層学習で扱う最適なニューラルネットワークのアーキテクチャを自動的に探索する最適化手法である。NAS では主に層 (Layer) の繋ぎ方と演算 (Operation) の選択といった 2 つの問題を対象に最適化することでネットワークアーキテクチャを探索する。進化的アルゴリズムや強化学習で最適化する手法 [1] [2] [3] [4] は NAS が登場した頃に提唱されたが、計算コストが大きく時間がかかる手法であるため工夫が必要となる。NAS を用いたアーキテクチャの探索はコストのかかるアプローチであるため、効率的なアーキテクチャ表現が求められており、Cell ベースの NAS 手法が主流な解決方法として登場した。Cell ベースの NAS 手法の中でも効率の良い構造探索を可能にした DARTS [5] や PC-DARTS [6] は現在でも主流な NAS 手法である。ここで、ネットワークアーキテクチャは、コンパクトなモジュールの集合体として定義され、各モジュールは有向非巡回グラフとして表現される。これにより、単純な探索空間内で大規模なアーキテクチャを作成することが容易となる。近年、Cell ベースの NAS アルゴリズム [7] による高精度なアーキテクチャが数多く発見されており、探索された高精度なアーキテクチャを再利用することが今後必要になる。

2.2 Graph Neural Network (GNN)

GNN [8] はグラフ構造のデータに深層学習の仕組みを組み込んだニューラルネットワークである。GNN の中にはグラフ分類やエッジ予測といったタスクがある。ノードとエッジで表現可能な構造であれば GNN の手法で扱うことが可能であるため、GNN は化学の分子構造 [9] [10] や物理の多体粒子シミュレーション [11] [12] などに応用されている。本研究では NAS の出力がグラフ構造で表現できるため、入力として扱えると考え GNN 手法を採用する。多くの GNN では Graph Convolution Network (GCN) [13] と呼ばれる畳み込み演算が使用される。特に GCN では式 (1) を用いて、隣接行列や次数行列を扱うことで局所的なグラフの構造学習に成功している。

$$\mathbf{F}_N^{(l+1)} = h(\mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} \mathbf{F}_N^{(l)} \mathbf{W}^{(l)}) \quad (1)$$

ここで、 $\mathbf{F}_N$ が層の入出力でありノードの特徴量を表す。 $\mathbf{A}$ はエッジの繋ぎ方を示す無向グラフの隣接行列と単位行列を足

し合わせた行列、 $\mathbf{W}$ は学習可能な重み行列、 $\mathbf{D}$ はノードの関係を示す次数行列、 h は活性化関数 (Softmax と ReLU を使い分ける) となっている。その他の代表的な GNN の演算として Graph Attention Network (GAT) [14] が挙げられる。GAT は GCN に複数の Attention 機構を組み込むことにより、ノード間の重要度に対して重みづけを行い表現力が向上した演算子である。Variational Graph Auto-Encoders (VGAE) [15] は GNN の一手法であり、Encoder と Decoder を用いてグラフの特徴量抽出を行うことができる。画像を扱う VAE [16] を拡張することで、グラフの表現を可能とした手法となっている。詳しい Encoder と Decoder の詳細については次節で示す。

2.3 関連手法と提案手法の特徴

本研究に類似したような手法はいくつか存在する。提案手法は以下の 5 つの特徴を持っている：

- (a) Alleviating computational cost and search time
- (b) Building the latent feature space using VGAE
- (c) Capable of handling continuous architectures
- (d) Directly generated architectures
- (e) Enabling information transfer in existing datasets

NAS は高い探索時間と計算コストが問題点として取り上げられることが多く、(a) のような大幅に計算コストを削減する研究 [17] [18] [19] が多く取り組まれている。また、アーキテクチャに対して特徴量を取り出し、潜在特徴量空間を構築する関連手法 [20] [21] [22] [23] [24] [25] [26] [27] [28] が数多く存在する。その中でも提案手法のように VGAE を使用し特徴量を取り出す手法 [21] [26] [27] [28] は一部であり、他と差別化できる点 (b) である。VGAE を用いることでグラフを介して細かいアーキテクチャの特徴量や性質を捉えることが可能となる。関連手法としては、精度等を予測する機構である predictor を構築する手法 [29] [30] [22] [23] [24] [31] [32] [33] もあるが、提案手法は直接潜在特徴量空間からアーキテクチャを生成 (d) する。また、提案手法は演算子の埋め込み表現も同時に学習することで離散的なアーキテクチャだけでなく、連続的なアーキテクチャも対象として扱うことが可能 (c) である。有望な機知情報を利用 (e) し性能を上げる点では、転移学習のアイデア [7] [34] [35] [36] も類似していると言える。転移学習の手法では、主にデータセット情報の転移に焦点を当てているが、提案手法ではアーキテクチャそのものの情報転移に特化している。関連手法の多くは主に精度向上を目指した手法であるのに対し、本研究では様々なデータセットに対する構造の調査を目的の 1 つとする。

単一の特徴を持つ手法だけでなく、複数の特徴を持つ手法も関連手法の中には存在する。しかし、提案手法は (a)～(e) の 5 つの特徴を全て持っており、これは関連研究と大きく異なる部分である。5 つの特徴はアーキテクチャの生成において全て重要である。さらに本研究の特徴として、改良した VGAE を使うことで、NAS で探索したアーキテクチャの特徴を詳しく捉えることが可能となる。複数のデータセットにおいて高い精度を達成可能なアーキテクチャの構築を目指す部分も本研究の特徴である。

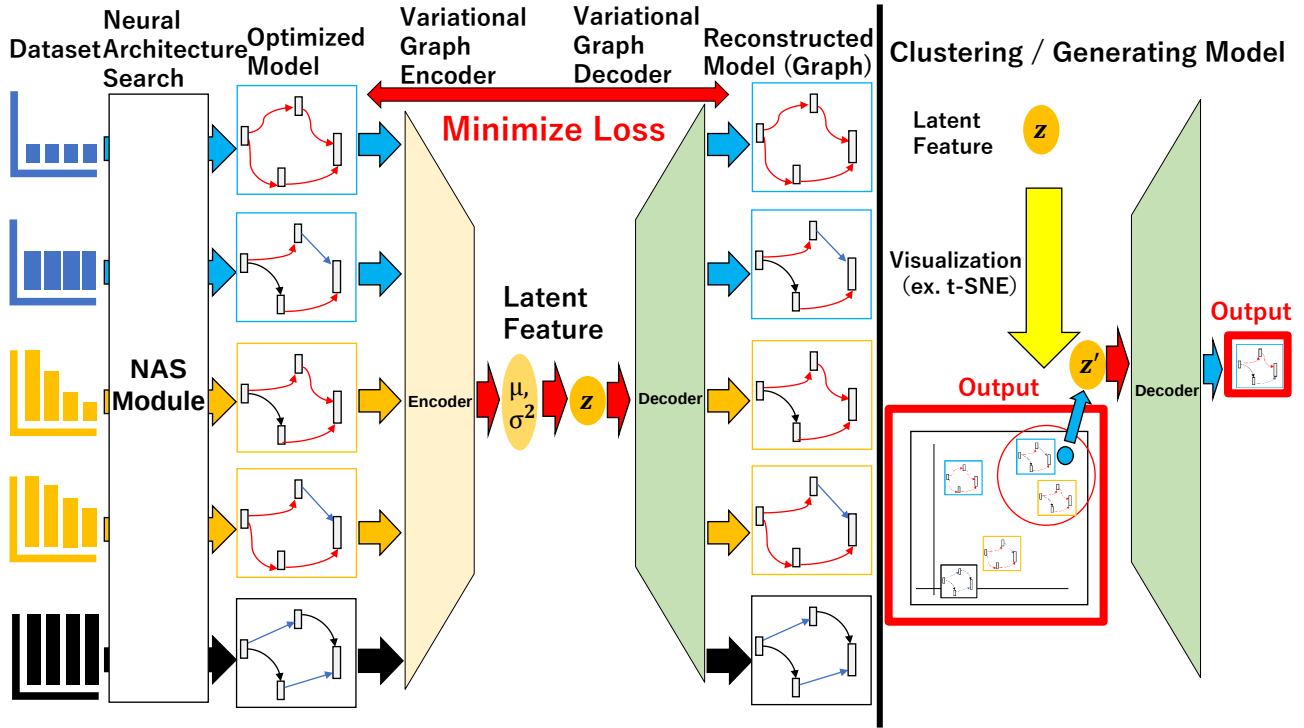


図 2: 提案手法：VGAE-NAS の概要

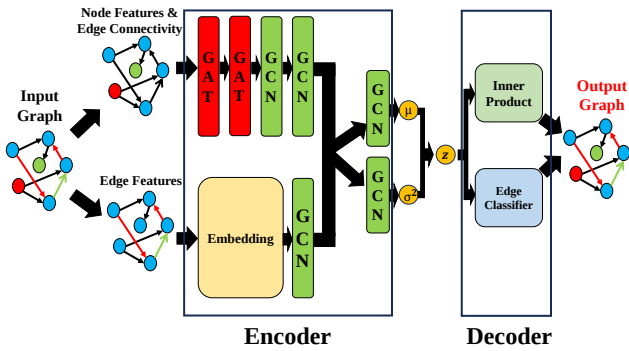


図 3: Encoder と Decoder の詳細. GAT: Graph Attention Network, GCN: Graph Convolution Network, Edge Classifier: ReLU と 2 つの全結合層

3 Variational Graph Auto-Encoders Neural Architecture Search

本研究では、NAS と VGAE を組み合わせて潜在特徴量を取り出し、アーキテクチャの特徴を獲得・可視化する手法である Variational Graph Auto-Encoders Neural Architecture Search (VGAE-NAS) を提案する。VGAE-NAS は潜在特徴量空間からアーキテクチャを生成する機構も含む。

3.1 提案手法の概要

VGAE-NAS の概要図を図 2, Encoder と Decoder の詳細を図 3, VGAE-NAS のアルゴリズムを Alg. 1 に示す。

VGAE-NAS は NAS, VGAE, Generating Model の 3 種類のコンポーネントで構成されており、Alg. 1 の手順 (NAS :

Line 1-5, VGAE : Line 6-15, Generating Model : Line 16-19) で実行される。図 2 に示すように、潜在特徴量を獲得した後は可視化手法によりクラスタリングすることや新たなモデルを生成することが可能となる。VGAE-NAS では NAS モジュールとして勾配法の PC-DARTS を用いる。PC-DARTS は短時間で高精度な結果が出せるため、複数の構造を活用する提案手法にて採用した。VGAE-NAS で扱う NAS モジュールは PC-DARTS 以外の NAS 手法を扱うことも可能である。可視化手法には、次元削減アルゴリズムである t-SNE [37] や PCA [38] を採用した。

3.2 Encoder

Encoder は式 (2) を用いて μ と $\log \sigma$ を算出する。提案手法の入力は NAS 手法で探索したネットワークアーキテクチャまたは NAS の探索空間からランダムサンプリング等で取り出したアーキテクチャである。

$$\begin{aligned}\mu &= \text{GNN}_{\mu}(\mathbf{F}_N, \mathbf{A}, \mathbf{F}_E) \\ \log \sigma &= \text{GNN}_{\sigma}(\mathbf{F}_N, \mathbf{A}, \mathbf{F}_E)\end{aligned}\quad (2)$$

ここで、 $\mathbf{F}_N$ はノードの特徴量、 $\mathbf{A}$ はエッジの隣接行列、 $\mathbf{F}_E$ はエッジの特徴量が含まれる。ノードの特徴量 $\mathbf{F}_N$ は NAS の入力ノード、中間ノード、出力ノードで構成されており、3 種類のノードの種類が特徴量として入力される。PC-DARTS の 2 種類の Cell を組み合わせる場合、入力ノードは 4 つ、中間ノードは 8 つ (ハイパパラメータ)、出力ノードは 2 つとなる。また、入力ではグラフのエッジ特徴量 $\mathbf{F}_E$ も同時に学習する。エッジ特徴量 $\mathbf{F}_E$ は NAS の探索空間に依存し、PC-DARTS では $|\mathcal{O}| = 8$ の演算候補 (Skip Connection, None, Average Pooling 3×3 ,

Algorithm 1 VGAE-NAS (1 試行)**Input:** Dataset $\mathbf{D}$ **Output:** Latent Feature Space $\mathbf{z}$, New Architecture $\mathbf{A}'_r$

```

1: Prepare dataset  $\mathbf{D}$ 
2: for Epoch = 1 to Specified Epochs in NAS do
3:   Update architecture and evaluate on  $\mathbf{D}$ 
4:   Retain architecture  $\mathbf{A}_r$  with best performance
5: end for
6: Convert  $\mathbf{A}_r$  into graph format  $\mathbf{F}_N, \mathbf{A}, \mathbf{F}_E$ 
7: for Epoch = 1 to Specified Epochs in VGAE do
8:    $\text{Encoder} \leftarrow \mathbf{F}_N, \mathbf{A}, \mathbf{F}_E$ 
9:    $\mu_i, \sigma_i^2 \leftarrow \text{Encoder}$ 
10:   $\mathbf{z} \leftarrow \text{Sample from } \mu_i, \sigma_i^2$ 
11:   $\text{Decoder} \leftarrow \mathbf{z}$ 
12:   $\mathbf{A}', \mathbf{F}'_E \leftarrow \text{Decoder}$ 
13:   $\text{Loss} = \text{Difference}(\mathbf{A}, \mathbf{F}_E, \mathbf{A}', \mathbf{F}'_E)$ 
14:  Update weights  $\mathbf{w}$  of Encoder and Decoder by descending  $\nabla_{\mathbf{w}} \text{Loss}$ 
15: end for
16:  $\mu_i, \sigma_i^2 \leftarrow \text{Trained Encoder with } \mathbf{F}_N, \mathbf{A}, \mathbf{F}_E$ 
17: Latent Feature Space  $\mathbf{z} \leftarrow \text{Sample from } \mu_i, \sigma_i^2$ 
18: Generate  $\mathbf{z}'$  using  $\mathbf{z}$  and the generating model algorithm (Section 3.5)
19: New Architecture  $\mathbf{A}'_r \leftarrow \text{Trained Decoder with } \mathbf{z}'$ 

```

Max Pooling 3×3 , Separable Convolution 3×3 , Separable Convolution 5×5 , Dilated Convolution 3×3 , Dilated Convolution 5×5 ）を使用する。各演算子がエッジ特徴量 $\mathbf{F}_E$ に対応しており、このエッジ特徴量 $\mathbf{F}_E$ に Embedding Layer を通す。訓練中はこの Embedding Layer も対象に学習していく。エッジ特徴量 $\mathbf{F}_E$ をもとに Embedding Layer を使用して埋め込むことによって演算子間での関係性を考慮するようになり、各演算子の情報がより潜在空間に反映されやすくなる。提案手法ではグラフ構造だけでなくエッジの演算を潜在空間で表現するため、元の VGAE にエッジ特徴量を埋め込むなど大きな変更を加えている。 $\mathbf{F}_N, \mathbf{A}$ を入力とする GNN の詳細は GCN のみを用いた層と図 3 のように GAT と GCN を組み合わせた層の 2 種類になっている。2 種類を組み合わせることで序盤の GAT Layer で近傍ノード間の具体的な関係性の情報を学習し、後半に GCN Layer を通すことで近傍ノード間の局所的なグラフ情報を学習するため表現力が向上する仕組みとなっている。

潜在特徴量 $\mathbf{z}$ は算出した μ と $\log \sigma$ で表現された正規分布に従うと仮定し、式 (3) で $\mathbf{z}$ を出力する。

$$q(\mathbf{z}|\mathbf{F}_N, \mathbf{A}, \mathbf{F}_E) = \prod_{i=1}^N q(\mathbf{z}_i|\mathbf{F}_N, \mathbf{A}, \mathbf{F}_E) \quad (3)$$

$$\text{with } q(\mathbf{z}_i|\mathbf{F}_N, \mathbf{A}, \mathbf{F}_E) = \text{Norm}(\mathbf{z}_i|\mu_i, \sigma_i^2)$$

ここで、 N はノード数である。Encoder で正規分布のパラメータに落とし込み、各ノードに対する正規分布の積を取ることによって全体の分布としている。

3.3 Decoder

Decoder では潜在特徴量 $\mathbf{z}$ から式 (4) の Inner Product でグラフ構造を再構成する。ここで、 $\hat{\mathbf{A}}$ は生成されたグラフの隣接行列である。 $\hat{\mathbf{F}}_E$ は生成されたエッジの特徴量であり、エッジの種類を表す。また、潜在特徴量 $\mathbf{z}$ から Edge Classifier を通すことによりエッジの種類を予測する。

Inner Product と Edge Classifier が含まれる Decoder で再構成したグラフと Encoder に入力する元のグラフが同一の形になるよう Loss を減らして最適化することで、対象となるグラフの潜在特徴量を取り出すことが可能となる。

$$p(\hat{\mathbf{A}}, \hat{\mathbf{F}}_E|\mathbf{z}) = \prod_{i=1}^N \prod_{j=1}^N p(\hat{A}_{i,j}, \hat{F}_{Ei,j}|\mathbf{z}_i, \mathbf{z}_j) \quad (4)$$

$$p(\hat{A}_{i,j} = 1|\mathbf{z}_i, \mathbf{z}_j) = \text{Sigmoid}(\mathbf{z}_i^T, \mathbf{z}_j)$$

3.4 Loss Function

VGAE-NAS の Loss Function は、VGAE の Reconstruction Loss (Recon Loss), Edge Class Loss と Kullback-Leibler Divergence Loss (KL Loss) の和で構成する。具体的には以下の式 (5) で表す。 W_{Edge} は Edge Class Loss に対する重要度、 W_{KL} は KL Loss に対する重要度を調整するハイパパラメータ (基本的には 1.0) である。

$$\text{Loss} = L_{Recon} + W_{Edge} * L_{Edge} + W_{KL} * L_{KL} \quad (5)$$

ここで、 L_{Recon} は VGAE の Recon Loss であり、Positive Loss と Negative Loss を用いた以下の式 (6) で表す。 L_{Recon} はグラフを指標にして構造を近づける役割である。 W_P は Positive Loss に対する重要度、 W_N は Negative Loss に対する重要度を調整するハイパパラメータである。

$$L_{Recon} = \sum_{(i,j) \in \mathbf{A}} W_P * A_{i,j} \log(\hat{A}_{i,j}) + W_N * (1 - A_{i,j}) \log(1 - \hat{A}_{i,j}) \quad (6)$$

ここで、 $A_{i,j}$ は真のグラフ、 $\hat{A}_{i,j}$ は生成されたグラフの隣接行列の i 行 j 列要素である。 L_{Edge} は、エッジの種類を予測する Edge Class Loss を表している。Edge Class Loss は、Cross-Entropy Loss を用いて計算する。 L_{Edge} はエッジの種類を指標にして構造を近づける役割である。その他の Loss は元の VGAE でも使用されるが、 L_{Edge} は NAS の細かい特徴を適用するため提案手法にて追加している。 L_{KL} は、潜在変数の正規分布と事前分布の KL Divergence であり、以下の式 (7) で表される。

KL Divergence とは、二つの確率分布がどれだけ異なるかを測る指標である。 L_{KL} は潜在空間を連続的にし、学習を安定させる役割をもつ。

$$L_{KL} = -\frac{1}{2} \sum_{i=1}^D (1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2) \quad (7)$$

式 (7) で、 D は潜在変数の次元数、 μ_i は潜在変数の平均、 σ_i^2

は潜在変数の分散を表す（式（2）で算出）．これらの Loss の合計が提案手法の目的関数である Loss Function となる．

3.5 Generating Model

- アーキテクチャは以下の手順に沿って生成する：
- STEP 1: 良い結果を示すネットワークアーキテクチャの潜在特徴量 $\mathbf{z}$ を特定
- STEP 2: 特定した潜在特徴量 $\mathbf{z}$ を他の潜在特徴量を考慮し各次元で $[0,1]$ の範囲に正規化
- STEP 3: 生成した方向ベクトルをスケーリングし，対応する STEP 2 の正規化した潜在特徴量に加算し，元の良い点の近傍にある新たな潜在特徴量を生成
- STEP 4: STEP 3 で生成した新たな潜在特徴量を STEP 2 の逆変換を使用して逆正規化し，新たな潜在特徴量の学習済みの VGAE の Decoder に入力
- STEP 5: 新たなネットワークアーキテクチャを生成

ここで，STEP 2-4 のスケーリングでは以下の式（8）のように係数をかけて小さくした新たな潜在特徴量 $\mathbf{z}'$ を生成する．Sgn は符号関数であり，入力値の符号を返すため，一様乱数 $\mathcal{U}(-0.5, 0.5)$ から各次元に対して -1 または 1 の値を持つ方向ベクトルが生成される．

$$\mathbf{z}_{\text{normalize}} = \frac{\mathbf{z} - \mathbf{z}_{\min}}{\mathbf{z}_{\max} - \mathbf{z}_{\min}}, \quad \mathbf{d} = \text{Sgn}(\mathcal{U}(-0.5, 0.5))$$
$$\mathbf{z}' = (\mathbf{z}_{\text{normalize}} + \gamma \cdot \mathbf{d})(\mathbf{z}_{\max} - \mathbf{z}_{\min}) + \mathbf{z}_{\min}$$

(8)

STEP 3 では元の潜在特徴量の近傍で新しい潜在特徴量の取得するために微小な摂動を表すため，一様乱数と γ を用いる． γ はハイパパラメータである．正規化した潜在特徴量空間の中で方向ベクトルの距離を一定にすることで， γ の値に応じた絶対距離を取る．生成した新たなネットワークアーキテクチャは元の良い点に対応するネットワークアーキテクチャと近い特性を持つと考える．次節の実験では生成したアーキテクチャを評価し，提案手法を検証する．

4 実 験

本研究では，複数の異なるデータセットを用いた実験によって提案手法 VGAE-NAS の有用性を評価する．探索したアーキテクチャの潜在特徴量を獲得し可視化できるかを評価するために実験 A を，潜在特徴量空間から新たに生成したアーキテクチャが高い性能を持つか評価するために実験 B を行った．実験では 1 試行に対して NVIDIA Tesla V100 SXM2 16GB を 1 つ用いる (ImageNet では 4 つの GPU を用いて並列計算)．

4.1 実験 A-I: CIFAR-10 と MNIST で探索したアーキテクチャの潜在特徴量可視化実験の設定

Experiment A では中身として異なる特性を持つ種類のデー

表 1: 実験設定 (↑ : NAS, ↓ : VGAE)	
Number of Initial Channels	16
Number of Training Epochs	50 Epochs
Loss Function	CrossEntropyLoss
Initial Learning Rate	0.1
Learning Rate Decay	→ 0 (Cosine Schedule)
Learning Rate for Architecture	0.0003
Data Augmentation	Random Crop, Horizontal Flip
Optimizer for Training Parameter	SGD
Optimizer for Architecture Weights	Adam
Number of Training Epochs	500000 Epochs
Initial Learning Rate	0.001
Learning Rate Decay	→ 0.00025 (Cosine Scheduler)
Optimizer	Adam
Dimension of Latent Features	4
Seed	2024
$W_P, W_N, W_{Edge}, W_{KL}$	2.0, 1.0, 0.5, 1.0

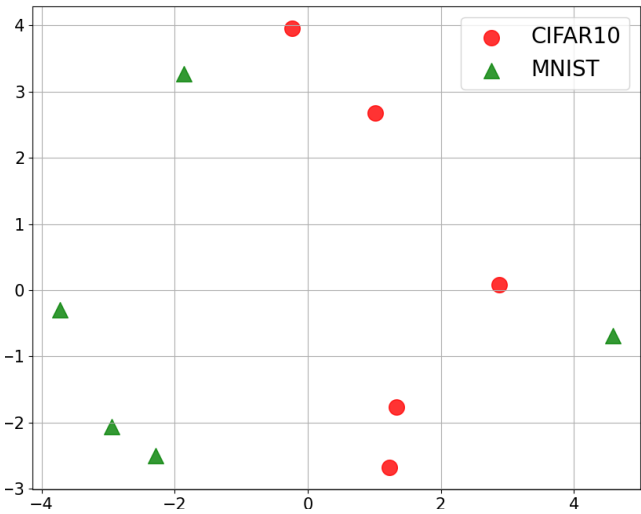


図 4: 提案手法で獲得した潜在特徴量の可視化図，軸：2次元の PCA プロット上の座標軸，色：各データセットの種類

タセットに対して NAS で最適なアーキテクチャを探索し，それぞれのデータセットに適したネットワーク構造の潜在特徴量を可視化する．データセットとしては CIFAR10 [39] と MNIST [40] を用いる．CIFAR10 は深層学習や NAS の実験でよく用いられるデータセットであり，60000 枚の画像がデータセットに含まれている．MNIST は手書き数字の画像をまとめたデータセットである．各データセットや構造の特徴を細かく捉えることを目的とするため，2 種類の Cell を繋げた構造を入力するアーキテクチャとして用いる．NAS や VGAE の設定は表 1 に示す．5 種類の Seed を用いるため，NAS は個別に 5 回探索しており，各 1 回は NAS アルゴリズムを最後まで実行し良い性能を示した構造を取り出している．VGAE の入力として，2 種類のデータセットと 5 回の試行で探索した 10 個のアーキテクチャを使用した．各アーキテクチャにはエッジの接続関係とエッジの種類が含まれており，ノードの数は固定されている．

4.2 実験 A-II: CIFAR10 と MNIST の実験結果と考察

図 4 に提案手法で獲得した潜在特徴量の可視化図を示す．

表 2: USPS データセットを用いた実験 B-II の実験結果

Architecture	Model Size (Mean $\pm$ SD) (M)	Best Test Acc. (Max $\pm$ SD) (%)	Search Time (h)
CENAS [35]	0.57	89.94	7.4
ASED [41]	-	97.75	153.6
NASDA [42]	2.70	98.00	7.2
VGAE-NAS (Random Model)	2.15 $\pm$ 0.28	97.96 $\pm$ 0.10	0
VGAE-NAS (CIFAR10 Model, $\gamma = 0.1$)	3.32 $\pm$ 0.63	98.01 $\pm$ 0.11	0
VGAE-NAS (MNIST Model, $\gamma = 0.0$)	3.37 $\pm$ 0.60	98.01 $\pm$ 0.09	0
VGAE-NAS (MNIST Model, $\gamma = 0.1$)	3.16 $\pm$ 0.66	98.11 $\pm$ 0.13	0

図 4 は PCA によって次元削減したグラフであり、軸は潜在特徴量の多次元データを圧縮した結果の座標軸に相当する。図 4 より、MNIST の右側の 1 点を除き、各データセットに対して潜在特徴量が座標軸として近傍に集まっている傾向が確認できる。CIFAR10 と MNIST では画像サイズやデータセットの特性 (CIFAR10 は RGB 画像、MNIST はグレースケール) が違うため、NAS ではよりデータセットの特徴を捉えたアーキテクチャが探索できていると考える。また、CIFAR10 は画像サイズが 32×32 であるのに対して MNIST は 28×28 であり入力する画像サイズも異なる。従って、データセットに対して最適な近い潜在特徴量がそれぞれ獲得できていることから、CIFAR10 と MNIST で傾向が異なっていることが確認できる。

4.3 実験 B-I: 提案手法で生成したアーキテクチャの性能評価実験の設定

実験 B は提案手法で生成したアーキテクチャとランダムに生成したアーキテクチャを USPS データセット [43] と ImageNet データセット [44] で比較し、提案手法が有用なアーキテクチャを生成できるか、そしてランダムなアーキテクチャと比較して性能向上が見られるか評価することを目的とする。新たに使う USPS (United States Postal Service) は、手書き数字の画像データセットで、 16×16 のグレースケール画像で内容は 0 から 9 までの手書きの数字が含まれる。USPS は MNIST からのドメイン適用としてもよく用いられる類似したデータセット [45] [46] [47] であるため、類似したデータセットに最適化されたアーキテクチャは同様の性能を持つかを実験 B-II により検証する。

本実験では、4 種類のアーキテクチャに対して再学習を行い性能を評価する。第 1 に、獲得した MNIST 周辺の潜在特徴量 ($\gamma = 0.1$) から作ったアーキテクチャ。第 2 に CIFAR10 周辺の潜在特徴量 ($\gamma = 0.1$) から作ったアーキテクチャ。第 3 に獲得した MNIST の潜在特徴量 ($\gamma = 0$)、第 4 にランダムな潜在特徴量から作ったアーキテクチャである。 $\gamma = 0.1$ とした理由としては、 $\gamma = 0.0$ のアーキテクチャの特徴を捉えつつ、一部異なるアーキテクチャであることから周辺の潜在特徴量を構築するパラメータとして設定する。MNIST と CIFAR10 はそれぞれ 5 構造の潜在特徴量が存在し、方向ベクトル生成時に発生する乱数を 4 回生成させアーキテクチャを構築した。4 種類で合計 50 構造 (5 Random Model, 5 MNIST Model ($\gamma =$

0.0), 20 CIFAR10 Model ($\gamma = 0.1$), and 20 MNIST Model ($\gamma = 0.1$)) を評価した。その他の実験設定は表 1 に記載する。

また、ImageNet は画像識別でよく用いられる大規模なデータセットであり、ベースとしている PC-DARTS でも CIFAR10 で探索したアーキテクチャの Cell を転移して ImageNet で評価している。実験 B-III では CIFAR-10 周辺の潜在特徴量から作ったアーキテクチャ (20 構造) を使用して ImageNet で評価する。ImageNet は元の画像サイズが大きいことから 224×224 にクロップし、ランダムに左右反転を行う RandomHorizontalFlip とランダムに明るさやコントラストなどを変化させる ColorJitter を Data Augumentation として採用した。Initial Learning Rate を 0.5、再学習の Epochs を 250 と設定するなど、ImageNet 用のハイパパラメータを用いて 4 つの GPU で並列計算を行った。

4.4 実験 B-II: USPS の実験結果と考察

表 2 に各 NAS 手法の結果を示す。表 2 より、精度が一番高いのは Best Test Acc.(Max) が 98.11 % を示した MNIST model ($\gamma = 0.1$) であることが確認できた。実験結果より、類似したデータセットで最適化された潜在特徴量を参考に構築したアーキテクチャが最も良い結果を示した。MNIST model が Random Model や CIFAR 10 Model より良い性能を示したのはもちろん、MNIST Model の中でも $\gamma = 0.0$ よりも $\gamma = 0.1$ のモデルが高精度であった。これは $\gamma = 0.0$ が MNIST で最適化されたアーキテクチャであり、USPS データセットと類似しつつも異なる内容であるためであると考えられる。新しいデータセットに対しても同様の手順で学習が可能となるため、NAS で 0 から探索しなくて済むため、探索時間を大幅に削減することができる。

4.5 実験 B-III: ImageNet の実験結果

表 3 に各 NAS 手法の結果を示す。表 3 より、精度が一番高いのは Best Test Acc.(Max) が Top1 で 75.43%, Top5 で 92.60% を示した提案手法の CIFAR10 model であることが確認できた。さらに、上記の結果が探索時間をかけずに示した結果であることを考慮すると、非常に効率の良い性能を示せたと言える。提案手法は画像サイズが大きなデータセットや Cell を転移させた大きなアーキテクチャでも有効に働くことが実験結果から明らかになった。

表 3: ImageNet データセットを用いた実験 B-III の実験結果

Architecture	Model Size (Mean $\pm$ SD)	Best Test Acc. (Top1) (Max $\pm$ SD) (%)	Best Test Acc. (Top5) (Max $\pm$ SD) (%)	Search Time (h)
Inception-v1 [48]	6.60	69.80	89.90	0
NASNet-A [17]	5.30	74.00	91.60	43200
AmoebaNet-A [49]	5.10	74.50	92.00	75600
PNAS [50]	5.10	74.20	91.90	$\sim$ 5400
DARTS (CIFAR10 Model) [5]	4.70	73.30	91.30	96
PC-DARTS (CIFAR10 Model) [6]	5.30	74.90	92.20	2.4
iDARTS (CIFAR10 Model) [51]	5.10	74.80	92.10	4.8
VGAE-NAS (CIFAR10 Model, $\gamma = 0.1$)	4.61 $\pm$ 0.75	75.43 $\pm$ 1.12	92.60 $\pm$ 0.64	0

5 結 論

本研究では、NAS と VGAE を組み合わせたフレームワーク VGAE-NAS を用いることで、NAS の探索結果の解釈性の向上を目指した。本研究の課題として、事前に用意した NAS 手法のアーキテクチャや探索空間にのみ対応している点がある。そのため、対応可能なアーキテクチャや探索空間を拡張して汎用性の高い手法に改良していく必要がある。提案手法の応用により、ネットワークアーキテクチャの構造や性能に関する新たな知見を提供できると考える。将来的には、提案手法を用いた複数のアーキテクチャの評価をさらに容易にするために、最適化された構造の精度以外の潜在的な特徴（例えば、モデルサイズや特定の演算子を多く含むアーキテクチャ）を抽出する予定である。また、データセット間の距離を測る手法 [52] [53] も存在するため、提案手法と組み合わせてデータセット間の距離を定量化し、潜在特徴量空間で最適化を行う手法へ改良したいと考える。この先進的なアプローチが、AutoML や Neural Architecture Search の分野を発展させ、ネットワーク設計と最適化の領域により深い洞察をもたらすことを期待する。

文 献

- [1] Barret Zoph and Quoc V Le, “Neural architecture search with reinforcement learning”, *ICLR*, 2016.
- [2] Yan Xu, Zhipeng Jia, Liang-Bo Wang, Yuqing Ai, Fang Zhang, Maode Lai, Eric I Chang, et al., “Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features”, *BMC bioinformatics*, vol. 18, no. 1, pp. 1–17, 2017.
- [3] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu, “Hierarchical representations for efficient architecture search”, *arXiv preprint arXiv:1711.00436*, 2017.
- [4] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V Le, and Alexey Kurakin, “Large-scale evolution of image classifiers”, in *International conference on machine learning*. PMLR, 2017, pp. 2902–2911.
- [5] Hanxiao Liu, Karen Simonyan, and Yiming Yang, “Darts: Differentiable architecture search”, *ICLR*, 2018.
- [6] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guojun Qi, Qi Tian, and Hongkai Xiong, “PC-DARTS: Partial channel connections for memory-efficient architecture search”, in *ICLR*, 2020.
- [7] Catherine Wong, Neil Houlsby, Yifeng Lu, and Andrea Gesmundo, “Transfer learning with neural automl”, in *Neural Information Processing Systems*, 2018.
- [8] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini, “The graph neural network model”, *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.
- [9] Dejun Jiang, Zhenxing Wu, Chang-Yu Hsieh, Guangyong Chen, Ben Liao, Zhe Wang, Chao Shen, Dongsheng Cao, Jian Wu, and Tingjun Hou, “Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models”, *Journal of cheminformatics*, vol. 13, no. 1, pp. 1–23, 2021.
- [10] Xiao Wang, Sean T Flannery, and Daisuke Kihara, “Protein docking model evaluation by graph neural networks”, *Frontiers in Molecular Biosciences*, vol. 8, pp. 647915, 2021.
- [11] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia, “Learning to simulate complex physics with graph networks”, in *International conference on machine learning*. PMLR, 2020, pp. 8459–8468.
- [12] Jonathan Shlomi, Peter Battaglia, and Jean-Roch Vlimant, “Graph neural networks in particle physics”, *Machine Learning: Science and Technology*, vol. 2, no. 2, pp. 021001, 2020.
- [13] Thomas N Kipf and Max Welling, “Semi-supervised classification with graph convolutional networks”, *arXiv preprint arXiv:1609.02907*, 2016.
- [14] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al., “Graph attention networks”, *stat*, vol. 1050, no. 20, pp. 10–48550, 2017.
- [15] Thomas N Kipf and Max Welling, “Variational graph auto-encoders”, *arXiv preprint arXiv:1611.07308*, 2016.
- [16] Diederik P Kingma and Max Welling, “Auto-encoding variational bayes”, *arXiv preprint arXiv:1312.6114*, 2013.
- [17] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le, “Learning transferable architectures for scalable image recognition”, *CVPR*, pp. 8697–8710, 2018.
- [18] Joe Mellor, Jack Turner, Amos Storkey, and Elliot J Crowley, “Neural architecture search without training”, in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 7588–7598, PMLR.
- [19] Arjun Krishnakumar, Colin White, Arber Zela, Renbo Tu, Mahmoud Safari, and Frank Hutter, “Nas-bench-suite-zero: Accelerating research on zero cost proxies”, *Advances in Neural Information Processing Systems*, vol. 35, pp. 28037–28051, 2022.
- [20] Renqian Luo, Fei Tian, Tao Qin, Enhong Chen, and Tie-Yan Liu, “Neural architecture optimization”, *Advances in neural information processing systems*, vol. 31, 2018.

- [21] David Friede, Jovita Lukasik, Heiner Stuckenschmidt, and Margret Keuper, “A variational-sequential graph autoencoder for neural architecture performance prediction”, *arXiv preprint arXiv:1912.05317*, 2019.
- [22] Shen Yan, Yu Zheng, Wei Ao, Xiao Zeng, and Mi Zhang, “Does unsupervised architecture representation learning help neural architecture search?”, in *NeurIPS*, 2020.
- [23] Jian Li, Yong Liu, Jiankun Liu, and Weiping Wang, “Neural architecture optimization with graph vae”, *ArXiv*, vol. abs/2006.10310, 2020.
- [24] Xuefei Ning, Yin Zheng, Tianchen Zhao, Yu Wang, and Huazhong Yang, “A generic graph-based neural architecture encoding scheme for predictor-based nas”, in *European Conference on Computer Vision*. Springer, 2020, pp. 189–204.
- [25] Hayeon Lee, Eunyoung Hyung, and Sung Ju Hwang, “Rapid neural architecture search by learning to generate graphs from datasets”, in *International Conference on Learning Representations*, 2021.
- [26] Michail Chatzianastasis, George Dasoulas, Georgios Siolas, and Michalis Vazirgiannis, “Graph-based neural architecture search with operation embeddings”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2021, pp. 393–402.
- [27] Jovita Lukasik, David Friede, Arber Zela, Frank Hutter, and Margret Keuper, “Smooth variational graph embeddings for efficient neural architecture search”, in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.
- [28] Gabriela Suchopárová and Roman Neruda, “Graph embedding for neural architecture search with input-output information”, 2022.
- [29] Wei Wen, Hanxiao Liu, Hai Helen Li, Yiran Chen, Gabriel Bender, and Pieter-Jan Kindermans, “Neural predictor for neural architecture search”, in *European Conference on Computer Vision*, 2019.
- [30] Yaofu Chen, Yong Guo, Qi Chen, Minli Li, Wei Zeng, Yaowei Wang, and Mingkui Tan, “Contrastive neural architecture search with neural architecture comparators”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9502–9511.
- [31] Colin White, Willie Neiswanger, and Yash Savani, “Bananas: Bayesian optimization with neural architectures for neural architecture search”, in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 10293–10301.
- [32] Jovita Lukasik, Steffen Jung, and Margret Keuper, “Learning where to look—generative nas is surprisingly efficient”, in *European Conference on Computer Vision*. Springer, 2022, pp. 257–273.
- [33] Andrea Agiollo and Andrea Omicini, “Gnn2gnn: Graph neural networks to generate neural networks”, in *Uncertainty in Artificial Intelligence*. PMLR, 2022, pp. 32–42.
- [34] Martin Wistuba, “Xfernas: Transfer neural architecture search”, in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*. Springer, 2021, pp. 247–262.
- [35] Mohan Singamsetti, Anmol Mahajan, and Matthew Guzdial, “Conceptual expansion neural architecture search (ce-nas)”, *arXiv preprint arXiv:2110.03144*, 2021.
- [36] Zhichao Lu, Gautam Sreekumar, Erik Goodman, Wolfgang Banzhaf, Kalyanmoy Deb, and Vishnu Naresh Boddeti, “Neural architecture transfer”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 2971–2989, sep 2021.
- [37] Laurens Van der Maaten and Geoffrey Hinton, “Visualizing data using t-sne”, *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [38] Karl Pearson, “Liii. on lines and planes of closest fit to systems of points in space”, *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.
- [39] Alex Krizhevsky, Geoffrey Hinton, et al., “Learning multiple layers of features from tiny images”, 2009.
- [40] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [41] Anton Muravev, Jenni Raitoharju, and Moncef Gabbouj, “Neural architecture search by estimation of network structure distributions”, *IEEE Access*, vol. 9, pp. 15304–15319, 2021.
- [42] Yichen Li and Xingchao Peng, “Network architecture search for domain adaptation”, 2021.
- [43] Jonathan J. Hull, “A database for handwritten text recognition research”, *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 5, pp. 550–554, 1994.
- [44] Olga Russakovsky, Jia Deng, et al., “Imagenet large scale visual recognition challenge”, *IJCV*, vol. 115, pp. 211–252, 2015.
- [45] Stefan Schrod, Jonas Lippl, Andreas Schäfer, and Michael Altenbuchinger, “Fact: Federated adversarial cross training”, 2023.
- [46] Jing Wang, Jiahong Chen, Jianzhe Lin, Leonid Sigal, and Clarence W de Silva, “Discriminative feature alignment: Improving transferability of unsupervised domain adaptation by gaussian-guided latent alignment”, *Pattern Recognition*, vol. 116, pp. 107943, 2021.
- [47] Geoffrey French, Michal Mackiewicz, and Mark Fisher, “Self-ensembling for visual domain adaptation”, *arXiv preprint arXiv:1706.05208*, 2017.
- [48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, “Going deeper with convolutions”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [49] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le, “Regularized evolution for image classifier architecture search”, in *Proceedings of the aaai conference on artificial intelligence*, 2019, vol. 33, pp. 4780–4789.
- [50] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy, “Progressive neural architecture search”, in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 19–34.
- [51] Huiqun Wang, Ruijie Yang, Di Huang, and Yunhong Wang, “idarts: Improving darts by node normalization and decorrelation discretization”, *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [52] David Alvarez-Melis and Nicolo Fusi, “Geometric dataset distances via optimal transport”, in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 21428–21439, Curran Associates, Inc.
- [53] Bjørn Magnus Mathisen, Agnar Aamodt, Kerstin Bach, and Helge Langseth, “Learning similarity measures from data”, *Progress in Artificial Intelligence*, vol. 9, 10 2019.