HOTNAS: Hierarchical Optimal Transport for Neural Architecture Search

Jiechao Yang[1,2] Yong Liu[1,2,*] Hongteng Xu[1,2]
[1] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[2] Beijing Key Laboratory of Big Data Management and Analysis Methods
{yangjiechao2021, liuyonggsai, hongtengxu}@ruc.edu.cn

Abstract

Instead of searching the entire network directly, current NAS approaches increasingly search for multiple relatively small cells to reduce search costs. A major challenge is to jointly measure the similarity of cell micro-architectures and the difference in macro-architectures between different cell-based networks. Recently, optimal transport (OT) has been successfully applied to NAS as it can capture the operational and structural similarity across various networks. However, existing OT-based NAS methods either ignore the cell similarity or focus solely on searching for a single cell architecture. To address these issues, we propose a hierarchical optimal transport metric called HOTNN for measuring the similarity of different networks. In HOTNN, the cell-level similarity computes the OT distance between cells in various networks by considering the similarity of each node and the differences in the information flow costs between node pairs within each cell in terms of operational and structural information. The network-level similarity calculates OT distance between networks by considering both the cell-level similarity and the variation in the global position of each cell within their respective networks. We then explore HOTNN in a Bayesian optimization framework called HOTNAS, and demonstrate its efficacy in diverse tasks. Extensive experiments demonstrate that HOTNAS can discover network architectures with better performance in multiple modular cell-based search spaces.

1. Introduction

Neural Architecture Search (NAS) [22, 59, 66, 68] has received widespread attention as it can automatically discover the well-performing network architectures for a given task. Current NAS methods [12, 13, 36, 66, 70] tend to search multiple relatively small cells instead of the entire network directly, which can significantly reduce the search cost and enhance flexibility. However, most previous cell-based NAS methods [36, 60, 70] focus only on the cell-level architecture search and ignore the network-level search. They usually repeatedly stack one or more identical cells to construct the entire network, which restricts network's diversity and efficiency [55]. To overcome these shortcomings, certain studies [10, 62] have proposed a modular cell-based search space, in which each cell of the whole network can have a different architecture. They have demonstrated that the generated network using the modular cell-based search space achieves superior performance while maintaining a good balance of flexibility and efficiency [55]. In this study, we mainly focus on searching networks in this general modular cell-based search space, though our proposed method can also be extended to more complex network search space.

Early NAS methods like random search [33], reinforcement learning [3, 69], and evolutionary search [37] require collecting a large number of neural networks, which is costly since training a deep neural network can take several hours or even days [1, 39]. One-shot methods [6, 36, 42] reduce the search cost by training a supernet and sampling subnetworks from it, where the weights of all subnetworks are inherited from the trained supernet. Nonetheless, training the complex supernet is a challenging task, and the predictive performance of subnetworks is not always indicative of their fully-trained performance [9]. Bayesian optimization methods [7, 18, 23, 44, 49, 59] offer a competitive alternative to solve the NAS problem in a more efficient manner. It relies on constructing a probabilistic surrogate model [4] that learns the complex relationship between the network architecture and its predictive performance. An acquisition function [20, 24, 30, 53] determines the next promising architecture based on the prediction of the surrogate model, which fully balance the exploration and exploitation of the whole search space. Through iteratively updating the surrogate model, Bayesian optimization methods can efficiently discover high-performing network architectures with a limited number of samples. In the NAS context, Bayesian optimization methods assume similar networks should exhibit similar performance [25]. Hence, a major challenge

*Corresponding Author

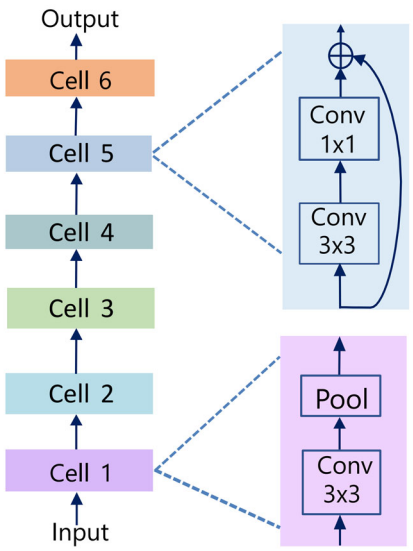


Figure 1. Example of a modular cell-based network WITH stacked cells, each having a different architecture.

of Bayesian optimization for NAS is accurately quantifying the similarity between different networks.

Measuring the similarity between different networks on the modular cell-based search space is not a trivial task [14] because the performance of a neural network may be influenced by the number of cells, the layout of each cell, the cell architectures, etc [14]. Therefore, it is necessary to consider the similarity of cells' internal architectures and the whole macro-architectures [15, 68]. Conventional Bayesian optimization methods rely on designing specific encoding schemes like vector encoding [34] and path encoding [59], which are inadequate for representing complex graph-like network architectures. Recently, optimal transport (OT) [5, 27, 41] has emerged as a promising technique fo NAS as it achieves impressive results across various tasks. As a neural network architecture can be viewed as a directed acyclic attributed graph, OT can naturally handle this graph-like architecture. Kandasamy *et al*. [25] explores the use of OT for measuring the similarity between different networks. However, the NASBOT method focuses solely on the entire network, neglecting the potential relationships and similarities between cells. To solve this problem, Nguyen *et al*. [40] have proposed the Tree-Wasserstein (TW) metric for quantifying the similarity between different cells. Unfortunately, they ignore similarities in macro-architectures, such as the number and layout of stacked cells, and the output channels associated with each cell.

Current optimal transport for NAS methods either ignore the local cell structure or limit to search for a single cell network. To adapt well to the modular cell-based search space, we propose a hierarchical optimal transportation metric called HOTNN, which leverages the hierarchical structure of cell-based networks to measure the similarity of cell internal architectures and that of macro-architectures of networks jointly. HOTNN involves a two-level transportation problem. At the cell level, it computes the similarity between cells by jointly exploiting the similarities between various nodes in different cells and the differences in information flow costs between different node pairs within each cell, in terms of both operational and structural information. At the network-level, it calculates the similarity between networks by considering both the similarity between cells in different networks and the difference in the position of each cell within their respective networks. To demonstrate the effectiveness of our proposed HOTNN metric, we integrate it into the Gaussian process surrogate model within the Bayesian optimization framework and named our proposed method as HOTNAS (Hierarchical Optimal Transport for Neural Architecture Search). We compare HOTNAS against other popular Bayesian optimization methods for NAS. The results show that our proposed HOTNAS method can find neural networks with superior performance across various tasks in multiple modular cell-based search spaces.

2. Related Work

Optimal Transport for NAS. Optimal transport has garnered significant attention due to its excellent ability to mine differences in features and geometric characteristics among structured objects [19, 29, 41, 57]. Recently, Kandasamy *et al*. [25] introduced a metric called OTMANN, , which leverages optimal transport to measure network similarity. It views each network as a probability distribution and computes the minimum transport distance between them as a similarity metric between networks. Unfortunately, they ignore the similarity between cells in different cell-based networks. Furthermore, the optimal transport metric is generally indefinite. To solve this problem, Nguyen *et al*. [40] proposed a negative definite tree-Wasserstein (TW) distance for network architectures by utilizing the frequency of layer operations and global structural information. Unfortunately, they ignore the structural location information of each layer and the local structural difference between different node pairs within each network. In addition, they are limited to searching for a single cell architecture and ignore the similarity of macro-architectures. Compared to the existing Bayesian optimization for NAS methods, our proposed HOTNN metric can jointly learn the similarity of cells' internal micro-architectures and that of the entire network's macro-architectures, thus allowing for finding the best-performing architectures in the general modular cell-based space. More related works on OT and Bayesian optimization for NAS are provided in the supplementary material Sec. S1.

3. Hierarchical Optimal Transport Approach for NAS

Problem Setting. The objective of NAS is to discover an optimal neural network architecture $\boldsymbol{a}^* \in \mathcal{A}$ that achieves the minimum validation loss on the validation set:

$$\boldsymbol{a}^* = \underset{\boldsymbol{a}\in\mathcal{A}}{\operatorname{argmin}} f(\boldsymbol{a}), \tag{1}$$

where $f(\boldsymbol{a})$ is the validation loss of a neural network $\boldsymbol{a}$, $\mathcal{A}$ is the search space. Here we choose the modular cell-based search space as it has been widely employed in designing neural network architectures [36, 43, 50, 55, 70]. Next, we will provide a formulation of neural networks in the modular cell-based search space.

Neural Network Formulation. As shown in Fig. 1, the entire neural network architecture $\boldsymbol{a}$ can be represented as a sequence of n cells, *i.e.*, $\boldsymbol{a} = B_1 \circ B_2 \ldots \circ B_n$. Each cell B_i can have a different structure. The internal structure of a cell B_i can be represented as an acyclic directed attributed graph $\mathcal{G}_{B_i} = (\mathcal{L}, \mathcal{E}, \ell_o, \ell_s)$, where $\mathcal{L}$ denotes the set of nodes in the cell. Each node in the cell corresponds to a particular layer of the cell network. The function $\ell_o : \mathcal{L} \to \Omega_o$ assigns a specific operation $o_u = \ell_o(u)$ to each node $u \in \mathcal{L}$ in the operational metric space (Ω_o, D). Note that the metric $D : \Omega_o \times \Omega_o \to \mathbb{R}_+$ is symmetric and measures the similarity of operations at different nodes. An edge $(u, v) \in \mathcal{E}$ is a topologically ordered pair of layers, representing an information flow from layer $u \in \mathcal{L}$ to the next layer $v \in \mathcal{L}$. The $\ell_s : \mathcal{L} \to \Omega_s$ is an implicit function, mapping each node $u \in \mathcal{L}$ to its structural representation $s_u = \ell_s(u)$ in the structural space (Ω_s, G). $G : \Omega_s \times \Omega_s \to \mathbb{R}_+$ encodes the structural differences between nodes in the cell network.

It is not a trivial work to quantify the similarity between different networks in this modular cell-based search space. There exist several challenges in defining this similarity metric: (i) it should jointly learn the similarity of cell-level micro-architectures and that of network-level macro-architectures in different networks. (ii) the similarity of cell micro-architectures should be able to leverage the operational and structural information of each cell network. (iii) the similarity of macro-architectures needs to take into account the similarity between cells in different networks and the difference in the position of each cell in their respective networks. Hence, there is an urgent need to design a network similarity metric to address the above challenges together. To this end, we propose a novel metric called HOTNN in Sec. 3.1 that utilizes hierarchical optimal transport to measure the similarity between different networks. We further integrate HOTNN into the Bayesian optimization framework and introduce our proposed HOT-NAS method in Sec. 3.2.

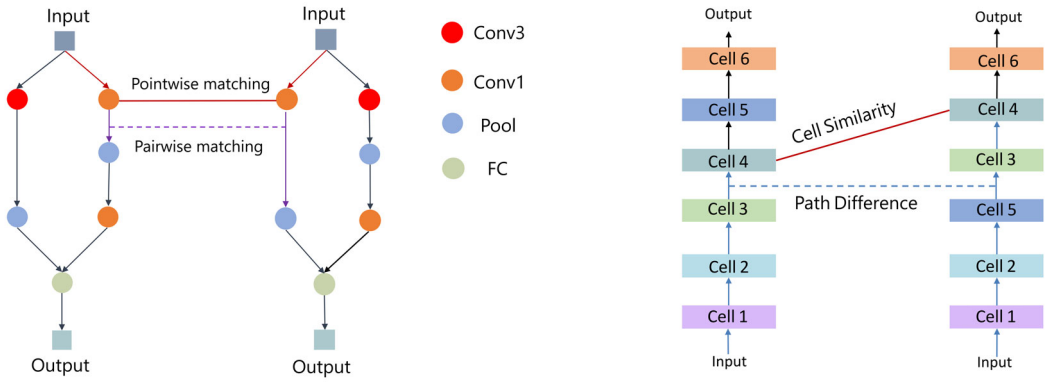
3.1. Hierarchical Optimal Transport for Neural Networks

3.1.1 Cell-level Similarity

Defining the similarity between cells requires simultaneously considering the similarity of operational and structural information. Recent works like NASBOT [25] and BO-TW [40] explore the usage of OT for solving this problem. Unfortunately, they focus only on the similarity of operational and structural information between different nodes within cells, while ignoring differences in the information flow costs between various node pairs within each cell, in terms of the operational and structural information. For example, as shown in Fig. 2a, the two cells B_1, B_2 share the same topological structures and operations, while they have different mappings of operations at each node. The distance between the two different cells is 0, according to the NASBOT or BO-TW methods. However, it is obvious to see that the information flows of the two cell networks are distinctly different. Therefore, the two cell networks are functionally different, and their distance is not 0.

To overcome these problems, we need to define a new OT distance to better characterize the similarity between different cell networks. Inspired by the ability of Fused Gromov-Wasserstein (FGW) metric [56] to compare graphs using the similarity of operational information at each node and local structural differences between node pairs within each graph, we propose an improved Fused Gromov-Wasserstein (iFGW) metric for measuring the similarity between different cells. Note that the original FGW metric is specifically designed for the undirected acyclic graph, while the cell network can be viewed as a directed acyclic graph. Unlike the original FGW metric that selects arbitrary undirected node pairs within each graph, our proposed iFGW metric selects only the topologically ordered node pairs on the paths from the input to output, *i.e.*, the selected node pair (u, v) is ordered where the node $u \in \mathcal{L}$ comes before the node $v \in \mathcal{L}$ on the paths from the u to v. Moreover, the iFGW metric can jointly learn the similarity between nodes in different cells and the differences in the information flow costs between node pairs within each cell, in terms of the operational and structural information. Next, we will give a detailed explanation of the iFGW metric.

In the context of OT, each cell network can be represented as a probability measure over the joint metric space of the operational and structural information, *i.e.*, $\tau = \sum_{i=1}^{n} w_i \delta_{(o_i, s_i)}$, where $\delta_{(o_i,s_i)}$ is the Dirac delta function at the node i with the operation o_i and location s_i, w_i is the layer mass that quantifies the importance of each node in the cell network, n is the number of nodes in the cell network. In this study, we assume the operation of each node contributes equally to the overall cell network, *i.e.*, $w_i = 1/n$, though our work can be ex-



(a) cell-level similarity

(b) network-level similary

Figure 2. An example of the cell-level similarity and network-level similarity in the HOTNN metric. (a) Each node has a different color, which represents a specific operation. Similar operations have a closed colour. The red solid line represents the pointwise matching between nodes in two networks, and the purple dotted line represents the pairwise matching between node pairs in their respective cell networks. (b) Each cell has a different color, which represents a specific architecture. The red solid line represents the cell-level similarity in two networks, the blue dotted line represents the location difference of each cell in their respective cell networks.

tended to a more general case in which each operation contributes differently. The discrete probability distribution $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ belongs to a probability simplex Δ_n, *i.e.*, $\Delta_n = \{w_i \in \mathbb{R}_+^n \mid \sum_{i=1}^n w_i = 1\}$. As such, the two cell networks $\mathcal{G}_{B_1}$, $\mathcal{G}_{B_2}$ with n and m nodes can be described separately by their discrete probability measure $\alpha = \sum_{i=1}^n p_i \delta_{(o_i^p, s_i^p)}$ and $\beta = \sum_{j=1}^m q_j \delta_{(o_j^q, s_j^q)}$, where $\mathbf{p} = (p_1, p_2, \ldots, p_n) \in \Delta_n$ and $\mathbf{q} = (q_1, q_2, \ldots, q_m) \in \Delta_m$ are probability distributions of each cell network. The OT problem aims to find the minimal transportation cost iFGW$(\mathcal{G}_{B_1}, \mathcal{G}_{B_2})$ with the optimal matching matrix $\mathbf{T} \in \mathbb{R}_+^{n \times m}$ between $\mathbf{p}$ and $\mathbf{q}$ of two cell networks. The transport matching matrix $\mathbf{T}$ must satisfy the mass conservation constraint with $U(\mathbf{p}, \mathbf{q})$:

$$\mathbf{T} \in U(\mathbf{p}, \mathbf{q}) = \{\mathbf{T} \in \mathbb{R}_+^{n \times m} \mid \mathbf{T}\mathbf{1}_m = \mathbf{p},\ \mathbf{T}^T \mathbf{1}_n = \mathbf{q}\}, \tag{2}$$

where each element $\mathbf{T}_{ij}$ describes the amount of masses transported from node i in the cell network $\mathcal{G}_{B_1}$ to another node j in the cell network $\mathcal{G}_{B_2}$.

Pointwise Matching. To measure the similarity of nodes in different cell networks, we design a pointwise matching scheme that maps each node i in $\mathcal{G}_{B_1}$ to a corresponding node j in $\mathcal{G}_{B_2}$. Since each node in the cell network has a unique operation type and structural location, we measure the similarity between nodes by simultaneously considering their differences in operational type and structural location information.

- **Operation similarity.** Let $\boldsymbol{D}(o_i^p, o_j^q)$ be the operational similarity cost matrix between node i with operation o_i^p and another node j with operation o_j^q. To model the similarity between different operations, we first build a predefined operation tree by hierarchically grouping similar operations like the tree-Wasserstein method proposed by Nguyen *et al.* [40]. Then, we transform this operation tree into an operational similarity cost matrix $\boldsymbol{D}^{pq} \in \mathbb{R}^{n \times m}$, where each element $\boldsymbol{D}(o_i^p, o_j^q)$ is computed by summing over all edge weights from the operation o_i^p to the operation o_j^q in the predefined operation tree. An example of calculating the similarity of different operations is shown in the supplementary material Sec. S2.

- **Location difference.** In this work, we use the shortest/longest/random-walk path lengths $(G_{in}^{sp}(i)/G_{in}^{lp}(i)/G_{in}^{rp}(i))$ from the input u_{in} to node i and the shortest/longest/random-walk path lengths $(G_{out}^{sp}(i)/G_{out}^{lp}(i)/G_{out}^{rp}(i))$ from node i to the output u_{out} to represent the structural location of node i in the whole cell network ($\boldsymbol{s}_i \overset{\text{def.}}{=} (G_{in}^{sp}(i), G_{in}^{lp}(i), G_{in}^{rp}(i), G_{out}^{sp}(i), G_{out}^{lp}(i), G_{out}^{rp}(i))$). Kandasamy *et al.* [25] have demonstrated that these path lengths are finite and can capture various types of structural location information of each node within each cell. The detailed definition and computation of the shortest, longest, and random-walk path lengths can be referred to Kandasamy *et al.* [25]. Let $H(\boldsymbol{s}_i^p, \boldsymbol{s}_j^q)$ be the structural location difference between the node i with location $\boldsymbol{s}_i^p$ and another node j with location $\boldsymbol{s}_j^q$, which is defined as:

$$H(\boldsymbol{s}_i^p, \boldsymbol{s}_j^q) = \frac{1}{6}(\boldsymbol{s}_i^p - \boldsymbol{s}_j^q)\mathbf{1}_6, \tag{3}$$

where $\mathbf{1}_6$ denotes a 6-dimensional all-one vector.

Let $\boldsymbol{C}^{pq} \in \mathbb{R}^{n\times m}$ be the cross node-node transport cost matrix, where each element $\boldsymbol{C}_{ij}^{pq}$ is a combination of the operational similarity $D(o_i^p, o_j^q)$ and location difference $H(\boldsymbol{s}_i^p, \boldsymbol{s}_j^q)$:

$$\boldsymbol{C}_{ij}^{pq} = \varepsilon D(o_i^p, o_j^q) + (1-\varepsilon)H(\boldsymbol{s}_i^p, \boldsymbol{s}_j^q), \\ 1 \le i \le n, 1 \le j \le m, \tag{4}$$

where $\varepsilon \in [0,1]$ is the hyperparameter that controls the trade-off between the operational similarity and location difference between different nodes in two networks. If the two nodes i and j have similar operations and share similar structural locations in their respective cell networks, the cross node-node transport cost is low, and they are likely to be matched together. Finally, the pointwise matching problem becomes seeking the minimum transportation cost between nodes in two cell networks $\mathcal{G}_{B_1}$ and $\mathcal{G}_{B_2}$:

$$\min_{\mathbf{T}\in U(\mathbf{p},\mathbf{q})} \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{T}_{ij}\boldsymbol{C}_{ij}^{pq}. \tag{5}$$

Pairwise Matching. We design a pairwise matching scheme to learn the differences in the movement cost of various information flows between pairs of nodes within each cell network. We assume there exists a node pair (i,k) and have different paths from node i to node k in the cell network $\mathcal{G}_{B_1}$. Similar to pointwise matching, we use the structural location difference $H(\boldsymbol{s}_i^p, \boldsymbol{s}_k^p)$ between node i and node k to represent the structural information of the node pair (i,k) in the network $\mathcal{G}_{B_1}$. The structural location difference $H(\boldsymbol{s}_i^p, \boldsymbol{s}_k^p)$ reflects the movement cost of various information flows from the node i with location $\boldsymbol{s}_i^p$ to the node k with location $\boldsymbol{s}_k^p$ in terms of structural information. In addition, we consider the operational difference between the node pair (i,k) and use $D(o_i^p, o_k^p)$ to represent the movement cost of various information flows from node i with operation o_i^p to node k with operation o_k^p in terms of the operational information. Let $\boldsymbol{C}_{ik}^p$ be the total movement costs of various information flows from the node i to the node k by considering both the operational and structural difference in the node pair (i,k) of the cell network $\mathcal{G}_{B_1}$:

$$\boldsymbol{C}_{ik}^p = \varepsilon D(o_i^p, o_k^p) + (1-\varepsilon)H(\boldsymbol{s}_i^p, \boldsymbol{s}_k^p), 1 \le i < k \le n, \tag{6}$$

where $\varepsilon \in [0,1]$ is the hyperparameter that controls the trade-off between the operational similarity and structural difference. Note that $\boldsymbol{C}_{ik}^p$ is ∞ if there exits no information flows from node i to node k. Similarly, let $\boldsymbol{C}_{jl}^q$ be the total movement cost of various information flows from node j to node l by considering both the operational and structural difference in the node pair (j,l) in the cell network $\mathcal{G}_{B_2}$:

$$\boldsymbol{C}_{jl}^q = \varepsilon D(o_j^q, o_l^q) + (1-\varepsilon)H(\boldsymbol{s}_j^q, \boldsymbol{s}_l^q), 1 \le j < l \le m. \tag{7}$$

Then we define the transport cost between the node pair (i,k) in the network $\mathcal{G}_{B_1}$ and another node pair (j,l) in the network $\mathcal{G}_{B_2}$ is the total movement cost difference of each node pair in their respective cell networks, *i.e.*, $|\boldsymbol{C}_{ik}^p - \boldsymbol{C}_{jl}^q|$. Finally, the pairwise-wise matching problem becomes seeking the minimum transport cost between different node pairs within each cell network:

$$\min_{\mathbf{T}\in U(\mathbf{p},\mathbf{q})} \sum_{i=1}^{n}\sum_{k=i+1}^{n}\sum_{j=1}^{m}\sum_{l=j+1}^{m} \mathbf{T}_{ij}\mathbf{T}_{kl}\left|\boldsymbol{C}_{ik}^p - \boldsymbol{C}_{jl}^q\right|. \tag{8}$$

If the node pairs (i,k) and (j,l) have similar total movement costs, they are likely to be matched together.

iFGW Metric. We propose the iFGW metric to simultaneously learn the similarity of operational and connection information in different cell networks. It combines pointwise and pairwise matching schemes (see Fig. 2a). We define the iFGW metric as the minimum transport cost by jointly exploiting the similarity between different nodes in two cell networks and the total movement cost difference between different pairs of nodes within each network:

$$\text{iFGW}(\mathcal{G}_{B_1}, \mathcal{G}_{B_2}) = \min_{\mathbf{T}\in U(\mathbf{p},\mathbf{q})} \sum_{i=1}^{n}\sum_{k=i+1}^{n}\sum_{j=1}^{m}\sum_{l=j+1}^{m} \lambda\mathbf{T}_{ij}\boldsymbol{C}_{ij}^{pq} \\ +(1-\lambda)\mathbf{T}_{ij}\mathbf{T}_{kl}\left|\boldsymbol{C}_{ik}^p - \boldsymbol{C}_{jl}^q\right|, \tag{9}$$

where $\lambda \in (0,1)$ is the hyperparameter that controls the tradeoff between the pointwise matching cost of the nodes i and j in each cell network and the pairwise matching cost of the node pair (i,k) and the node pair (j,l).

Theorem 3.1. *Given two cell internal networks $\mathcal{G}_{B_1}$ and $\mathcal{G}_{B_2}$, iFGW$(\mathcal{G}_{B_1}, \mathcal{G}_{B_2})$ is a pseudo-metric. The iFGW metric is 0 if the two cell networks have the same number of nodes with the same type of operations and are connected by the same edges.*

The proof of this theorem can be found in the supplementary material Sec. S3. The iFGW metric is generally lower when two cell networks are more similar. This theorem provides a reasonable interpretation of why the two cell networks are similar in terms of operational information and network's structure. It means the iFGW metric tends to map node pairs with similar pairs of operations and obtains small differences in structural location on their respective cell networks.

3.1.2 Network-level Similarity

As is shown in Fig. 2b, suppose there exit two networks $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$ with N and M cells, respectively. Each network consists of a sequence of different cell networks *i.e.*, $\boldsymbol{a}^1 = B_1^1 \circ B_2^1 \ldots \circ B_N^1$ and $\boldsymbol{a}^2 = B_1^2 \circ B_2^2 \ldots \circ B_M^2$. Similar to the iFGW metric, each network can be represented by a probability measure separately *i.e.*, $\mu = \sum_{s=1}^{N} f_s \delta_{B_s^1}$ and $\nu = \sum_{t=1}^{M} g_t \delta_{B_t^2}$, where $\delta_{B_s^1}$ and $\delta_{B_t^2}$ is the Dirac delta function at the cell B_s^1 of the network $\boldsymbol{a}^1$ and at the cell B_t^2 of the network $\boldsymbol{a}^2$, respectively. $\mathbf{f} = (f_1, f_2, \ldots, f_N) \in \Delta_N$ and $\mathbf{g} = (g_1, g_2, \ldots, g_M) \in \Delta_M$ are probability distributions of each network, where each element is the mass of cells in their respective networks. Here we use the relative size of output channels in each cell as the cell mass *i.e.*, $f_s = c_s / \sum_{s=1}^{N} c_s, g_t = c_t / \sum_{t=1}^{M} c_t$, where c_s and c_t are the output channels of each cell in the networks $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$, respectively. The similarity between two networks $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$ is the minimal transportation cost that has the optimal matching $\Gamma \in \mathbb{R}_+^{\mathrm{N} \times \mathrm{M}}$ between $\mathbf{f}$ and $\mathbf{g}$. The transport matching Γ must satisfy the mass conservation constraint with $V(\mathbf{f}, \mathbf{g})$:

$$\Gamma \in \mathrm{V}(\mathbf{f}, \mathbf{g}) = \{\Gamma \in \mathbb{R}_+^{\mathrm{N} \times \mathrm{M}} \mid \Gamma \mathbf{1}_{\mathrm{M}} = \mathbf{f}, \Gamma^{\mathrm{T}} \mathbf{1}_{\mathrm{N}} = \mathbf{g}\}, \tag{10}$$

where each element of the transport matching matrix Γ_{st} describes the amount of mass transported from the cell B_s^1 in the network $\boldsymbol{a}^1$ to another cell B_t^2 in the network $\boldsymbol{a}^2$.

Let $\boldsymbol{S} \in \mathbb{R}^{N \times M}$ be the cost matrix between cells in networks $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$. The same cell appearing in different positions within a network may have varying impacts on the final network performance. We quantify the transport cost between cells in two networks by considering both the similarity between cells in two networks and the difference in the global position of each cell in their respective networks. Let $I(B_s^1, B_t^2)$ be the similarity between cell B_s^1 in the network $\boldsymbol{a}^1$ and cell B_t^2 in the network $\boldsymbol{a}^2$, which is also the iFGW$(\mathcal{G}_{B_s^1}, \mathcal{G}_{B_t^2})$. Let $P(B_s^1, B_t^2)$ be the global position difference between cell B_s^1 in the network $\boldsymbol{a}^1$ and cell B_t^2 in the network $\boldsymbol{a}^2$, which is defined as follows:

$$\begin{aligned} P(B_s^1, B_t^2) = &\mid \delta^1(B_s^1)/\delta^1 - \delta^2(B_t^2)/\delta^2 \mid, \\ &1 \le s \le N, 1 \le t \le M, \end{aligned} \tag{11}$$

where $\delta^1(B_s^1)$ and $\delta^2(B_t^2)$ denote the longest path length from the input to the cell B_s^1 in the network $\boldsymbol{a}^1$ and the cell B_t^2 in the network $\boldsymbol{a}^2$, respectively. δ^1 and δ^2 denote the longest path length from the input to the output in the network $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$, respectively. Then, the transport cost between the cell B_s^1 in the network $\boldsymbol{a}^1$ and the cell B_t^2 in the network $\boldsymbol{a}^2$ is:

$$\boldsymbol{S}_{st}^{12} = (1 - \eta) I(B_s^1, B_t^2) + \eta P(B_s^1, B_t^2), \tag{12}$$

where $\eta \in [0, 1]$ is the hyperparameter that controls the tradeoff between the similarity between different cells in two networks and the difference in the global position of each cell in their respective networks.

HOTNN Metric. To measure the similarity between different networks, we propose the HOTNN metric that combines both the similarity between different cells in two networks and the difference in the global position of each cell in their respective networks, which is defined as follows:

$$\mathrm{HOTNN}(\boldsymbol{a}^1, \boldsymbol{a}^2) = \min_{\Gamma \in \mathrm{V}(\mathbf{f}, \mathbf{g})} \sum_{s=1}^{N} \sum_{t=1}^{M} \Gamma_{\mathrm{st}} \boldsymbol{S}_{st}^{12}. \tag{13}$$

The HOTNN metric looks for the optimal map Γ with the minimum transport cost between two networks.

Theorem 3.2. *Given two networks $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$, HOTNN$(\boldsymbol{a}^1, \boldsymbol{a}^2)$ is a pseudo-metric and negative semi-definite. The HOTNN metric is 0 if the two networks have the same number of cells with the same architectures.*

The proof of this theorem can be found in the supplementary material Sec. S3. The HOTNN metric is generally lower when the two networks are more similar. This theorem provides a reasonable explanation as to why the two networks are similar in terms of cell micro-architectures and network macro-architectures. The HOTNN metric tends to match the cells that have similar architectures and similar global positions in their respective networks.

3.2. Neural Architecture Search via Bayesian Optimization with Hierarchical Optimal Transport

In this section, we explore the usage of HOTNN distance in the Bayesian optimization for the NAS framework to demonstrate its effectiveness across various modular cell-based search spaces. The Bayesian optimization method uses surrogate models to learn the complex relationship between the network architecture and its validation performance. The commonly-used surrogate models in Bayesian optimization include the Gaussian process (GP) [61], Bayesian neural networks [51, 52], and random forest [21]. Here we choose the GP as our surrogate model because it provides a closed-form of the smooth predictive distribution and accurately estimates the predictive uncertainty [4]. Moreover, it can optimize model hyperparameters automatically by maximizing the log marginal likelihood [61]. The GP employs a kernel function to model the similarity between different input pairs. The kernel function value is higher if the two inputs are more similar. Therefore, it is natural to incorporate our proposed HOTNN distance

Table 1. Comparisons of the best-found valid loss and test loss on the TransNAS-Bench-101 benchmark and the DARTS benchmark.

Search Space	Tasks	Loss	Random Search	Evolutionary Search	BO-edit	NASBOT	HOTNAS
TransNAS-Bench-101	Autoencoding	Valid Error	28.09±0.18	28.19±0.24	28.55±0.28	29.25±0.25	**25.80±0.04**
		Test Error	26.65±0.18	26.74±0.24	27.14±0.28	27.82±0.25	**24.36±0.01**
	Object Classification	Valid Error	53.21±0.02	52.96±0.02	53.42±0.03	53.28±0.02	**52.69±0.00**
		Test Error	46.49±0.02	46.25±0.04	46.67±0.03	46.30±0.03	**45.90±0.02**
	Scene Classification	Valid Error	43.85±0.03	43.43±0.03	43.58±0.03	43.78±0.04	**43.19±0.01**
		Test Error	35.43±0.02	35.25±0.03	35.29±0.02	35.25±0.03	**35.00±0.01**
	Jigsaw	Valid Error	3.58±0.03	3.25±0.01	3.31±0.00	3.27±0.01	**3.17±0.01**
		Test Error	3.79±0.04	3.35±0.01	3.34±0.01	3.38±0.02	**3.29±0.01**
	Surface Normal	Valid Error	39.20±0.04	37.55±0.16	37.42±0.14	38.80±0.15	**36.65±0.20**
		Test Error	36.27±0.04	34.72±0.15	34.69±0.14	35.99±0.15	**33.91±0.19**
	Room Layout	Valid Error	59.98±0.04	59.83±0.03	59.95±0.06	60.19±0.05	**58.92±0.05**
		Test Error	53.94±0.06	55.54±0.10	54.02±0.03	54.72±0.09	**53.80±0.04**
	Semantic Segmentation	Valid Error	71.59±0.04	71.39±0.05	71.01±0.05	70.89±0.04	**70.51±0.01**
		Test Error	68.97±0.01	68.11±0.05	68.45±0.05	68.30±0.04	**67.98±0.03**
DARTS	CIFAR-10	Valid Error	5.90±0.07	5.50±0.09	5.42 ±0.14	5.73±0.07	**5.37±0.01**
		Test Error	3.28±0.09	2.87±0.04	2.72±0.07	2.93±0.12	**2.43±0.04**
	CIFAR-100	Test Error	21.47±0.08	19.75±0.13	20.62±0.12	19.95±0.17	**18.46±0.09**

into the design of the kernel function. Here we choose the Gaussian kernel, which is defined as:

$$\boldsymbol{K}(\boldsymbol{a}^i, \boldsymbol{a}^j) = \exp\left(-\text{HOTNN}(\boldsymbol{a}^i, \boldsymbol{a}^j)/\sigma_l^2\right), \qquad (14)$$

where σ_l is the length-scale hyperparameter, $\text{HOTNN}(\boldsymbol{a}^i, \boldsymbol{a}^j)$ is the HOTNN distance between two networks $\boldsymbol{a}^i$ and $\boldsymbol{a}^j$, the kernel has a higher value if the HOTNN distance is closer. The detailed algorithm description of our proposed HOTNAS method is given in the supplementary material Sec. S4.

4. Experiments

To prove the effectiveness of our proposed HOTNAS method, we compare it with a range of NAS methods on multiple modular cell-based search spaces across different tasks. Further, we visualize the relationship between HOTNN distance and the validation performance to make a reasonable interpretation for our proposed HOTNAS method (see supplementary material Sec. S5). Finally, we conduct ablative experiments on our proposed HOTNAS approach (see supplementary material Sec. S7).

Architecture Search on NAS-Bench-101 Benchmark The search space of NAS-Bench-101 is a single cell network architecture with 7 nodes and a maximum of 9 edges. The benchmark contains a total of 423,624 unique cell neural architectures. More detail of the NAS-Bench-101 benchmark can refer to [66]. We compare our proposed HOTNAS method to a range of existing Bayesian optimization for NAS methods, including BO with edit distance [23], NASBOT [25], BANANAS [59], BO-TW [40], BO-TW-2G [40], and NAS-BOWL [44]. We also include common baselines like random search [33] and evolutionary search [37] for comparison. The implementation details and experimental settings of these methods are summarized in the supplementary material Sec. S6. The best-found validation loss and test loss of different NAS methods on the NAS-Bench-101 dataset are summarized in Tab. S2. The best-found validation loss over the number of iterations of these NAS methods on the NAS-Bench-101 benchmark are presented in Fig. S3. The results show that our proposed HOTNAS method achieves the lowest valid loss and test loss, which reveals that our proposed HOTNAS method can find a neural architecture with better performance on the NAS-Bench-101 search space.

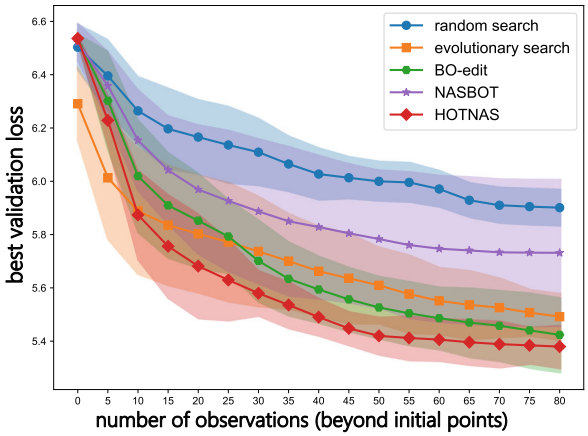


Figure 3. The best-found validation loss over the number of iterations (beyond initial points) of various NAS methods on the DARTS benchmark.

Architecture Search on TransNAS-Bench-101 Benchmark. The TransNAS-Bench-101 is a benchmark dataset

containing network performance across diverse tasks, including object classification, scene classification, room layout, jigsaw, autoencoding, surface normal, and semantic segmentation. We focus on searching its macro skeleton search space, where each network can stack a different number of modules, and each modular can decide various operations (*e.g.*, where to raise the channel or downsample the resolution). It consists of 3256 networks across seven tasks. More detail on this benchmark can refer to [14]. To demonstrate the effectiveness of our proposed HOTNAS method we compare it to several prevalent NAS methods, including random search [33], evolutionary search [37], BO with edit distance [23], and NASBOT [25]. Note that we have not included BANANAS [59], BO-TW [40], BO-TW-2G [40], NAS-BOWL [44] methods as they are limited to a single cell architecture search and ignore the differences in the macro-architecture of the whole network. These methods' implementation details and experimental settings are summarized in the supplementary material Sec. S6. The best-found validation loss and test loss of different NAS methods across seven tasks on the TransNAS-Bench-101 benchmark are summarized in Tab. 1. The best-found validation loss over the number of iterations of these NAS methods on the TransNAS-Bench-101 benchmark are presented in the supplementary material Fig. S3. Overall, our proposed HOTNAS method performs best among these NAS methods on all tasks. Especially on the autoencoding and surface normal tasks, our proposed HOTNAS method can find an architecture with significantly better performance. Besides, our proposed HOTNAS method has a low variance of the best-found validation loss on most tasks, which reveals its strong stability.

Architecture Search on DARTS Benchmark. The DARTS is a large-scale non-tabular benchmark about 10^{18} neural architectures. It consists of two types of cells *i.e.*, the normal cell and the reduced cell. Each cell can choose different operations and connections. The details of the DARTS search space can refer to [36]. Unlike most existing NAS methods on the DARTS benchmark that focus only on searching the cell's internal architectures, we also search the depth of the entire network. We choose the number of stacked cells from 5 to 20. This is challenging as the size of DARTS search space expands to 16×10^{18}. We compare our proposed HOTNAS method to several prevalent NAS methods, including random search [33], evolutionary search [37], BO with edit distance [23], and NASBOT [25]. These methods' implementation details and experimental settings are summarized in the supplementary material Sec. S6. We first search the networks on the training set of the CIFAR-10 dataset and then evaluate the searched best network on the test set of the CIFAR-10 dataset. Besides, we directly transfer the best-performing network searched on the CIFAR-10 dataset to the CIFAR-100 dataset and train it from scratch to test its generalization ability. The best-found validation losses in the CIFAR-10 dataset and the test losses in the CIFAR-10 and CIFAR-10 datasets of different NAS methods on the DART search space are shown in Tab. 1. Fig. 3 presents the best-found validation loss over the number of iterations of these NAS methods on the DARTS benchmark. The results indicate that our proposed HOTNAS method outperforms other BO for NAS methods. Note that the best-found network of our proposed HOTNAS method achieves a lower test error in the CIFAR-10 dataset, which demonstrates its excellent generalization ability. Besides, it still performs well on the CIFAR-100 dataset, which reveals our proposed HOTNAS method can find a network with better transferability.

5. Conclusion

In this paper, we design a hierarchical optimal transportation metric called HOTNN to measure the similarity between different networks by leveraging the hierarchical structure of the cell-based networks. Our proposed HOTNN metric considers the similarity between cells' internal architectures and the differences in the macro-architectures of various networks. We then integrate the HOTNN metric into the GP surrogate model in the Bayesian optimization framework called HOTNAS and demonstrate its efficacy by comparing the current popular Bayesian optimization for NAS methods. The experimental results show that our proposed method can find a neural network architecture with better performance. We hope our proposed method can extend to more complex cell-based search spaces in future work.

Acknowledgments

We thank the anonymous reviewers for their valuable and constructive suggestions and comments. This work is supported by the National Natural Science Foundation of China (No.62076234, No.92270110); the National Key Research and Development Project (No.2022YFB2703102); the Beijing Natural Science Foundation (No.4222029); the "Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China"; the Beijing Outstanding Young Scientist Program (No.BJJWZYJH012019100020098); the Public Computing Cloud, Renmin University of China; the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (No.2021030199); the Huawei-Renmin University joint program on Information Retrieval; the Unicom Innovation Ecological Cooperation Plan; the CCF-Huawei Populus Grove Fund; and Huawei-CAAI Award Fund.

References

[1] Mohamed S. Abdelfattah, Abhinav Mehrotra, Lukasz Dudziak, and Nicholas Donald Lane. Zero-cost proxies for lightweight NAS. In *ICLR*, 2021. 1

[2] David Alvarez-Melis, Tommi Jaakkola, and Stefanie Jegelka. Structured optimal transport. In *AISTATS*, pages 1771–1780, 2018. 12

[3] Barret Zoph ands Quoc V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017. 1

[4] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In *NeurIPS*, pages 21524–21538, 2020. 1, 6, 15

[5] Jean-David Benamou, Brittany D Froese, and Adam M Oberman. Numerical solution of the optimal transportation problem using the monge–ampère equation. *Journal of Computational Physics*, 260:107–126, 2014. 2, 12

[6] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *ICML*, pages 550–559, 2018. 1

[7] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010. 1, 12

[8] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *ICCV*, pages 1294–1303, 2019. 17

[9] Xiangxiang Chu, Bo Zhang, and Ruijun Xu. Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. In *ICCV*, pages 12219–12228, 2021. 1

[10] Jiequan Cui, Pengguang Chen, Ruiyu Li, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast and practical neural architecture search. In *ICCV*, pages 6508–6517, 2019. 1

[11] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 12

[12] Xuanyi Dong, Lu Liu, Katarzyna Musial, and Bogdan Gabrys. Nats-bench: Benchmarking NAS algorithms for architecture topology and size. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3634–3646, 2022. 1

[13] Xuanyi Dong and Yi Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. In *ICLR*, 2020. 1

[14] 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 *CVPR*, pages 5251–5260, 2021. 2, 8

[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 *CVPR*, pages 5251–5260, 2021. 2

[16] Mourad El Hamri, Younès Bennani, and Issam Falih. Hierarchical optimal transport for unsupervised domain adaptation. *Machine Learning*, pages 1–24, 2022. 12

[17] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. 16

[18] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018. 1, 12

[19] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *CVPR*, pages 303–312, 2021. 2, 12

[20] José Miguel Henrández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *NeurIPS*, pages 918–926, 2014. 1

[21] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *LION*, pages 507–523, 2011. 6

[22] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated Machine Learning: Methods, Systems, Challenges*. Springer Nature, 2019. 1, 13

[23] Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *SIGKDD*, pages 1946–1956, 2019. 1, 7, 8, 13

[24] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998. 1, 13, 15

[25] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabás Póczos, and Eric P. Xing. Neural architecture search with bayesian optimisation and optimal transport. In *NeurIPS*, pages 2020–2029, 2018. 1, 2, 3, 4, 7, 8, 13

[26] Leonid V Kantorovich. On the translocation of masses. *Journal of Mathematical Sciences*, 133(4):1381–1382, 2006. 12

[27] Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine*, 34(4):43–59, 2017. 2, 12

[28] Soheil Kolouri, Yang Zou, and Gustavo K. Rohde. Sliced wasserstein kernels for probability distributions. In *CVPR*, 2016. 15

[29] Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. In *ICLR*, 2023. 2

[30] H. J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1):97–106, 03 1964. 1, 15

[31] Tam Le, Makoto Yamada, Kenji Fukumizu, and Marco Cuturi. Tree-sliced variants of wasserstein distances. In *NeurIPS*, 2019. 13

[32] John Lee, Max Dabagia, Eva L Dyer, and Christopher J Rozell. Hierarchical optimal transport for multimodal distribution alignment. In *NeurIPS*, pages 13475–13485, 2019. 12

[33] Liam Li and Ameet Talwalkar. Random search and reproducibility for neural architecture search. In *UAI*, pages 367–377, 2019. 1, 7, 8

[34] Zhihang Li, Teng Xi, Jiankang Deng, Gang Zhang, Shengzhao Wen, and Ran He. GP-NAS: gaussian process based neural architecture search. In *CVPR*, pages 11930–11939, 2020. 2

[35] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. Hierarchical representations for efficient architecture search. In *ICLR*, 2018. 12

[36] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In *ICLR*, 2019. 1, 3, 8, 17

[37] Yuqiao Liu, Yanan Sun, Bing Xue, Mengjie Zhang, Gary G. Yen, and Kay Chen Tan. A survey on evolutionary neural architecture search. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2021. 1, 7, 8

[38] Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus, and Fabian-Robert Stöter. Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *ICML*, pages 4104–4113, 2019. 12

[39] Jisoo Mok, Byunggook Na, Ji-Hoon Kim, Dongyoon Han, and Sungroh Yoon. Demystifying the neural tangent kernel from a practical perspective: Can it be trusted for neural architecture search without training? In *CVPR*, pages 11861–11870, 2022. 1, 12

[40] Vu Nguyen, Tam Le, Makoto Yamada, and Michael A Osborne. Optimal transport kernels for sequential and parallel neural architecture search. In *ICML*, pages 8084–8095, 2021. 2, 3, 4, 7, 8, 13, 16, 17

[41] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 2, 12

[42] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *ICML*, pages 4095–4104, 2018. 1

[43] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys*, 54(4):1–34, 2021. 3

[44] Binxin Ru, Xingchen Wan, Xiaowen Dong, and Michael Osborne. Interpretable neural architecture search via bayesian optimisation with weisfeiler-lehman kernels. In *ICLR*, 2020. 1, 7, 8, 13, 17

[45] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving gans using optimal transport. In *ICLR*, 2018. 12

[46] Filippo Santambrogio. Optimal transport for applied mathematicians. *Progress in Nonlinear Differential Equations and Their Applications*, 87:XXVII, 353, 2017. 12

[47] Bernhard Schmitzer and Christoph Schnörr. A hierarchical approach to optimal transport. In *SSVM*, pages 452–464, 2013. 12

[48] Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large scale optimal transport and mapping estimation. In *ICLR*, 2018. 12

[49] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: a review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015. 1, 12

[50] Yao Shu, Wei Wang, and Shaofeng Cai. Understanding architectures learnt by cell-based neural architecture search. In *ICLR*, 2019. 3

[51] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep neural networks. In *ICML*, pages 2171–2180, 2015. 6

[52] Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. In *NeurIPS*, pages 4134–4142, 2016. 6

[53] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML*, 2010. 1, 15

[54] Fariborz Taherkhani, Ali Dabouei, Sobhan Soleymani, Jeremy Dawson, and Nasser M Nasrabadi. Transporting labels via hierarchical optimal transport for semi-supervised learning. In *ECCV*, pages 509–526, 2020. 12

[55] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, pages 2820–2828, 2019. 1, 3

[56] Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. Optimal transport for structured data with application on graphs. In *ICML*, pages 6275–6284, 2019. 3

[57] Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. TrajectoryNet: A dynamic optimal transport network for modeling cellular dynamics. In *ICML*, pages 9526–9536, Jul 2020. 2, 12

[58] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009. 12

[59] Colin White, Willie Neiswanger, and Yash Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. In *AAAI*, number 12, pages 10293–10301, 2021. 1, 2, 7, 8, 13, 15, 17

[60] Colin White, Arber Zela, Robin Ru, Yang Liu, and Frank Hutter. How powerful are performance predictors in neural architecture search? In *NeurIPS*, pages 28454–28469, 2021. 1

[61] Christopher K Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*, volume 2. MIT press Cambridge, MA, 2006. 6, 15

[62] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *CVPR*, pages 10734–10742, 2019. 1

[63] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR*, June 2020. 12

[64] 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 *ICLR*, 2020. 17

[65] Anna Yeaton, Rahul G Krishnan, Rebecca Mieloszyk, David Alvarez-Melis, and Grace Huynh. Hierarchical optimal transport for comparing histopathology datasets. *arXiv preprint arXiv:2204.08324*, 2022. 12

[66] Chris Ying, Aaron Klein, Eric Christiansen, Esteban Real, Kevin Murphy, and Frank Hutter. Nas-bench-101: Towards reproducible neural architecture search. In *ICML*, pages 7105–7114, 2019. 1, 7

[67] Mikhail Yurochkin, Sebastian Claici, Edward Chien, Farzaneh Mirzazadeh, and Justin Solomon. Hierarchical optimal transport for document representation. In *NeurIPS*, pages 1601–1611, 2019. 12

[68] Arber Zela, Julien Niklas Siems, Lucas Zimmer, Jovita Lukasik, Margret Keuper, and Frank Hutter. Surrogate nas benchmarks: Going beyond the limited search spaces of tabular nas benchmarks. In *ICLR*, pages 1–36, 2022. 1, 2

[69] Zhao Zhong, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Practical block-wise neural network architecture generation. In *CVPR*, 2018. 1

[70] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *CVPR*, pages 8697–8710, 2018. 1, 3