Does Unsupervised Architecture Representation Learning Help Neural Architecture Search?

Shen Yan, Yu Zheng, Wei Ao, Xiao Zeng, Mi Zhang

Michigan State University

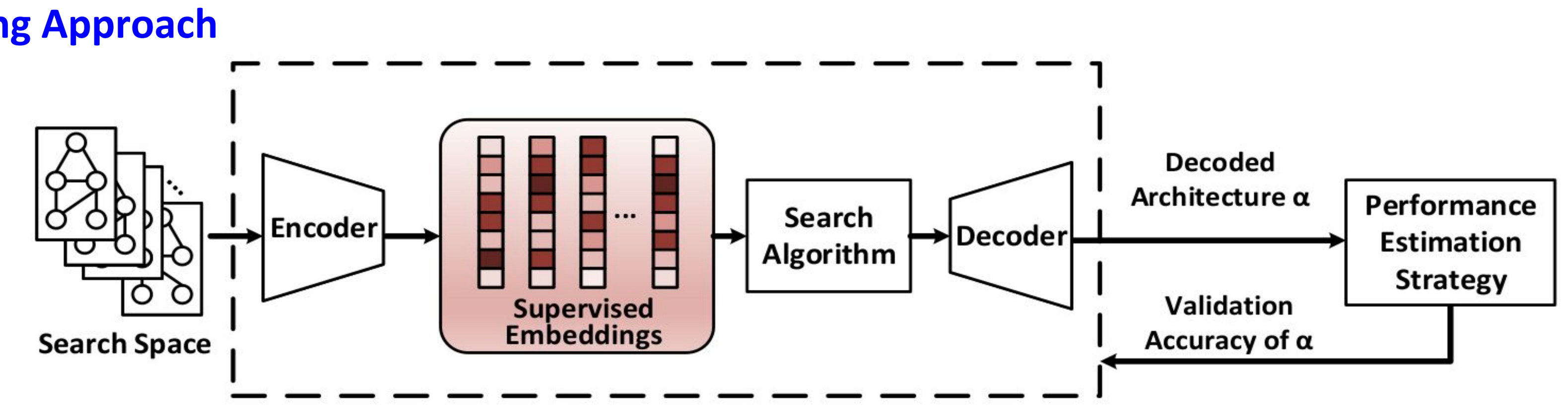
Introduction

Typical NAS methods encode the search space using the adjacency matrix-based encoding. However, the size of the adjacency matrix grows quadratically as search space scales up, making downstream architecture search less efficient in large search spaces [1]. To improve search efficiency, recent NAS methods propose to learn continuous embeddings of neural architectures [2,3]. In these methods, architecture embeddings and search algorithms are jointly optimized in a supervised way, guided by the accuracies of architectures selected by the search algorithms. However, it cannot necessarily improve embedding learning due to entangling architecture representation learning and architecture search together.

Method

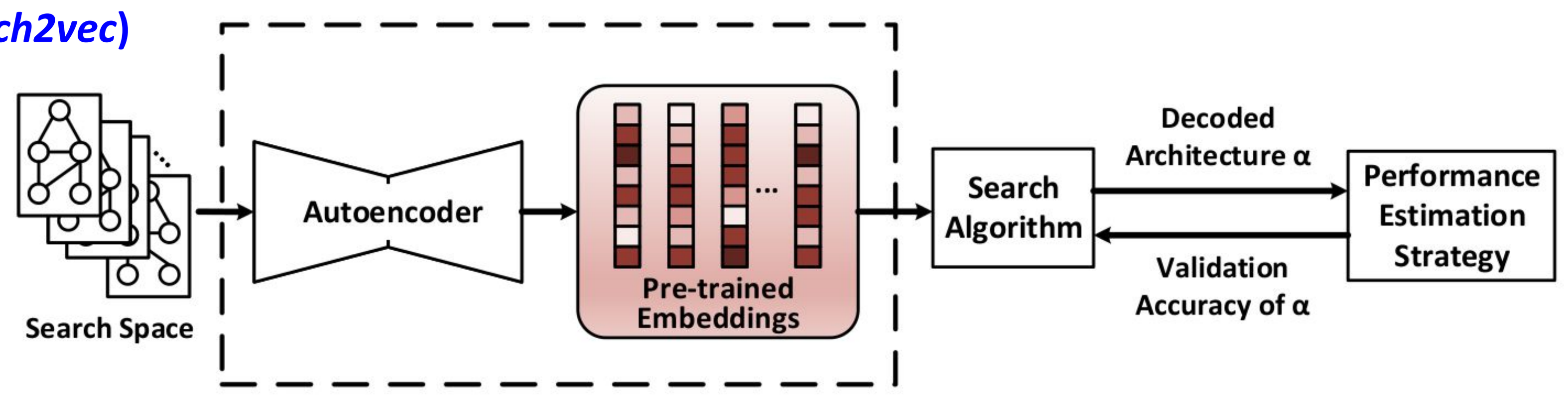
- **We propose *arch2vec*, a simple yet effective *unsupervised* architecture representation learning method for neural architecture search.**
- ***Decouple* architecture embedding learning and architecture search into two *separate* processes.**
- **Better *preserve local structure relationship* of neural architectures and helps construct a *smoother* latent space, which benefits downstream search.**

Existing Approach



Architecture embeddings and search strategies are jointly optimized in a supervised manner

Our Approach (*arch2vec*)



Pre-training architecture embeddings in an unsupervised manner

Let **A** denote **Adjacency Matrix**, **X** denote **Operation Matrix**. Augment **A** as $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{A}^T$ to transfer original directed graph into undirected one to allow bi-directional information flow.

Encoder: $q(\mathbf{Z}|\mathbf{X}, \tilde{\mathbf{A}}) = \prod_{i=1}^{N} q(\mathbf{z}_i|\mathbf{X}, \tilde{\mathbf{A}})$, with $q(\mathbf{z}_i|\mathbf{X}, \tilde{\mathbf{A}}) = \mathcal{N}(\mathbf{z}_i|\boldsymbol{\mu}_i, diag(\boldsymbol{\sigma}_i^2))$

L-layer Graph Isomorphism Network (GINs): $\mathbf{H}^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot \mathbf{H}^{(k-1)} + \tilde{\mathbf{A}}\mathbf{H}^{(k-1)}\right), k = 1, 2, \ldots, L$

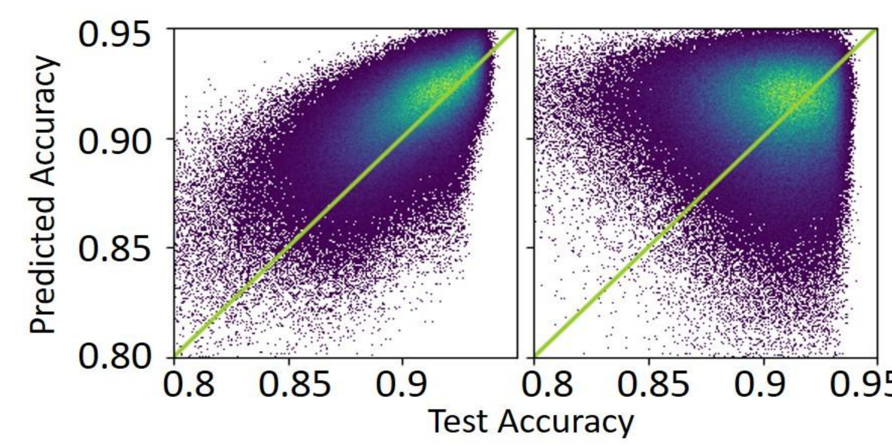
Decoder: $p(\hat{\mathbf{A}}|\mathbf{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{N} P(\hat{A}_{ij}|\mathbf{z}_i, \mathbf{z}_j)$, with $p(\hat{A}_{ij} = 1|\mathbf{z}_i, \mathbf{z}_j) = \sigma(\mathbf{z}_i^T \mathbf{z}_j)$,

$$p(\hat{\mathbf{X}} = [k_1, \ldots, k_N]^T|\mathbf{Z}) = \prod_{i=1}^{N} P(\hat{\mathbf{X}}_i = k_i|\mathbf{z}_i) = \prod_{i=1}^{N} \mathrm{softmax}(\mathbf{W}_o\mathbf{Z} + \mathbf{b}_o)_{i,k_i}$$

Training Objective: $\mathcal{L} = \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\tilde{\mathbf{A}})}[\log p(\hat{\mathbf{X}}, \hat{\mathbf{A}}|\mathbf{Z})] - \mathcal{D}_{KL}(q(\mathbf{Z}|\mathbf{X}, \tilde{\mathbf{A}})||p(\mathbf{Z}))$

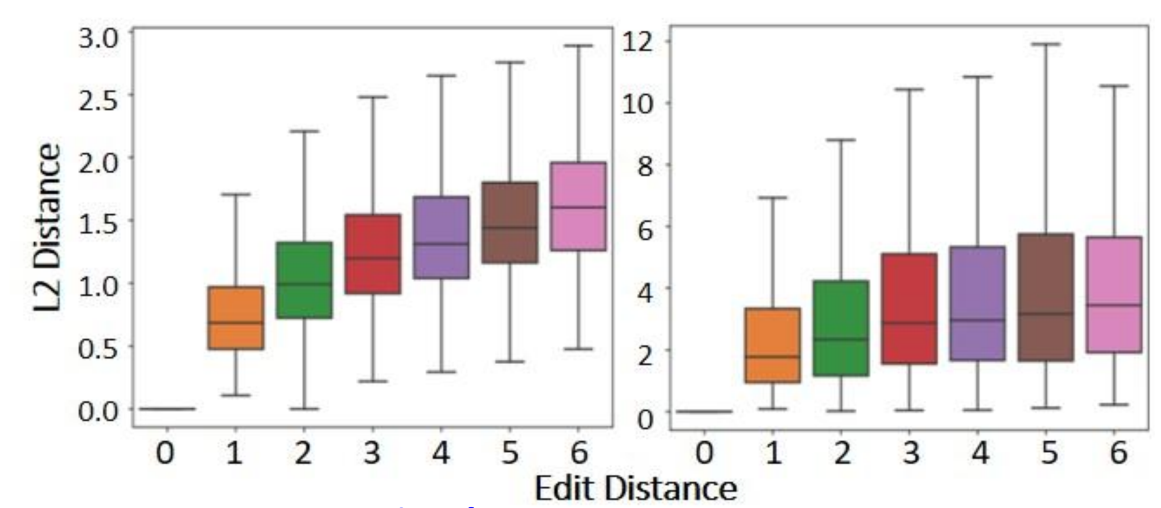
Experiments

- We compare the predictive performance of the pretrained embeddings and supervised embeddings. This metric measures how well the embeddings can predict the performance of the corresponding architectures.
- We train a Gaussian Process model with 250 sampled data to predict all data and report the results across 10 different seeds. We use RMSE and the Pearson correlation coefficient to evaluate points with test accuracy larger than 0.8.



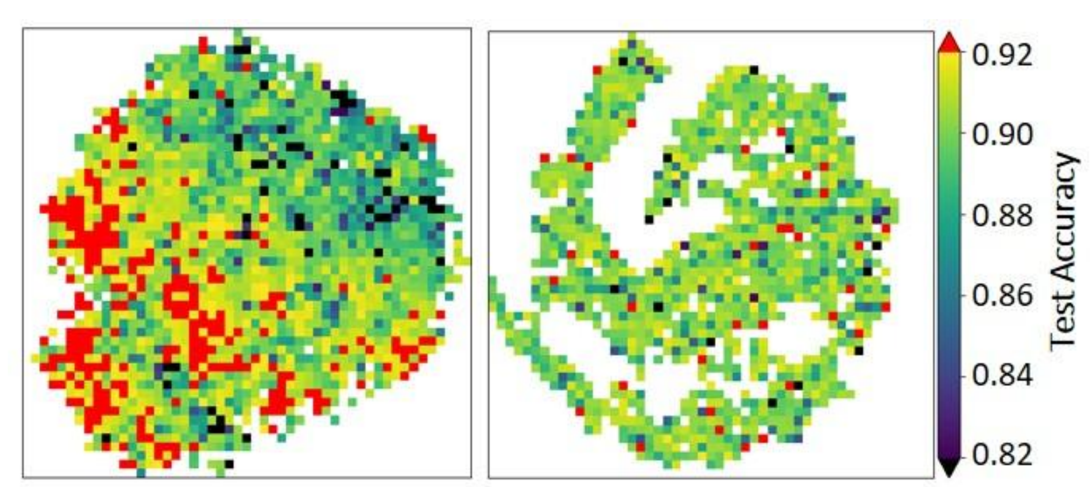
Pretrained Embeddings (ours) **Supervised Embeddings**

- We compare the distribution of L2 distance between architecture pairs by edit distance, measured by 1,000 architectures sampled in a long random walk with 1 edit distance apart from consecutive samples. The L2 distance of pretrained embeddings grows monotonically with increasing edit distance.
- This observation indicates that the pretrained embeddings are able to better capture the structural information of neural networks, and thus make similar architectures clustered better.



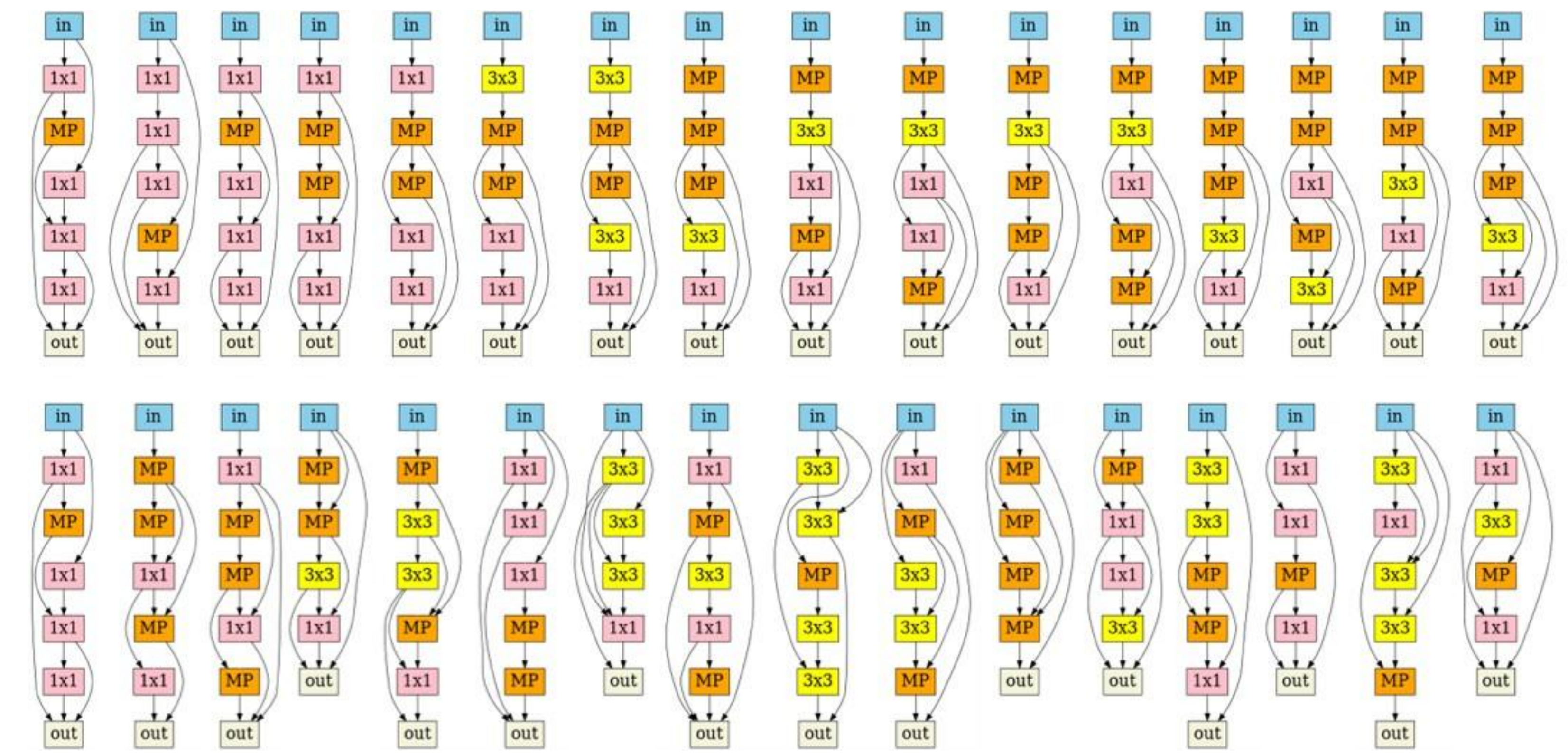
Pretrained Embeddings (ours) **Supervised Embeddings**

- We visualize the latent spaces learned by *arch2vec* and its supervised learning counterpart in 2-dimensional space. Compared to supervised embeddings, pretrained embeddings span the whole latent space, and architectures with similar accuracies are clustered and distributed more smoothly in the latent space.
- Conducting architecture search on such smooth performance surface is much easier and is hence more efficient.



Pretrained Embeddings (ours) **Supervised Embeddings**

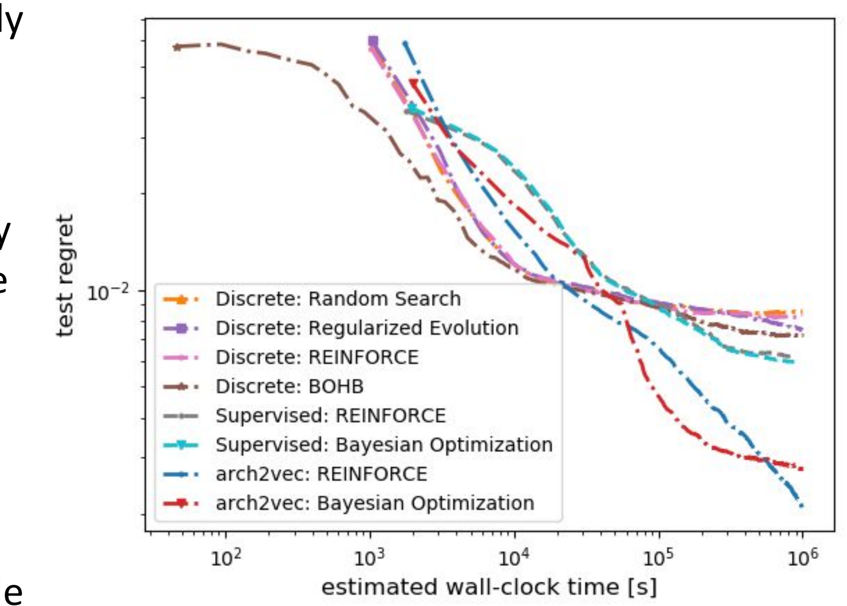
- We visualize a sequence of architecture cells decoded from the learned latent space of arch2vec and supervised approach.



Top: Pretrained Embeddings (edit distances between adjacent architectures are 4, 6, 1, 5, 1, 1, 1, 5, 2, 3, 2, 4, 2, 5, 2)

Bottom: Supervised Embeddings (edit distances between adjacent architectures are 8, 6, 7, 7, 9, 8, 11, 11, 6, 10, 10, 11, 10, 11, 9)

- In **NAS-Bench-101**, *arch2vec* considerably outperforms its supervised counterpart and the discrete encoding after 50,000 wall clock seconds.
- In **NAS-Bench-201**, *arch2vec* consistently outperforms other approaches on all the three datasets, leading to better validation and test accuracy as well as reduced variability.
- In **DARTS**, *arch2vec* leads to competitive search performance among different cell-based NAS methods with comparable model parameters.



NAS Methods	CIFAR-10		CIFAR-100		ImageNet-16-120	
	validation	test	validation	test	validation	test
RE [41]	91.08±0.43	93.84±0.43	73.02±0.46	72.86±0.55	45.78±0.56	45.63±0.64
RS [59]	90.94±0.38	93.75±0.37	72.17±0.64	72.05±0.77	45.47±0.65	45.33±0.79
REINFORCE [10]	91.03±0.33	93.82±0.31	72.35±0.63	72.13±0.79	45.58±0.62	45.30±0.86
BOHB [12]	90.82±0.53	93.61±0.52	72.59±0.82	72.37±0.90	45.44±0.70	45.26±0.83
arch2vec-RL	91.32±0.42	94.12±0.42	73.13±0.72	73.15±0.78	46.22±0.30	46.16±0.38
arch2vec-BO	**91.41±0.22**	**94.18±0.24**	**73.35±0.32**	**73.37±0.30**	**46.34±0.18**	**46.27±0.37**

	Test Error		Params (M)	Search Cost				
NAS Methods	Avg	Best		Stage 1	Stage 2	Total	Encoding	Search Method
Random Search [15]	3.29±0.15	-	3.2	-	-	4	-	Random
ENAS [61]	-	2.89	4.6	0.5	-	-	Supervised	REINFORCE
ASHA [62]	3.03±0.13	2.85	2.2	-	-	9	-	Random
RS WS [62]	2.85±0.08	2.71	4.3	2.7	6	8.7	-	Random
SNAS [16]	2.85±0.02	-	2.8	1.5	-	-	Supervised	GD
DARTS [15]	2.76±0.09	-	3.3	4	1	5	Supervised	GD
BANANAS [43]	2.64	2.57	3.6	100 (queries)	-	11.8	Supervised	BO
Random Search (ours)	3.1±0.18	2.71	3.2	-	-	4	-	Random
DARTS (ours)	2.71±0.08	2.63	3.3	4	1.2	5.2	Supervised	GD
BANANAS (ours)	2.67±0.07	2.61	3.6	100 (queries)	1.3	11.5	Supervised	BO
arch2vec-RL	2.65±0.05	2.60	3.3	100 (queries)	1.2	9.5	Unsupervised	REINFORCE
arch2vec-BO	**2.56±0.05**	**2.48**	3.6	100 (queries)	1.3	10.5	Unsupervised	BO

Reference

[1] Neural Architecture Search: A Survey. Elsken et. al., JMLR 2019.
[2] Neural Architecture Optimization. Luo et. al., NeurIPS 2018.
[3] Darts: Differentiable architecture search. Liu et. al., ICLR 2019.