Supplementary Material: Invariant Structure Learning for Better Generalization and Causal Explainability

Yunhao Ge[♦,★,*]**, Sercan Ö. Arık**[♦]**, Jinsung Yoon**[♦]**, Ao Xu**[★]**, Laurent Itti**[★]**, and Tomas Pfister**[♦]
{yunhaoge, aoxu, itti}@usc.edu, {soarik, jinsungyoon, tpfister}@google.com

[♦] *Google Cloud AI, Sunnyvale, CA, USA*
[★] *University of Southern California, Los Angeles, CA, USA*

Reviewed on OpenReview: *https://openreview.net/forum?id=A9yn7KTwsK*

A Selection of ISL hyperparameters

Thresholds: As described in Sec. 3.2 and Algorithm 1, after Eq.3 converges at all environments, we employ a threshold t to convert the adjacency matrix W to a DAG. To find a proper threshold, we use the following strategy. We set a minimum edges number E_{min} and a maximum edges number E_{max} based on the dataset information. Usually, E_{min} is half of the number of nodes $|E|/2$ and E_{max} is $5|E|$. We also set a range of threshold $t \in [t_{min}, t_{max}]$ and a step size t_s base on the value range of W. Usually we use $t_{min} = \min(W)$ and $t_{max} = \max(W)$. Then, we employ a grid search over the range $[t_{min}, t_{max}]$ with a step size t_s, and keep the thresholds and corresponding DAG that satisfied the following requirements: (1) The graph after the filtering with threshold should be a DAG (no cyclicity). (2) The number of graph edges $E_{min} < E < E_{max}$. For the selected threshold values and DAGs, we remove the duplicated group as different threshold may obtain the same DAG, which further refines the interval. Then, for each threshold, we use the selected $Pa(Y)$ as input to train a one-layer MLP to predict Y and select the threshold t that has smallest Y reconstruction error in the validation set.

Regularization coefficients: For training of ISL, we use different loss terms. The hyperparameter γ controls the trade off between Y reconstruction and DAG constrain among environments. As we decrease the value of γ, the training would focus more on the target Y reconstruction. We also have 4 regularization hyperparameters: $\mathcal{L}_{sparse}(\theta) = \beta_1||\theta_1^Y||_1 + \beta_2||\theta_r^Y||_2 + \beta_3||\theta^X||_1 + \beta_4||\theta^X||_2$, where $||\cdot||_1$ and $||\cdot||_2$ denote l_1 and l_2 regularization. β_1 controls the importance of the l_1 regularization on the θ_1^Y, increasing β_1 makes the selection of $Pa(Y)$ more conservative (most of the values of the first column in W would be zero). β_2 helps avoid overfitting of $h(\cdot)$. β_3 and β_4 controls the regularization on θ^X. We choose the value of γ and β_i that achieves the smallest target Y reconstruction on the validation set. We find the parameters: $\gamma = 1; \beta_1 = 0.001; \beta_2 = 0.01; \beta_3 = 0.01; \beta_4 = 0.01$ as reasonable choices across many different settings, although they are not extensively optimized.

Table. 1 shows the results on Boston Housing for the prediction target of median value of homes (MED) ISL with different regression parameters. We demonstrate that the results are not too sensitive to the change of regularization. That is because the regularization coefficients mainly influence the DAG learning process, and we apply fine-tuning for $h(\cdot)$ after convergence of DAG learning, which provides a mechanism to mitigate the differences at the first training stage.

B Building environments

To show the efficacy of the proposed unsupervised environment building method based on k-means clustering, we present comparisons to the setting with the environment building based on known data source information, i.e. the data comes with the indication on how the environments are split based on data collection or

[*] Work done while at Google

Table 1: Boston Housing median value of homes (MED) target prediction results by ISL with different regression parameters.

γ	β_1	β_2	β_3	β_4	MSE (↓)
1	0.001	0.01	0.01	0.01	0.052
5	0.001	0.01	0.01	0.01	0.054
1	0.001	0.001	0.01	0.01	0.057

generation process. Table. 2 shows that their difference is quite small (much smaller than the outperformance of ISL compared to the other methods) and the proposed ISL is highly effective and robust with unsupervised environment building by clustering.

We employed standard clustering evaluation techniques such as the Elbow and Silhouette Method. These methods enabled us to assess various cluster numbers systematically, identifying the point at which adding more clusters led to diminishing returns (the "elbow") and evaluating how well each object lies within its cluster (the silhouette score). More specifically, for the Elbow method, we plot the sum of squared distances from each point to its assigned center (known as inertia or within-cluster sum of squares) for a range of values of K (e.g., K from 1 to 10) and automatically detect the "elbow" point on the plot, where the reduction in inertia starts to slow down. For the silhouette method, we calculate its silhouette score, which measures how similar a point is to its cluster compared to other clusters for each data point, and compute the average silhouette score for different values of K and plot them. An algorithm is then applied to look for an optimal value K that maximizes the average silhouette score across all data points. This process allowed us to arrive at an optimal K value, and our extensive experimentation and analysis demonstrated that our approach remains robust across different numbers of environments. Our empirical evaluations provide further evidence of the effectiveness and stability of our selection method.

Table 2: Unsupervised vs. supervised environment building for ISL on synthetic data. We observe very small difference between them, showing the efficacy of the proposed unsupervised environment construction mechanism.

Number of nodes	Metrics (↓)	Supervised	Unsupervised
3 (c=2, s=1)	ID MSE	**0.005** ±0.0001	**0.005** ±0.0001
	OOD MSE	**0.010** ±0.0002	**0.010** ±0.0002
	Average SHD	**0**±0	**0**±0
4 (c=2, s=2)	ID MSE	**0.006** ±0.0002	**0.006** ±0.0002
	OOD MSE	**0.009** ±0.0001	**0.009** ±0.0001
	Average SHD	**0**±0	**0**±0
5 (c=3, s=2)	ID MSE	**0.004** ±0.0001	**0.004** ±0.0001
	OOD MSE	**0.004** ±0.0001	**0.004** ±0.0001
	Average SHD	**0**±0	**0**±0
9 (c=4, s=5)	ID MSE	**0.004** ±0.0006	**0.004** ±0.0006
	OOD MSE	**0.005** ±0.0001	**0.005** ±0.0001
	Average SHD	**0**±0	**0**±0
20 (c=10, s=10)	ID MSE	**0.007** ±0.0005	**0.009** ±0.0005
	OOD MSE	**0.007** ±0.0001	**0.061** ±0.009
	Average SHD	**1**±0	**2**±1

C Error statistics

In this section, we present the standard deviations for the errors to highlight the statistical significance of ISL improvements (Table. 3 and Table. 4). Overall, the improvements of ISL are much larger than the standard deviation values. In addition, the variance of performance is observed to be lower for ISL compared to the other methods, indicating its superiority in robustness.

Table 3: Supervised learning experiment results on real-world data along with their standard deviations. Note that MLP and CASTLE cannot provide DAGs (and thus don't have SHD values).

		MSE (↓)				SHD (↓)	
Dataset	Target	MLP	NOTEARS-MLP	CASTLE	ISL (Ours)	NOTEARS-MLP	ISL (Ours)
Boston Housing	MED	0.16 ±0.02	0.12 ±0.03	0.10 ±0.01	**0.05** ±0.008	2 ±0	**1**±0
Insurance	'PropCost'	0.40 ±0.02	0.99 ±0.02	0.36 ±0.001	**0.34** ±0.004	2 ±0	**0**±0
	'MedCost'	0.69 ±0.09	1.03 ±0.01	0.55 ±0.03	**0.52** ±0.002	2 ±1	**0**±0
	'LiabilityCost'	0.94 ±0.08	0.39 ±0.01	0.38 ±0.06	**0.25** ±0.0004	1 ±0	**0**±0
	'CarValue'	0.23 ±0.01	0.60 ±0.05	0.23 ±0.03	**0.23** ±0.0004	2 ±0	**1**±0

Table 4: Synthetic tabular data experiments in supervised learning setting. Note that black-box MLP and CASTLE can't provide DAGs. ISL yields lower MSE for ID and OOD, and lower SHD.

Number of nodes	Metrics (↓)	MLP	NOTEARS-MLP	CASTLE	ISL (Ours)
3 (c=2, s=1)	ID MSE	0.008 ±0.002	0.101 ±0.010	0.016 ±0.007	**0.005** ±0.0001
	OOD MSE	0.016 ±0.002	0.195 ±0.005	0.017 ±0.004	**0.010** ±0.0002
	Average SHD	-	2 ±0	-	**0**±0
4 (c=2, s=2)	ID MSE	0.006 ±0.009	0.087 ±0.005	0.017 ±0.002	**0.006** ±0.0002
	OOD MSE	0.040 ±0.022	0.174 ±0.024	0.036 ±0.010	**0.009** ±0.0001
	Average SHD	-	2 ±0	-	**0**±0
5 (c=3, s=2)	ID MSE	0.004 ±0.002	0.110 ±0.018	0.025 ±0.006	**0.004** ±0.0001
	OOD MSE	0.004 ±0.002	0.078 ±0.020	0.019 ±0.004	**0.004** ±0.0001
	Average SHD	-	3 ±0	-	**0**±0
9 (c=4, s=5)	ID MSE	0.012 ±0.006	0.070 ±0.010	0.034 ±0.010	**0.004** ±0.0006
	OOD MSE	0.052 ±0.024	0.201 ±0.028	0.152 ±0.022	**0.005** ±0.0001
	Average SHD	-	4 ±0	-	**0**±0
20 (c=10, s=10)	ID MSE	0.009 ±0.008	0.061 ±0.011	0.121 ±0.021	**0.007** ±0.0005
	OOD MSE	0.094 ±0.061	0.303 ±0.050	0.272 ±0.046	**0.007** ±0.0001
	Average SHD	-	9 ±0	-	**1**±0

D Time Complexity and Scalability Comparison

With respect to runtime and scalability, our runtime is 3 to 5 times greater than that of NOTEAR **?** due to the additional clustering step. The computational complexity of our framework aligns closely with that of NOTEARS. Specifically, the complexity for NOTEARS-MLP is given by $O(nd^2m + d^2m + d^3)$ FLOPS per iteration of L-BFGS-B, where n is the number of data samples, d is the number of nodes, and m is the number of edges.

In our framework, as we partition the dataset into k different environments, each requiring its own convergence, the effective runtime becomes $O(k(nd^2m + d^2m + d^3))$. However, since k (the number of environments) is generally a small constant—typically ranging from 3 to 5—the overall time complexity remains on the same order as $O(nd^2m + d^2m + d^3)$.

The following table summarizes some of the quantitative results we have recorded. The time measurements were obtained on an Apple M1 Pro chip with 16GB of memory.

Experiment	NOTEAR time (s)	ISL (ours)	Self-supervised ISL (ours)
x2s1	32.6	100.5	280.5
x2s2	48.1	139.7	400.2
x3s2	62.9	188.5	520.8
x4s5	139.0	420.1	1220.3
x10s10	414.5	1300.4	3600.4

Table 5: Time benchmarks for NOTEAR and our proposed methods.

E Discussion

While we have indeed demonstrated our idea through an image classification task, this example serves more to elucidate the intuition of our algorithm rather than showcase a specific target application. Our work primarily focuses on tabular data because most of the real world domains like economics, biology, and social social that we are interested in applying the causal discovery algorithm gather data in structured, tabular form and most causal discovery algorithms such as conditional independence tests and structural learning algorithms are designed to work with structured data. We have evaluated our approach on two well-known real-world datasets: the Sachs dataset [9] and the Insurance dataset. Both of these datasets are widely recognized and extensively used as benchmarks in the field of causal discovery (Zheng et al., 2018; 2020; He et al., 2021; Wei et al., 2020; Yu et al., 2019). We have clarified this in the revised manuscript.

F Limitations and the societal impact

In this paper, we propose a novel method for causal structure discovery, which can improve the explainability and generalization of key machine learning use cases. Lack of their explainability remains to be a bottleneck for widespread adoption of DNNs for many high-stakes applications, such as from Healthcare, Finance, Public Sector, Insurance, Legal etc. There are other forms of explainability methods used in practice, but since they cannot explicitly distinguish the causality from the correlations, there are many cases that they cannot satisfy the high bar for explainability in such applications. We believe that our method constitutes an important contribution towards this, as it can be directly adopted to applications where obtaining accurate causal explanations is crucial. In some cases, causal explanations can uncover the undesired biases in the data such as when the dominant factor for the output label comes from one of the features that corresponds to a sensitive attribute such as gender. In these cases, the causal explanations can be further validated with additional analyses (as our model is still far away from achieving the perfect SHD of 0 on complex real-world data with many features), and if they seem to be convincing, further data manipulation or model debiasing actions can be performed. In addition to causal explainability, the improved generalization aspect is expected to play a major positive role, as the distribution differences between training and testing settings can sometimes hinder the reliability of machine learning models. In some applications where the data collection is limited to certain locations or times or subsets, our method can be utilized to enhance the performance of the trained models when they are deployed to operate for different locations or times or subsets.

Overall, we believe there is significant room for improvement in causal structure discovery. Especially for complex real-world data with many features, the obtained SHD values are not very low in the literature. Further research in unsupervised environment building with better representation learning, end-to-end approaches in combining graph discovery and supervised learning, and adding more nonlinearity to the model to make it higher capacity in a systematic way, can be promising towards this direction. We demonstrate the robustness of our model in various settings, but further exploration of theoretical convergence guarantee can be useful as well. Lastly, methods to improve hyperparameter tuning and model selection with small validation data, without relying on ground truth causal graph structure, would be of high value.

References

Yue He, Peng Cui, Zheyan Shen, Renzhe Xu, Furui Liu, and Yong Jiang. Daring: Differentiable causal discovery with residual independence. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge*

Discovery & Data Mining, pp. 596–605, 2021.

Dennis Wei, Tian Gao, and Yue Yu. Dags with no fears: A closer look at continuous optimization for learning bayesian networks. *Advances in Neural Information Processing Systems*, 33:3895–3906, 2020.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pp. 7154–7163. PMLR, 2019.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pp. 3414–3425. PMLR, 2020.