TEST TIME TRAINING FOR SUPERVISED CAUSAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised Causal Learning (SCL) has shown promise in causal discovery by framing it as a supervised learning problem. However, it suffers from significant out-of-distribution generalization challenges. We reveals three fundamental limitations of previous SCL practices: fragility to distribution shifts, failure in compositional generalization, and a significant performance gap between synthetic benchmarks and real-world data, collectively questioning its real-world applicability. To address this, we propose Test-Time Training for Supervised Causal Learning (TTT-SCL), a novel framework that dynamically generates training data explicitly aligned with any specific test instance. We find that the similarity between training and test data can be implicitly captured through distributional alignment, which we operationalize via a proposed Alignment of Distribution (AD) metric. To prevent degenerate solutions and enforce causal minimality, we incorporate sparsity constraints into the optimization. Building on this foundation, we introduce Test-time Aligned Causal Training with Informed Construction (TACTIC), the first instantiation of TTT-SCL, which jointly optimizes AD and sparsity via stochastic graph refinement to dynamically generate aligned training data. Extensive experiments on synthetic benchmarks, pseudo-real and real-world dataset demonstrate that TACTIC significantly outperforms existing SCL and traditional causal discovery methods.

1 INTRODUCTION

Causal discovery aims to infer causal relationships from observational data (Pearl, 2009; Spirtes et al., 2000). Supervised Causal Learning (SCL) has recently emerged as a promising paradigm that approaches causal discovery as a supervised learning problem (Dai et al., 2023; Lorch et al., 2022; Ke et al., 2022; Zhang et al., 2025). During training, a causal instance comprising causal graph G^k_{train} [1] and associated dataset D^k_{train} is either collected or synthetically generated. Specifically, a graph G^k_{train} is sampled from the DAG space and parameterized by assigning specific causal mechanisms and noise distributions. The corresponding dataset D^k_{train} is then generated through forward sampling from this parameterized graph. The SCL model learns to map input data D^k_{train} to output graph G^k_{train}. At test time, given a test data D_{test}, the trained model predicts the underlying causal graph G_{test}.

A key factor influencing the success of SCL is the design of the training data: what properties should training instances $\{(D^k_{train}, G^k_{train})\}_{k=1}^K$ possess to ensure strong performance on an unknown test instance D_{test}? Two complementary principles emerge: **diversity** and **concentration**. Diversity seeks broad coverage of generative factors, including variations in graph structures, causal mechanisms, and noise distributions, to enhance generalization. Concentration, in contrast, aims to align training data with the specific characteristics of the test domain.

Current SCL approaches largely prioritize diversity, pre-training on synthetic data generated from varied categories of graph, mechanism and noise (Lorch et al., 2022; Ke et al., 2022). However, achieving true diversity is inherently intractable due to the super-exponential size of the DAG space and the uncountable set of mechanism spaces. As a result, it performs well in-distribution but suffers

[1] The superscript indicates that the instance comes from the training set, and the subscript indicates the k-th instance in the training set.

severe performance degradation under distribution shifts. Further, we design a series of experiment to illustrate the three fundamental limitations of previous SCL practices: fragility to distribution shifts, failure in compositional generalization, and a significant performance gap between synthetic benchmarks and real-world data, collectively questioning its real-world applicability.

These findings motivate a shift from diversity to concentration, i.e., constructing training data aligned with the specific properties of the test instance. In particular, we operationalize this idea through **Test-Time Training for Supervised Causal Learning (TTT-SCL)**, which replaces static pre-training with dynamic adaptation. Under TTT-SCL, given a test instance D_{test}, we generate targeted training instances on-the-fly, train a specialized SCL model, and apply it to infer G_{test}. In this sense, **concentration** is realized through instance-specific, distributionally aligned training data generation. Therefore, the central question becomes: how can we ensure alignment between the generated training data and the test instance?

Our key insight is to leverage the similarity of data distribution through what we term Structure-Induced Mechanism (SIM). Suppose a generated graph G^k_{train} matches G_{test} (i.e., $G^k_{train} = G_{test}$), then, mechanisms can be regressed from D_{test} using G^k_{train}, and synthetic data D^k_{train} can be forward-sampled to closely approximate D_{test}. While exact graph matches are rare, structurally similar graphs can still yield distributionally similar data under SIM. Thus, data distributional similarity emerges as a proxy for causal alignment.

To quantify the data distributional similarity between D^k_{train} and D_{test}, we propose Alignment of Distribution (**AD**), a metric that implicitly captures both structural and mechanistic similarity. However, relying solely on distributional similarity can lead to degenerate solutions, such as overly dense graphs that match the data distribution but violate causal minimality. To address this, we incorporate a sparsity constraint on G^k_{train}, ensuring sparse structures and preventing overfitting to spurious edges. Together, AD and sparsity provide a tractable measure of training instance quality.

Building on this foundation, we introduce Test-time Aligned Causal Training with Informed Construction (**TACTIC**), the first instantiation of TTT-SCL. TACTIC jointly optimizes AD and sparsity via stochastic graph refinement, dynamically generating training data aligned with D_{test}. Experiments on synthetic benchmarks, pseudo-real and real-world dataset demonstrate that TACTIC consistently outperforms existing methods.

Our main contributions are as follows:

1. We reveals three fundamental limitations of static SCL pre-training: fragility to distribution shifts, failure in compositional generalization, and a significant performance gap between synthetic benchmarks and real-world data, collectively questioning its real-world applicability.
2. We introduce the TTT-SCL framework, enabling dynamic generation of aligned training data at test time. This includes the formulation of AD as a tractable metric for causal similarity via distributional alignment, and a sparsity constraint that ensures causal minimality and avoids degenerate graphs.
3. We propose TACTIC, the first concrete method under TTT-SCL. TACTIC dynamically constructs effective training datasets tailored to each test instance, achieving excellent performance across both synthetic, pseudo-real and real-world datasets.

2 BACKGROUND

We begin by formalizing the core components of causal learning. A Structural Causal Model (SCM) consists of three key elements: causal graph, causal mechanisms, and noise distributions (Pearl, 2009; Peters et al., 2017). Specifically:

- Causal Graph: Let $G = (V, E)$ be a Directed Acyclic Graph (DAG) with vertex set $V = \{X_1, \ldots, X_d\}$ and edge set $E \subseteq V \times V$, where d is the number of variables. The adjacency matrix $A \in \{0, 1\}^{d \times d}$ encodes edge relationships where $A_{ij} = 1$ iff $X_i \to X_j \in E$.
- Causal mechanisms and noise: Each variable X_i is generated by a causal mechanism and exogenous noise. In this work, we focus on the Additive Noise Model (ANM) (Hoyer et al., 2008), a

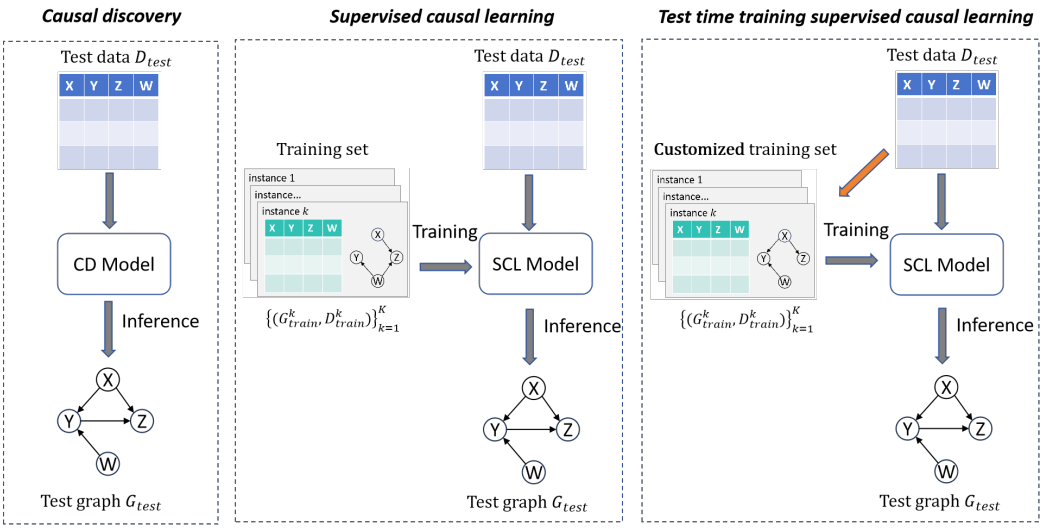

Figure 1: TTT-SCL compare with SCL and CD.

common assumption that ensures the causal structure is identifiable from observational data. This is formalized as:

$$X_i = f_i(\mathbf{Pa}_G(X_i)) + \varepsilon_i, \tag{1}$$

where $\mathbf{Pa}_G(X_i)$ denotes parents of X_i in G, $f_i : \mathbb{R}^{|\mathbf{Pa}_G(X_i)|} \to \mathbb{R}$ is the causal mechanism, and ε_i is exogenous noise. The full SCM is thus characterized by the tuple $(G, \{f_i\}_{i=1}^d, \{\varepsilon_i\}_{i=1}^d)$.

In supervised causal learning, we work with **causal instances**. A causal instance is defined by a graph G and a dataset D containing n observations $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\} \in \mathbb{R}^{n \times d}$, generated from the SCM $(G, \{f_i\}_{i=1}^d, \{\varepsilon_i\}_{i=1}^d)$. The **training set** comprises K such instances, denoted as $\{(D^k_{train}, G^k_{train})\}_{k=1}^K$, where each D^k_{train} is generated from its corresponding G^k_{train}. Similarly, at test time, we are given a single **test instance** (D_{test}, G_{test}), where D_{test} is observed but G_{test} is unknown. To avoid notation clutter, we adopt the following conventions: indices i, j refer to variable/node indices within a graph, and subscripts "train" and "test" distinguish between training and test entities.

Causal discovery aims to estimate the causal graph G_{test} from D_{test} using a model or algorithm M. Supervised causal learning (SCL) frames this as a supervised learning problem, where a model (typically a neural network) is trained on synthetic causal instances to learn a mapping from observational data to graph structures. Formally, the SCL objective is to learn:

$$\mathcal{M} : \mathbb{R}^{n \times d} \to \{0, 1\}^{d \times d}, \tag{2}$$

which maps an input data matrix (e.g., D_{test}) to an output adjacency matrix (representing G_{test}). The model is trained on synthetic pairs $\{(D^k_{train}, G^k_{train})\}_{k=1}^K$.

Previous SCL methods rely on training with **synthetic data**, where the generative distribution is explicitly controlled along three dimensions consistent with the SCM framework: graph structure, causal mechanisms, and noise distributions (Lorch et al., 2022; Ke et al., 2022; Froehlich & Koeppl, 2024). Typically, graphs are sampled from random graph models (e.g., Erdős–Rényi (Gilbert, 1959), Scale-Free (Barabási, 2009); mechanisms are chosen from a limited set of function classes (e.g., Linear, Random Fourier features (Rahimi & Recht, 2007)); and noise is drawn from parametric families (e.g., Gaussian, Uniform).

3 Out-of-Distribution Challenges for SCL

Out-of-distribution generalization has long been a challenge in machine learning, and we will show that it poses particularly severe implications for SCL. Unlike conventional ML domains where real-world training data is often available, SCL faces a fundamental constraint: causal graphs are rarely

available for real-world datasets. This forces SCL methods to rely largely on synthetic training data, making the bridge between synthetic simulation and real-world application the primary bottleneck for SCL.

Current SCL models are typically evaluated under constrained synthetic shifts, for instance, training and testing on the same mechanism type with slightly different parameter ranges. While such evaluations demonstrate robustness to mild parametric variations, they represent a weak form of generalization that remains within synthetic data distributions. These approaches cover only narrow mechanism families, while real-world causal relationships may involve complex, unmodeled functional forms. When test mechanisms fall outside the convex hull of training mechanisms, structural diversity alone cannot guarantee accurate estimation.

We point out three issues in previous SCL practices that collectively undermine their real-world applicability. First, these models are vulnerable to distribution shifts, exhibiting performance degradation when test distributions differ categorically from training in graph structure, mechanisms, or noise **(Issue 1)** . Second, they fail in compositional generalization, as models trained on diverse components cannot handle novel combinations of them, suggesting mere memorization of training configurations rather than learning modular causal representations **(Issue 2)**. Third, and most critically, they show divergent generalization patterns where strong performance on synthetic benchmarks fails to translate to real-world data, revealing a fundamental overfitting to the synthetic domain **(Issue 3)**. We use a series of experiments to illustrate these issues.

3.1 Experiment setup

To comprehensively evaluate generalization, we use both synthetic benchmarks, pseudo-real and real-world dataset.

Synthetic data: We generate test instances from a factorial combination of distributions. A test instance is defined by the tuple $(G_{test}, f_{test}, \varepsilon_{test})$.

- Graph (G): We use two random graph models: Erdos-Renyi (ER) and Scale-Free (SF) (Gilbert, 1959; Barabási, 2009).
- Mechanism (f): We use three function classes: Linear, Random Fourier Features (RFF) (Rahimi & Recht, 2007), and Chebyshev polynomials (Froehlich & Koeppl, 2024).
- Noise (ε): Gaussian noise is used for RFF and Chebyshev mechanisms, while Uniform noise is used for Linear mechanisms to ensure identifiability.

This yields six primary test settings: RFF_ER_G, RFF_SF_G, Linear_ER_U, Linear_SF_U, Chebyshev_ER_G, and Chebyshev_SF_G.

Real-world data: We use the Sachs dataset (Sachs et al., 2005), a well-established benchmark in causal discovery. It contains 853 measurements of 11 proteins and a consensus causal graph derived from biological knowledge.

Pseudo-real data: We also incorporate pseudo-real datasets generated by the SynTReN generator (Van den Bulcke et al., 2006). This generator is specifically designed to simulate synthetic transcriptional regulatory networks with biologically plausible structures and parameters, producing gene expression data that closely resembles experimental microarray data.

Model architecture: We mainly use the AVICI as the model backbone (Lorch et al., 2022), a DNN-based architecture which is currently widely followed by the community and open source. Results with other backbones are consistent and shown in Appendix B.

The training data is set up as follows:

- **i.i.d**: The training data and test data are exactly the same distribution.
- **Graph/Noise/Mechanism shift**: The mechanism/graph/noise of the training data is different from that of the test data, but the other two distributions are the same. Specifically, each specific training data setting is indicated above the results.
- **AVICI (mixed)**: The training data is a mixture of RFF_U_ER, RFF_U_SF, Linear_G_ER, Linear_G_SF, Chebysev_U_ER, and Chebysev_U_SF. This is to demonstrate the distributional combination problem of SCL training data.This makes the model see all components, mechanism (RFF,

Linear), graph (ER, SF), noise (G, U), but not see the specific combination in the test instance, such as RFF_G_ER.

- **AVICI (scm-v0)**: This model was trained on SCM data simulated from a large variety of graph models with up to 100 nodes, both linear and nonlinear causal mechanisms, and homogeneous and heterogeneous additive noise from Gaussian, Laplace, and Cauchy distributions. It can be considered the strongest model of open source under the SCL paradigm. (https://github.com/larslorch/avici)

3.2 Limitations of current SCL paradigms

Our experimental results validate the three issues outlined above, collectively exposing the limitations of static pre-training in SCL.

Issue 1. The results in Table 1 demonstrate that distribution shifts across all three dimensions (graph structure, causal mechanism, and noise distribution) significantly degrade SCL performance. Models struggle when the test-time graph structure ("Graph shift" compared to "iid"), causal mechanism ("Mechanism shift" compared to "iid"), or noise distribution ("Noise shift" compared to "iid") differs categorically from those seen during training. While performance drops are observed in all cases, "Mechanism shifts" emerge as particularly damaging, underscoring the profound impact of the underlying mechanism functional form on model generalization.

Issue 2. Even when trained on data containing all individual components, the model still exhibits performance drop on unseen combinations of these components, as seen when comparing "AVICI (mixed)" to "iid" in Table 1. This compositional failure indicates that SCL models memorize specific (G, f, ε) configurations rather than learning a modular understanding of causal factors.

Table 1: Fragility to categorical and compositional shifts. Each column represents a different test setting. AUROC performance reveals SCL's sensitivity to unseen graph, mechanism, and noise configurations. Results are presented as AUROC (standard deviation).

	RFF_G_ER	**RFF_G_SF**	**Linear_U_ER**	**Linear_U_SF**	**Chebysev_G_ER**	**Chebysev_G_SF**
iid	90.0 (2.7)	100.0 (0.0)	91.7 (4.3)	100.0 (0.0)	93.0 (2.9)	100.0 (0.0)
Graph shift	RFF_G_SF 81.0 (5.5)	RFF_G_ER 92.5 (2.8)	Linear_U_SF 78.8 (6.1)	Linear_U_ER 96.2 (1.9)	Chebysev_G_SF 63.4 (9.7)	Chebysev_G_ER 91.9 (5.9)
Noise shift	RFF_U_ER 85.0 (3.7)	RFF_U_SF 94.1 (1.4)	Linear_L_ER 80.7 (10.0)	Linear_L_SF 78.3 (11.2)	Chebysev_U_ER 85.8 (5.0)	Chebysev_U_SF 79.0 (13.2)
Mechanism shift	Chebysev_G_ER 73.7 (8.4)	Chebysev_G_SF 42.4 (14.0)	RFF_U_ER 78.4 (8.7)	RFF_U_SF 87.5 (5.0)	RFF_G_ER 72.3 (9.3)	RFF_G_SF 55.9 (9.4)
AVICI (mixed)	84.8 (4.7)	89.0 (2.0)	88.0 (3.8)	87.9 (5.9)	82.6 (6.5)	89.0 (9.5)

Issue 3. The results in Table 2 question the value of synthetic benchmarks by demonstrating that strong synthetic performance fails to guarantee effectiveness on real-world data. Here, we merge the dimensions of the graph and analyze more from the perspective of the mechanism. While AVICI (scm-v0) excels on synthetic data similar to its training distribution (e.g., RFF-G, 97.8), its performance collapses on the real-world Sachs dataset (62.3). In contrast, traditional methods like PC maintain consistent, albeit lower, performance across domains. This divergence reveals that SCL models overfit to the artifacts of their synthetic training environment, lacking the cross-domain consistency required for real-world applicability.

Table 2: Divergent generalization patterns. Strong synthetic performance does not guarantee effectiveness on real-world data. Results are presented as AUROC (standard deviation).

	RFF_G	Linear_U	Chebyshev_G	Sachs	Syntren
PC	61.1 (4.9)	60.9 (4.7)	59.8 (6.6)	67.1	58.1
AVICI (scm-v0)	97.8 (1.3)	75.6 (13.8)	81.7 (10.5)	62.3	65.4

In summary, the dual failure of fragility under distribution shifts and inconsistency across domains fundamentally undermines the static pre-training paradigm. These limitations are not artifacts of a specific architecture, as validated by consistent failure patterns using the SiCL backbone

(**Appendix B** (Table 5)). The results compellingly argue that robust causal discovery requires a shift from static, diversity-seeking pre-training to dynamic, test-time adaptation.

4 TEST-TIME TRAINING FOR SUPERVISED CAUSAL LEARNING

From the perspective of concentration, there remains an opportunity for SCL to overcome the limitations of static pretraining. We introduce the Test-Time Training for Supervised Causal Learning (TTT-SCL) framework, representing a paradigm shift from seeking universal diversity to generating targeted concentration, as shown in Fig 1.

Under standard conditions for ANM (Peters et al., 2014), the true graph G_{test} is identifiable from distribution of D_{test}. This implies that if the distribution of generated data D^k is closely aligned with D_{test} (i.e., $D^k \approx D_{test}$), then the candidate graph G^k is likely a close approximation of G_{test} (i.e., $G^k \approx G_{test}$). This observation reframes the challenge as a search problem: among candidate graphs, find those that yield data distributions aligned with the test data. This search formulation naturally leads to two key sub-problems:

- **Quantifying similarity.** Since exactly identical graphs yield exactly identical distributions, what metric can we use to quantify "similarity" between a candidate graph and the test graph?
- **Searching effectively.** Given the intractability of brute-force search over the DAG space, how can we design a practical search procedure to identify promising candidates?

4.1 QUANTIFYING SIMILARITY: THE ALIGNMENT OF DISTRIBUTION

A natural way to connect candidate graphs with the test data is through **Structure-Induced Mechanism (SIM)**. SIM directly operationalizes how a graph explains data: given a candidate graph G^k, we regress the corresponding mechanisms from the observed D_{test}, and then forward-sample synthetic data D^k. If the generated distribution is close to D_{test}, this indicates that G^k is a good approximation of the true graph G_{test}. In this sense, SIM provides a practical bridge from structural hypotheses to observable distributional alignment, making it possible to evaluate candidate graphs by how well they reproduce the test distribution.

This motivates the need for a metric of alignment between a candidate training graph and the test data. Such a metric, which we denote as **Alignment of Distribution (AD)**, should satisfy structure and mechanism similarity. While there are many ways to implement AD as discussed in Appendix A, in the main text we use the implementation based on likelihood:

$$AD(G^k_{train}, D_{test}) = \frac{1}{d}\sum_{i=1}^{d}\left[\log p\left(X_i \mid f_i^k\right)\right], \tag{3}$$

where f_i^k is the fitting function of X_i according $\mathbf{Pa}^k_{train}(X_i)$ based on G^k_{train} and D_{test} by SIM.

This formulation is attractive because likelihood inherently combines both structure and mechanism aspects. Changing the **graph structure** alters the conditioning set $\mathbf{Pa}^k_{train}(X_i)$, directly modifying the conditional distributions being estimated. Changing the **mechanisms** alters the functional mapping f_i^k, thereby changing the probability assigned to the observed data. As a result, the likelihood score simultaneously reflects structural correctness and mechanistic fidelity, and thus serves as a principled measure of distributional alignment between candidate training graphs and the test data.

Enforcing Causal Minimality with Sparsity Constraints. However, optimizing AD alone can lead to degenerate dense solutions that fit distributions without respecting causal minimality. To counteract this, we incorporate the principle of causal minimality by adding a sparsity penalty term based on the L_0 norm of the adjacency matrix A_G:

$$\text{Sparsity}(G) = ||A_G||_0. \tag{4}$$

The Joint Optimization Score. By combining these two components, we form a unified score function to evaluate any candidate training graph:

$$score(G) = \text{AD}(G, D_{\text{test}}) - \lambda \cdot \text{Sparsity}(G). \tag{5}$$

where λ is a hyperparameter balancing the trade-off. This score serves as the central optimization target for generating high-quality training data within the TTT-SCL framework.

4.2 TACTIC: Efficient Search in the Graph Space

Exhaustively searching the entire DAG space is intractable, and theoretical results confirm that finding the exact G_{test} is essentially impossible. Nevertheless, this does not imply that the problem is hopeless. In practice, good initializations combined with guided refinement can yield graphs that are close enough to G_{test} to support effective training. We instantiate this idea with TACTIC (Test-time Aligned Causal Training with Informed Construction), a concrete implementation of our TTT-SCL framework. TACTIC proceeds in three stages:

1. **Seed Initialization.** We start from an initial graph G_{seed}, obtained either by (i) applying a traditional causal discovery method (e.g., PC, NOTEARS) on D_{test}, or (ii) sampling a random DAG. This provides a useful prior rather than searching from scratch.
2. **Stochastic Graph Refinement.** From the seed, we iteratively propose local modifications to the graph (edge additions, deletions, or reversals) while maintaining the DAG constraint. Each candidate G_{k+1} is evaluated using the joint score function $score(G)$ as Formula (5) and accepted with probability proportional to its score. This stochastic refinement process ensures that search is efficient and directed, guided by AD and sparsity rather than random exploration.
3. **Training Data Generation.** For the final refined graph set $\{G^k_{train}\}_{k=1}^K$, we regress mechanisms via SIM, forward-sample synthetic datasets $\{D^k_{train}\}_{k=1}^K$, and assemble them into a customized training set. An SCL model is then trained on this set and applied to infer G_{test}.

By combining AD, sparsity, and practical heuristics (initialization + stochastic refinement), TACTIC realizes an efficient and directed approach to searching the graph space at test time, as shown in Fig 2.

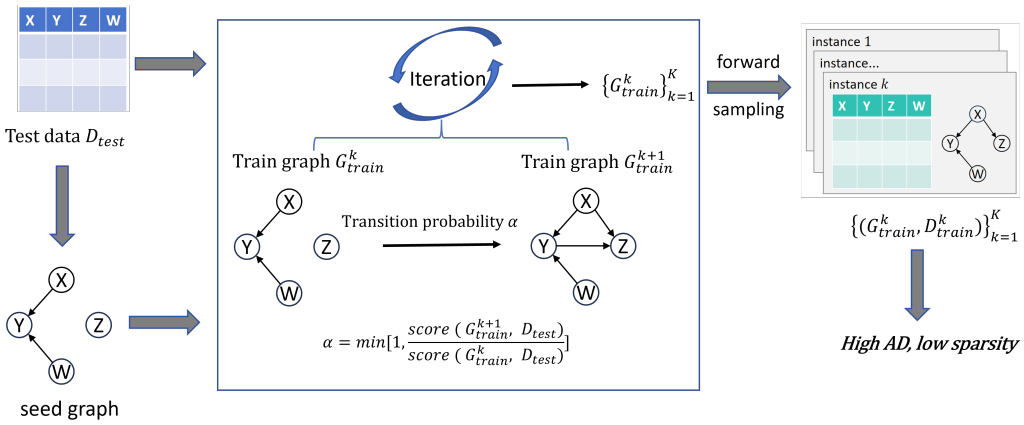

Figure 2: Workflow of TACTIC

4.3 The Performance of TACTIC

In this subsection, we compare the performance of TACTIC with multiple baseline methods on various synthetic data, pseudo-real data and real data. These datasets are consistent with the content of Section 3.1.

Baselines: We compare against traditional causal discovery methods PC (Spirtes et al., 2000), GES (Chickering, 2002), NOTEARS (Zheng et al., 2018) and AVICI (Lorch et al., 2022), a DNN-based SCL method which is currently widely followed by the community and open source. We use the open-source pre-trained AVICI (scm-v0) model, which is trained on a vast mixture of synthetic data and represents the strongest publicly available SCL baseline.

Our Method (TACTIC): For our TTT-SCL approach, we set the number of dynamically generated training graphs to $K = 200$. The number of variables d is 10 for synthetic data, 11 for Sachs and 20 for Syntren. The observation n for each generated dataset matches that of the test data. We evaluate two variants of our method: TACTIC (random) which initializes the seed graph with a random DAG, and TACTIC (Notears) which uses a graph estimated from D_{test} by the NOTEARS algorithm as a smarter starting point.

Evaluation metrics: We use multiple metrics to evaluate the predicted graphs, including Area Under the Receiver Operating Curve (AUROC), Area Under the Precision-Recall Curve (AUPRC), F1 score and Accuracy (ACC). In the main text, we primarily report **AUROC** for edge prediction to succinctly explore the impact of training data quality on model performance. Results based on other metrics are provided in Appendix C.

Table 3: TACTIC performance on synthetic, real and pseudo-real datasets. Results are presented as AUROC (standard deviation).

	RFF_G	Linear_U	Chebyshev_G	Sachs	Syntren
PC	61.1 (4.9)	60.9 (4.7)	59.8 (6.6)	67.1	58.1
GES	66.0 (10.6)	69.0 (10.8)	59.6 (5.9)	61.8	36.8
Notears	80.5 (4.0)	82.0 (4.6)	52.2 (3.5)	61.8	49.8
AVICI (scm-v0)	**97.8** (1.3)	75.6 (13.8)	81.7 (10.5)	62.3	65.4
TACTIC (random)	88.4 (7.0)	82.3 (7.0)	79.6 (6.7)	58.6	72.0
TACTIC (Notears)	91.8 (3.1)	**86.3** (4.4)	**83.0** (8.7)	**78.9**	**80.1**

The results are summarized in Table 3. Overall, TACTIC demonstrates robust and highly competitive performance. The pre-trained AVICI (scm-v0) model achieves optimal performance on the RFF_G datasets, as it was explicitly trained on this distribution. TACTIC's performance on RFF_G is slightly lower but remains strong, indicating its ability to approximate even in-distribution performance without prior exposure. Crucially, TACTIC achieves state-of-the-art performance on all other datasets, including Linear_U, Chebyshev_G, real-world Sachs, and pseudo-real Syntren dataset. This confirms that TACTIC excels in the most challenging and realistic scenarios involving distribution shifts, where static pre-training fails. Furthermore, the TACTIC (Notears) variant consistently outperforms TACTIC (random), demonstrating that a reasonable initial graph from a traditional method provides a valuable prior for the optimization. However, the strong performance of both variants confirms the robustness of our core approach. These conclusions hold consistently across multiple evaluation metrics, as demonstrated in **Appendix C** (Table 6), where TACTIC maintains superior performance in ACC, F1-score, and AUPRC under various distribution shifts.

We also design experiments to empirically validate how these two components contribute to the quality of the generated training data. We first ablate the sparsity term in the optimization objective to isolate its effect. We compare the full **TACTIC (Notears)** method against a variant, **TACTIC (Notears-s)**, where the sparsity penalty is removed ($\lambda = 0$), thus optimizing for AD alone. Results in Table 4 show that removing the sparsity term leads to a consistent and significant performance drop across all test settings. These dense graphs achieve high AD by introducing spurious edges with negligible mechanisms, but they violate the causal minimality principle and thus constitute poor-quality training data for teaching the SCL model the correct causal structure.

Table 4: Ablation experiment of sparsity. Results are presented as AUROC (standard deviation).

	RFF_G	Linear_U	Chebyshev_G	Sachs	Syntren
TACTIC (Notears)	91.8 (3.1)	86.3 (4.4)	83.0 (8.7)	78.9	80.1
TACTIC (Notears-s)	86.8 (2.9)	84.3 (7.9)	69.7 (12.4)	63.5	76.1

To further demonstrate the effectiveness of AD and the necessity of sparsity, the AD, sparsity, score of the training data obtained by different methods under different test data, as well as the AUROC on the test data were recorded in **Appendix D**. The results show that both AD and sparsity are indispensable and important elements, and they have certain indicative significance for performance.

5 Related works

Causal discovery has a long history rooted in constraint-based methods (e.g., PC, FCI (Spirtes et al., 2000)), function-based methods (e.g., LiNGAM (Shimizu et al., 2006), ANM (Hoyer et al., 2008)) and score-based methods (e.g., GES (Chickering, 2002), NOTEARS (Zheng et al., 2018), DAG-GNN (Yu et al., 2019), GraN-DAG (Lachapelle et al., 2020)). These approaches operate unsupervised and infer causal graphs directly from observational data using statistical independencies, asymmetry assumptions or various scores. While principled, they often suffer from high sample complexity, sensitivity to faithfulness violations, and limited scalability to high-dimensional settings.

Supervised Causal Learning (SCL) has recently emerged as a promising paradigm that approaches causal discovery as a supervised learning problem (Dai et al., 2023; Lorch et al., 2022; Ke et al., 2022). It trains a machine learning model to take observational data as input and output the causal graph or relations and leverage powerful models to learn mappings from data patterns to causal structures, instead of hand-crafted heuristics. The analysis of SCL can be conducted from the following three aspects:

Model architecture. Prior SCL methods employ diverse architectures to map datasets to graphs. For example, Ma et al. (2022) propose cascade classifiers that sequentially test conditional independencies by increasing the conditioning order. Dai et al. (2023) design architecture featurizes variable neighborhoods and classifies unshielded triples. Lorch et al. (2022), Ke et al. (2022), and Froehlich & Koeppl (2024) use the attention-based transformer that treats the data as a 3D tensor (observations $\times$ variables $\times$ features) and alternates self-attention over samples and variables. In addition, Zhang et al. (2025) propose pairwise attention to capture the node features and node-pair features.

Target output representation. SCL methods target different representations of causal relationships. Some methods learn only the undirected skeleton of the graph, e.g. Ma et al. (2022) aims to recover the full skeleton. Others focus on orienting local structures: for instance, Dai et al. (2023) takes as input the graph skeleton and classifies each unshielded triple as a v-structure or not, then orients edges accordingly. Ke et al. (2022)'s transformer outputs a full directed adjacency matrix via an autoregressive decoder over all node pairs, and Lorch et al. (2022)'s network similarly predicts edge probabilities between every ordered pair. Many methods only guarantee recovery up to Markov equivalence: for example, Zhang et al. (2025) train a model to output the skeleton and v-structure and Froehlich & Koeppl (2024) learns the moralized graphs.

Training data strategy and test time training. Most SCL approaches are pre-trained on large static datasets of synthetic causal models. These methods often suffer from different degree of generalization failures. There are very less methods try to solve this problem from the perspective of training data. In the field of machine learning, there have been considerable studies that use information from test data to design or generate training data (Liang et al., 2025; Sun et al., 2020; Wang et al., 2020; Liu et al., 2021; Sinha et al., 2023).

6 Conclusion

In this work, we identified fundamental limitations of static SCL paradigms, demonstrating their fragility under distribution shifts, failure in compositional generalization, and poor transfer from synthetic benchmarks to real-world data. To address these out-of-distribution generalization challenges, we introduced TTT-SCL, a paradigm-shifting framework that addresses the out-of-distribution generalization problem in supervised causal learning through test-time training of causally-aligned data. Our proposed AD metric, combined with sparsity constraints, provides a tractable and effective way to ensure causal similarity between training and test data. The TACTIC method, as an instantiation of TTT-SCL, dynamically generates high-quality training data tailored to each test instance, achieving good performance on both synthetic, pseudo-real and real-world datasets. Our theoretical and empirical results underscore the effectiveness of AD and necessity of sparsity. This work not only advances the field of supervised causal learning but also opens new avenues for robust and adaptive causal discovery in real-world settings.

REFERENCES

Albert-László Barabási. Scale-free networks: a decade and beyond. science, 325(5939):412–413, 2009.

David Maxwell Chickering. Optimal structure identification with greedy search. Journal of machine learning research, 3(Nov):507–554, 2002.

Haoyue Dai, Rui Ding, Yuanyuan Jiang, Shi Han, and Dongmei Zhang. Ml4c: Seeing causality through latent vicinity. In Proceedings of the 2023 SIAM International Conference on Data Mining (SDM), pp. 226–234. SIAM, 2023.

Philipp Froehlich and Heinz Koeppl. Graph structure inference with bam: Neural dependency processing via bilinear attention. Advances in Neural Information Processing Systems, 37:128847–128885, 2024.

Edgar N Gilbert. Random graphs. The Annals of Mathematical Statistics, 30(4):1141–1144, 1959.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. Advances in neural information processing systems, 21, 2008.

Nan Rosemary Ke, Silvia Chiappa, Jane Wang, Anirudh Goyal, Jorg Bornschein, Melanie Rey, Theophane Weber, Matthew Botvinic, Michael Mozer, and Danilo Jimenez Rezende. Learning to induce causal structure. arXiv preprint arXiv:2204.04875, 2022.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=rklbKA4YDS.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. International Journal of Computer Vision, 133(1):31–64, 2025.

Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? Advances in Neural Information Processing Systems, 34:21808–21820, 2021.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized inference for causal structure learning. Advances in Neural Information Processing Systems, 35: 13104–13118, 2022.

Pingchuan Ma, Rui Ding, Haoyue Dai, Yuanyuan Jiang, Shuai Wang, Shi Han, and Dongmei Zhang. Ml4s: Learning causal skeleton from vicinal graphs. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1213–1223, 2022.

Judea Pearl. Causality. Cambridge university press, 2009.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. The Journal of Machine Learning Research, 15(1):2009–2053, 2014.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Elements of causal inference: foundations and learning algorithms. The MIT press, 2017.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. Advances in neural information processing systems, 20, 2007.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. Science, 308(5721): 523–529, 2005. doi: 10.1126/science.1105809. URL https://www.science.org/doi/abs/10.1126/science.1105809.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. Journal of Machine Learning Research, 7(10), 2006.

Samarth Sinha, Peter Gehler, Francesco Locatello, and Bernt Schiele. Test: Test-time self-training under distribution shift. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2759–2769, 2023.

Peter Spirtes, Clark N Glymour, and Richard Scheines. Causation, prediction, and search. MIT press, 2000.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In International conference on machine learning, pp. 9229–9248. PMLR, 2020.

Tim Van den Bulcke, Koenraad Van Leemput, Bart Naudts, Piet van Remortel, Hongwu Ma, Alain Verschoren, Bart De Moor, and Kathleen Marchal. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. BMC bioinformatics, 7(1):43, 2006.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. arXiv preprint arXiv:2006.10726, 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.

Jiaru Zhang, Rui Ding, Qiang Fu, Bojun Huang, Zizhen Deng, Yang Hua, Haibing Guan, Shi Han, and Dongmei Zhang. Learning identifiable structures helps avoid bias in dnn-based supervised causal learning. arXiv preprint arXiv:2502.10883, 2025.

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.

A Implementation of AD

In the main text, we propose the Alignment of Distribution (AD) metric as a core measure of causal similarity between the generated training data D_{train} and the test instance D_{test}. While the likelihood-based implementation was used in our primary experiments, we provide alternative formulations here to accommodate different data distributions and modeling assumptions.

A.1 R^2-based AD

For continuous variables under additive noise models, the coefficient of determination (R^2) provides an intuitive measure of goodness-of-fit for each causal mechanism:

$$AD_{R^2}(G_{train}, D_{test}) = \frac{1}{d}\sum_{i=1}^{d}\left[\frac{1}{K}\sum_{k=1}^{K} R^2\left(f_i^k(\mathbf{Pa}^k(X_i)), X_i\right)\right]$$

This value approaches 1 when the fitted mechanisms explain the variance in D_{test} well, indicating strong alignment.

A.2 Normalized Wasserstein Distance-based AD

For multi-modal or heavy-tailed distributions, the Wasserstein distance offers a robust metric for comparing empirical distributions. We define a *Normalized Wasserstein Distance (NWD)* based AD metric as follows:

For a given variable X_i and a candidate graph G^k with its fitted mechanism f_i^k, we compute:

$$\text{NWD}(f_i^k, G^k, D^{test}) := 1 - \frac{W_1\left(\{x_i\},\ \{f_i^k(\mathbf{Pa}^k(X_i))\}\right)}{\max(\mathcal{U}) - \min(\mathcal{U})}$$

where:

- $\{x_i\}$ are the observed values of X_i in D_{test}.
- $\{f_i^k(\mathbf{Pa}^k(X_i))\}$ are the values generated by applying the fitted mechanism f_i^k to the parent values in D_{test}.
- W_1 is the 1-Wasserstein distance (Earth Mover's Distance). For two equally sized, sorted collections of values $\{a^{(j)}\}$ and $\{b^{(j)}\}$, it is defined as:

$$W_1(\{a\},\{b\}) = \frac{1}{n}\sum_{j=1}^{n}|a^{(j)} - b^{(j)}|$$

- $\mathcal{U} = \{x_i\} \cup \{f_i^k(\mathbf{Pa}^k(X_i))\}$ is the union of the observed and generated values for X_i.
- The denominator, $\max(\mathcal{U}) - \min(\mathcal{U})$, is the range of the combined set, used for normalization.

The resulting NWD value lies between 0 and 1, where 1 indicates a perfect match between the generated and observed distributions for that variable. The overall AD metric is then the average NWD across all variables and generated graphs:

$$AD_{\text{NWD}}(G_{train}, D_{test}) = \frac{1}{K}\sum_{k=1}^{K}\left[\frac{1}{d}\sum_{i=1}^{d}\text{NWD}(f_i^k, G^k, D^{test})\right]$$

A.3 Selection Guidance

The **likelihood-based** AD is most natural for probabilistic models and was used in our main experiments. The **R^2-based** AD is suitable for continuous variables under additive noise assumptions, often leading to computationally efficient and intuitive scores. The **NWD-based** AD is recommended for complex, non-Gaussian, or heavy-tailed distributions where likelihood or R^2 might be less informative or robust. The TTT-SCL framework is agnostic to the specific choice of AD metric, allowing users to select the most appropriate one for their domain.

B Consistency on other model backbones

To further validate the generality of the TTT-SCL framework and the observed o.o.d generalization challenges across different model architectures, we conduct experiments using the Pairwise Attention from Zhang et al. (2025) (**SiCL**) as an alternative model backbone. Unlike the AVICI transformer used in the main experiments, which predicts a full directed adjacency matrix (DAG), SiCL incorporates pairwise attention mechanisms and is trained to predict the undirected skeleton and v-structures of the causal graph. This setup allows us to investigate whether the identified o.o.d failure patterns persist when using a fundamentally different architecture (with pairwise attention) and a different learning target (skeleton and v-structures instead of a full DAG), thereby testing the robustness of our conclusions.

B.1 Experimental Setup

Backbone Model is SiCL (Pairwise Attention Network) Zhang et al. (2025). Learning Target is Undirected graph skeleton. The training strategy for the static baseline models (i.i.d. and SiCL(mixed)) follows the same data generation procedures described in Section 5.1.1 of the main text, but the ground-truth labels are converted to the appropriate representation for SiCL (skeleton labels). Evaluation Metric is AUROC for edge presence in the predicted skeleton. OOD Settings is identical to those defined for Table 1 in the main text: *i.i.d.*, *Graph shift*, *Noise shift*, *Mechanism shift*. The *AVICI(mixed)* is replaced with *SiCL(mixed)*, respectively.

B.2 Results and Analysis

Table 5 presents the AUROC for skeleton discovery under different distribution shifts.Consistent with the findings in Table 1 using the AVICI backbone, the SiCL backbone—which employs a fundamentally different pairwise attention architecture and learns undirected skeletons rather than

full DAGs—exhibits the same pattern of out-of-distribution generalization failure. Under i.i.d. conditions, SiCL achieves perfect or near-perfect performance. However, significant performance degradation occurs across all types of distribution shifts, with mechanism shifts proving particularly damaging (e.g., dropping to 66.5 on RFF_G_SF and 58.4 on Chebyshev_G_SF). Critically, the SiCL(mixed) variant, while trained on data containing all individual distributional components (graph types, mechanisms, and noise distributions), still fails to generalize to novel combinations of these factors. This demonstrates that SCL models struggle with compositional generalization—they memorize specific configuration patterns rather than learning modular causal representations. These results demonstrate that the OOD generalization challenge is not specific to a particular model architecture or output representation, but represents a fundamental limitation of the static pre-training paradigm in supervised causal learning. The consistent failure patterns across both transformer-based (AVICI) and pairwise-attention-based (SiCL) models strongly validate the need for test-time adaptation frameworks like TTT-SCL.

Table 5: OOD generalization performance for skeleton using the SiCL (Pairwise Attention) backbone.

	RFF_G_ER	RFF_G_SF	Linear_U_ER	Linear_U_SF	Chebysev_G_ER	Chebysev_G_SF
iid	82.1(6.7)	100.0(0.0)	81.4(6.9)	100.0(0.0)	94.3(2.8)	100.0(0.0)
Graph shift	66.4(9.0)	85.4(4.1)	65.8(6.9)	94.0(2.5)	73.0(5.7)	92.9(4.3)
Noise shift	60.0(8.9)	91.7(3.8)	65.3(7.4)	84.0(7.7)	88.6(5.3)	89.3(5.4)
Mechanism shift	62.1(7.4)	66.5(6.3)	59.4(4.7)	83.8(4.8)	76.1(8.9)	58.4(9.7)
SiCL(mixed)	64.4(8.0)	74.4(10.7)	66.7(7.3)	82.7(8.2)	85.6(3.7)	91.2(4.1)

C PERFORMANCE IN OTHER METRICS

In the main text, we primarily reported the AUROC for edge prediction to succinctly demonstrate the impact of training data quality on model performance. For a more comprehensive evaluation, we provide results on additional standard causal discovery metrics in this appendix:

- **Accuracy (ACC)**: The proportion of correctly predicted edge presence/absence across all possible edges. Higher is better. This metric can be viewed as a normalized version of the Structural Hamming Distance (SHD), where instead of counting the number of incorrect edges, it measures the proportion of correct edge predictions relative to the total possible edges.
- **F1-Score**: The harmonic mean of precision and recall for edge prediction. Higher is better.
- **Area Under the Precision-Recall Curve (AUPRC)**: Particularly informative under class imbalance (sparse graphs). Higher is better.

Table 6: Comprehensive evaluation across multiple datasets and metrics. Mean (standard deviation) over multiple runs are reported for synthetic data. Best results are in **bold**.

Method	RFF_G			Linear_U			Chebyshev_G			Sachs			Syntren		
	ACC↑	F1↑	AUPRC↑	ACC↑	F1↑	AUPRC↑	ACC↑	F1↑	AUPRC↑	ACC↑	F1↑	AUPRC↑	ACC↑	F1↑	AUPRC↑
PC	75.6(4.2)	39.5(9.7)	37.5(6.3)	74.0(4.4)	40.4(8.4)	37.2(6.2)	73.7(5.1)	37.4(12.6)	36.8(7.8)	84.2	45.7	30.1	84.75	16.43	6.89
GES	76.7(7.7)	49.8(16.3)	43.5(12.3)	71.9(9.8)	55.4(12.9)	45.3(9.3)	72.1(5.1)	38.5(10.5)	35.7(6.6)	82.6	36.3	24.2	65.50	1.42	4.79
NOTEARS	86.6(3.3)	73.2(6.2)	64.1(7.4)	89.1(2.9)	76.6(7.3)	69.7(8.6)	72.3(2.7)	14.5(8.7)	29.8(3.3)	82.6	36.3	24.2	**94.75**	0.00	5.00
AVICI(scm-v0)	**93.1(1.6)**	**87.3(3.4)**	**94.9(3.1)**	73.9(7.1)	41.8(18.4)	52.8(17.0)	80.6(5.4)	58.4(14.5)	69.3(14.2)	83.4	23.0	31.6	93.00	22.22	25.53
TACTIC (random)	83.2(5.3)	72.8(9.4)	68.8(10.8)	75.9(6.1)	59.8(11.8)	56.2(11.5)	75.5(7.0)	56.6(10.8)	60.0(10.0)	68.5	24.0	24.5	72.50	16.66	**53.91**
TACTIC (Notears)	86.8(3.5)	78.4(6.1)	76.0(8.6)	**78.7(3.9)**	**65.4(8.0)**	**65.0(9.9)**	**77.1(6.7)**	**61.9(10.2)**	**66.0(16.3)**	**85.9**	**56.4**	**53.6**	90.50	**32.14**	51.85

Table 6 presents the performance of all compared methods across three distinct synthetic data settings (RFF_G, Linear_U, and Chebyshev_G) and the real-world Sachs dataset. TACTIC (Notears) achieves highly competitive performance across all datasets and evaluation metrics (ACC, F1, AUPRC), demonstrating its robustness to distribution shifts. It consistently outperforms traditional methods (PC, GES, NOTEARS) and the strong pre-trained SCL baseline AVICI(scm-v0) on most settings, particularly on the challenging Chebyshev_G and real-world Sachs dataset. While AVICI(scm-v0) excels in the RFF_G setting it was trained on, its performance degrades significantly under mechanism shifts (Linear_U) and on real data, highlighting the limitation of static pre-training. The superior performance of TACTIC across multiple metrics confirms that its test-time training

strategy generates high-quality, causally-aligned training data, leading to more accurate and reliable causal discovery.

D MORE EXPERIMENTS ABOUT AD AND SPARSITY

The main text established the necessity of the sparsity constraint in the TACTIC optimization objective to prevent degenerate, overly dense solutions. This appendix provides further empirical evidence to dissect the roles of the AD metric and the sparsity constraint.

D.1 THE ROLE OF AD AND SPARSITY

To further demonstrate the effectiveness of AD and the necessity of sparsity, the AD, sparsity, score of the training data obtained by different methods under different test data, as well as the AUROC on the test data were recorded in Fig 3. The combined optimization of AD and sparsity is critical for generating high-quality training data. Without sparsity constraints (TACTIC(Notears-s)), high AD values alone lead to overly dense graphs that overfit the test distribution, violating causal minimality and resulting in lower AUROC. In contrast, jointly optimizing AD and sparsity (TACTIC(Notears)) yields training data that is both distributionally aligned and structurally sparse, closely matching the true causal graph. The resulting composite score strongly correlates with final model AUROC, confirming that both components are essential for robust generalization under distribution shifts, especially mechanism shifts.

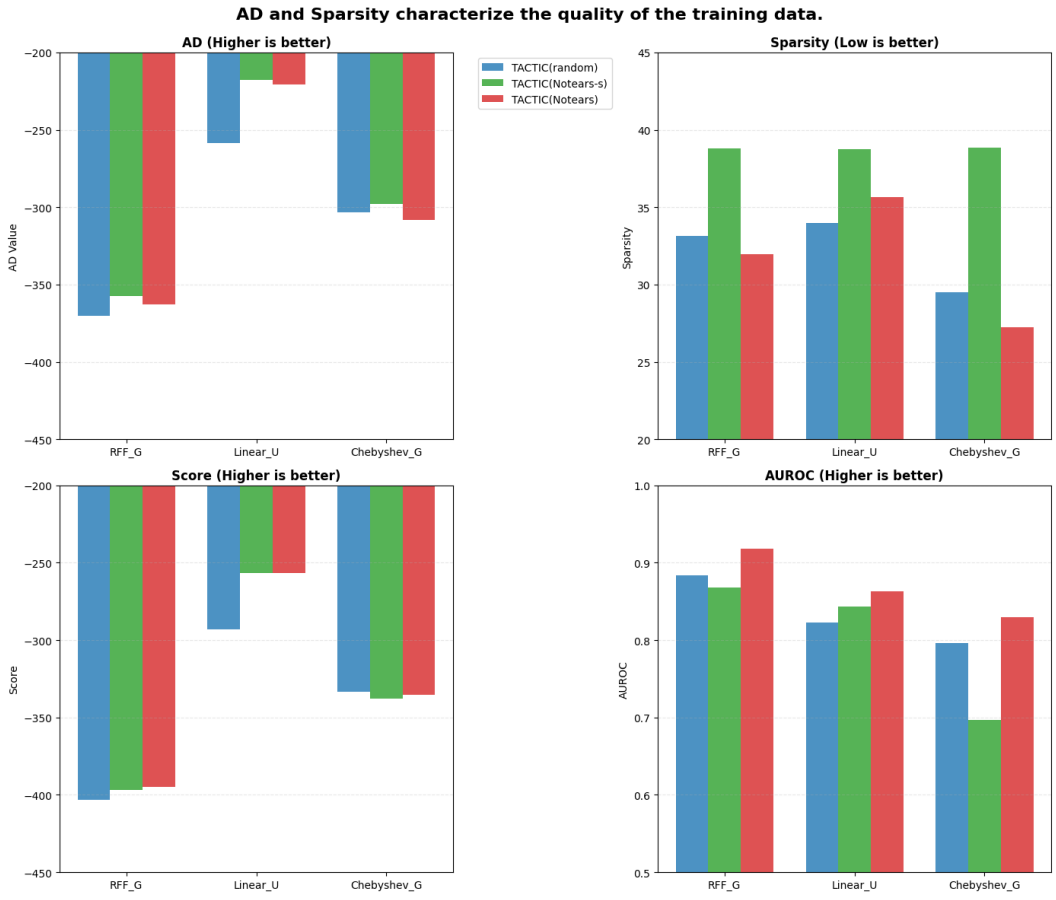

Figure 3: Combined score indicate training data quality. Training data quality is determined not by AD or sparsity alone, but by their combined score.

D.2 CONTROL AD, CHANGE SPARSITY

To control sparsity independent of AD, we design a controlled experiment based on the ground-truth test graph G_{test}. For a given G_{test} and its observational data D_{test}, we generate alternative candidate training graphs G_{train} by **gradually adding extra edges** to G_{test} (while ensuring the resulting graph remains a DAG). This creates a series of graphs that are supergraphs of the true graph.

- **Setting 1 (Sparse)**: Add a small number of extra edges ($|E_{add}| = m_1$).
- **Setting 2 (Medium)**: Add a medium number of extra edges ($|E_{add}| = m_2, m_2 > m_1$).
- **Setting 3 (Dense)**: Add a large number of extra edges ($|E_{add}| = m_3, m_3 > m_2$).

For each generated supergraph G_{train} in these settings, we then: 1. Parameter Fitting: Regress the mechanisms f_i and noise distributions from D_{test} using G_{train} (via SIM). 2. Forward Sampling: Generate synthetic training data D_{train} from the fitted SCM (G_{train}, f, ϵ). 3. Calculate Metrics: Compute the AD score between D_{train} and D_{test}, and the sparsity of G_{train}. 4. Train & Evaluate: For each (G_{train}, D_{train}) pair, train an SCL model (AVICI backbone) and evaluate its AUROC on recovering the *true* G_{test} from D_{test}.

This procedure is repeated for K graphs per setting. The key insight is that by construction, all generated G_{train} graphs are capable of representing the data distribution D_{test}. Therefore, we expect them to achieve similar, high AD scores. However, only the sparsest graph (G_{test} itself) represents the true causal structure.

Table 7 shows the results for the **RFF_ER_G** dataset, which are representative of the overall trend.

Table 7: Control AD, change sparsity

RFF_ER_G	setting	AD	sparsity	AUROC
Control AD, change sparsity	1	-375	25.91	1.0(0)
	2	-368 (+1.8%)	32.32(+24.7%)	0.972(0.017)
	3	-362(+3.4%)	36.59(+41.2%)	0.908(0.023)

The results clearly demonstrate the critical, independent role of the sparsity constraint. All supergraphs achieve a high and similar AD score (variation $< 4\%$), confirming that many different graphs can explain the observed data distribution nearly equally well. This illustrates the identifiability crisis without further constraints. As expected, adding more edges increases the sparsity metric (number of edges). Crucially, the downstream performance (AUROC) of the SCL model **degrades significantly as the graphs become denser**, even though the AD score remains high. The model trained on the true graph (Setting 1, perfect sparsity) achieves perfect AUROC. Performance drops to 0.972 for medium density and further to 0.908 for high density.