Stabilizing Causal Structure Learning under Heteroscedasticity: Analysis and Mitigation of Optimization Failures

Eunjung Choi*
Dept. of Artificial Intelligence
Hanyang University
Seoul, South Korea
sinegi@hanyang.ac.kr

Seonggyeom Kim*
Dept. of Artificial Intelligence
Hanyang University
Seoul, South Korea
canino@hanyang.ac.kr

Dong-Kyu Chae
Dept. of Computer Science
Hanyang University
Seoul, South Korea
dongkyu@hanyang.ac.kr

Abstract

This study focuses on learning causal *directed acyclic graphs* (DAGs) under *heteroscedastic noise models* (HNMs), where each effect is modeled as a function of its causes and a Gaussian noise term whose variance depends on the causes. While HNMs theoretically guarantee identifiability of causal structures, we show that gradient-based continuous DAG learning can fail in practice due to an adverse interaction between heteroscedastic likelihood optimization and the acyclicity constraint. Specifically, because the reconstruction gradient is scaled by the predicted variance, it can be heavily attenuated in early training; as a result, the DAG parameters may be updated primarily by the acyclicity constraint before the data reconstruction signal is sufficiently learned, hindering effective structure learning. We identify and formalize this failure mode. To mitigate it, we propose a graduated optimization strategy based on a surrogate loss that decouples the variance term from the reconstruction loss, thereby preventing early gradient attenuation. We further introduce a scheduling coefficient that initially assigns a high weight to the surrogate loss for stable mean learning, and then gradually transitions to the full heteroscedastic likelihood to refine variance estimates and strictly enforce acyclicity. This strategy avoids the identified failure mode and guides the learned DAG to better reflect the data. Experimental results on both synthetic and real-world datasets verify the effectiveness of our approach. Our Github repository including code and supplementary material is here: https://github.com/Sinegi/HNM/.

CCS Concepts

• **Mathematics of computing → Causal networks**; • **Theory of computation → Continuous optimization**.

Keywords

DAG learning, Heteroscedastic noise models, Optimization failure modes, Graduated optimization, Loss weight schedule

*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
KDD '26, Jeju Island, Republic of Korea
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3818156

ACM Reference Format:
Eunjung Choi, Seonggyeom Kim, and Dong-Kyu Chae. 2026. Stabilizing Causal Structure Learning under Heteroscedasticity: Analysis and Mitigation of Optimization Failures. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3770855.3818156

1 Introduction

Learning a *Directed Acyclic Graph* (DAG) from observational data can yield important insights in fields such as biology [20], healthcare [35], and economics [18]. Structural Equation Models (SEMs) [21] provide a framework for causal discovery under certain conditions, generally modeling each variable as a function of its direct parents and an exogenous noise term. Recently, Zheng et al. [36] have proposed to learn DAGs by parameterizing SEMs and enforcing acyclicity through a smooth differentiable constraint.

Subsequently, several methods [10, 34, 37] have adopted and extended this continuous optimization framework. These approaches are typically instantiated as SEMs with additive noise. For each variable, the associated noise is modeled as independent and identically distributed across observations with constant variance, which corresponds to homoscedastic noise. Meanwhile, Mooij et al. [17] and Immer et al. [11] report that, under this (homoscedastic noise) assumption, models exhibit limitations in terms of direction identifiability and model fit when the noise variance can vary with the input (direct parents), and they address these limitations by explicitly modeling the noise variance as a function of the parents using *Heteroscedastic Noise Models* (HNMs).

Several prior studies have explored causal discovery under heteroscedasticity [11, 12, 29, 32], but mostly in simplified scenarios such as bivariate causal direction. To generalize to multivariate data, prior works had to first learn an undirected graph structure (e.g., using PC [27]) and then apply those heteroscedastic techniques to orient the edges. The authors of HOST [7] and ICDH [33] both establish identifiability of HNMs in a general multivariate setting. HOST achieves this by first estimating a causal ordering of the variables, and then reconstructing the DAG via conditional independence tests. In contrast, ICDH performs gradient-based continuous optimization with a differentiable acyclicity constraint [36] to directly recover the DAG, while modeling the noise distribution with neural networks. More recently, SkewScore [16] proposes an ordering-based approach that leverages the skewness of conditional distributions to identify causal directions under HNMs.

Despite this progress, learning DAGs with HNMs presents an optimization challenge because both the mean and variance functions must be learned jointly. When training with a negative log-likelihood (NLL) loss, the gradient is often dominated by the variance term, to the detriment of accurate mean estimation. This issue is well documented in heteroscedastic regression [24, 26, 28], and various methods aim to mitigate it by decoupling the variance term's influence on the mean. Consequently, HNM-based DAG learning inherits these same pitfalls of NLL optimization. For example, Yin et al. [33] improved training stability by updating the mean and variance networks separately in an alternating fashion, similar to the work of Skafte et al. [26].

We identify a new challenge specific to structure learning with HNMs, which arises from the NLL optimization issue discussed above. In DAG learning under HNMs, the training objective consists of a reconstruction loss (likelihood) and an acyclicity constraint, which together encourage the learning of a DAG that accurately reflects the data. Under this objective, both components are expected to sufficiently contribute to the parameter updates. However, the consequences of the NLL optimization issue for DAG learning, where the reconstruction term and the acyclicity constraint coexist, have not yet been explored. We observe that, as the gradient of the reconstruction loss becomes heavily attenuated, the acyclicity constraint starts to dominate the parameter updates even from the early stages of training. When the acyclicity constraint dominates the updates from the beginning, the structure parameters are updated mainly in the direction that satisfies the acyclicity constraint, without sufficiently incorporating the data reconstruction signal, which hinders effective structure learning. This paper analyzes and formalizes this failure mode (see Figure 1).

To address this issue, we propose a graduated optimization strategy based on weighted loss scheduling to (i) start from a surrogate loss and smoothly transition to the standard NLL, and (ii) gradually activate the acyclicity constraint. In early training, we set the scheduling coefficient high initially to the surrogate loss that decomposes the standard NLL into an easier-to-optimize form. This, in turn, ensures that the updates of the DAG parameters are primarily driven by the data reconstruction signal. As training progresses, we gradually decrease the scheduling coefficient, so that the surrogate loss smoothly transitions to the NLL loss and the acyclicity constraint increasingly enforces an acyclic structure. In that it starts from a simple surrogate loss and then transitions to the standard NLL, our method is in the spirit of homotopy-based DAG learning [5], but extends it from the simple linear and bivariate settings to a nonlinear multivariate HNM regime, which is more realistic and challenging. In this case, global optimality is no longer guaranteed, but we formalize sample-dependent conditions where the proposed optimization strategy is sufficient to improve DAG recovery, and verify them empirically. We also show that our method induces a much more stable optimization trajectory than standard NLL training, and empirically outperforms existing baselines in terms of structure learning performance on both synthetic and real datasets.

2 Preliminaries

Throughout the paper, several additional details and analyses are provided in our **supplementary material** (**link is here**: https://github.com/Sinegi/HNM/blob/main/Supplementary_Material.pdf). Its overall structure is explained in Appendix.

2.1 Background

Let $X = [X_1, \ldots, X_d]$ be a d-dimensional random vector, and let $x^{(n)} = [x_1^{(n)}, \ldots, x_d^{(n)}] \in \mathbb{R}^d$ denote the n-th sample of X. Given observed data $\mathbf{X} \in \mathbb{R}^{N \times d}$ with N samples, the underlying causal structure can be represented as a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, d\}$ is a set of nodes corresponding to d variables, and $\mathcal{E}$ is the set of directed edges with no cycles. For each node $j \in \mathcal{V}$, we define the parent set of j, $pa(j) = \{i \in \mathcal{V} \mid (i \to j) \in \mathcal{E}\}$, so that $X_{pa(j)}$ denotes parent variables and direct causes of X_j. Such relationships among variables can be encoded in a weighted matrix $W \in \mathbb{R}^{d \times d}$. We assume the data are generated by a structural equation model (SEM) with the following *heteroscedastic* additive noise [7]:

Definition 1 (Heteroscedastic noise model). *A data vector X follows Heteroscedastic Noise Models (HNMs) if, for each node $j \in \mathcal{V}$,*

$$X_j = \mu_j(X_{pa(j)}) + \sigma_j(X_{pa(j)}) \cdot E_j, \quad \text{where } (E_1, \ldots, E_d)^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d). \tag{1}$$

Here, $\mu_j : \mathbb{R}^{|pa(j)|} \to \mathbb{R}$ is a nonlinear deterministic mean function, and the scale function $\sigma_j : \mathbb{R}^{|pa(j)|} \to \mathbb{R}_{>0}$ adjusts the variance of the exogenous noise E_j depending on the values of $X_{pa(j)}$. That is, we assume that $E_1, \ldots, E_d$ are mutually independent standard Gaussians, but their variances are scaled differently for each variable and observation by $\sigma_j(X_{pa(j)})^2$.

Theorem 2.1 (Identifiability of HNMs (Yin et al. [33], restated)). *Under HNMs, if $\mu(\cdot)$ is a nonlinear function, $\sigma(\cdot)$ is a piecewise function, and noise variables E are independent and Gaussian, then the true causal graph W^* is identifiable from the joint distribution of the observed variables P_X. Its proof is provided in Section E of our supplementary material.*

Definition 2 (Global solution equivalence class). *Let P_{X^*} be the true data distribution generated by the ground-truth DAG W^*. The global solution equivalence class [6] $\mathcal{S}(P_{X^*})$ is the set of all model parameters (Θ, W) that perfectly reproduce this distribution, thereby achieving the minimum possible population risk.*

Under the conditions of Theorem 2.1, for any parameter set $(\Theta, W) \in \mathcal{S}(P_{X^*})$, the graph structure W must be the true DAG W^*. This establishes the existence of a unique global solution, providing a clear theoretical target for our proposed method. Although W^* may exist, it is generally difficult to find it when dealing with finite samples. We analyze how the algorithm actually behaves on finite samples under HNMs and focus on this issue.

Under HNM, the conditional distribution of X_j given $X_{pa(j)}$ follows $p(X_j|X_{pa(j)}) \sim \mathcal{N}(\mu_j(X_{pa(j)}), \sigma_j^2(X_{pa(j)}))$ due to $\mathbb{E}[E_j|X_{pa(j)}] = 0$ and $Var[E_j|X_{pa(j)}] = 1$. Based on this conditional distribution, the Gaussian negative log-likelihood (NLL) for the entire data (over all d variables) can be defined as the reconstruction loss with the mean function μ_j and the variance function σ_j parameterized by

$\Theta = \{\theta_\mu, \theta_\sigma\}$:

$$\mathcal{L}_{\text{NLL}}(\Theta, W) = \sum_{j=1}^{d} \left[\frac{1}{2} \log \sigma_j^2(X; W_j, \theta_\sigma) + \frac{\left(X_j - \mu_j(X; W_j, \theta_\mu)\right)^2}{2\sigma_j^2(X; W_j, \theta_\sigma)} \right], \tag{2}$$

where $W_j \in \mathbb{R}^d$ represents the directed connectivity from $X_{pa(j)}$ to X_j. Since the matrix W must represent a DAG, the smooth acyclicity constraint $h(W) = \text{tr}(e^{W \circ W}) - d$, where $W \circ W$ denotes the Hadamard (element-wise) product of W, should be imposed [36]; enforcing $h(W) = 0$ yields the form of an equality-constrained optimization, and this can be solved via the Augmented Lagrangian Method (ALM) [2, 36]:

$$\min_{W,\Theta} \ \mathcal{L}_{\text{NLL}}(\Theta, W) + \frac{\rho}{2} h(W)^2 + \alpha h(W), \tag{3}$$

where $\rho > 0$ is the penalty coefficient and α is the Lagrange multiplier. As $h(W) \to 0$, the learned W becomes an acyclic structure. The parameterization and use of W in our work follow Zheng et al. [37] and Yin et al. [33], with further implementation details provided in Section G.4 of our supplementary material.

2.2 Motivation

2.2.1 Overestimated predictive variance due to joint optimization. Skafte et al. [26] and Seitzer et al. [24] showed that directly optimizing a heteroscedastic model's NLL loss to jointly learn both the mean and variance functions with a neural network can cause the predicted variance to dominate the gradients, hindering accurate mean fitting. This can be understood by examining the gradients of the NLL loss with respect to the mean and variance:

$$\begin{aligned} \nabla_\mu \mathcal{L}_{\text{NLL}} &= \frac{X_j - \mu_j(X; W_j, \theta_\mu)}{\sigma_j^2(X; W_j, \theta_\sigma)}, \\ \nabla_\sigma \mathcal{L}_{\text{NLL}} &= \frac{\sigma_j^2(X; W_j, \theta_\sigma) - (X_j - \mu_j(X; W_j, \theta_\mu))^2}{2\sigma_j^4(X; W_j, \theta_\sigma)}. \end{aligned} \tag{4}$$

Because the gradient $\nabla_\mu L_{\text{NLL}}$ is inversely proportional to the predictive variance, samples for which the model predicts a smaller variance have a disproportionately larger impact on the parameter updates. Consequently, instead of improving its accuracy on hard-to-predict regions, the model tends to reduce overall loss by increasing the predicted variance in regions that are already well-fit or inflating the variance for more difficult samples [24, 30]. To address this issue, several techniques have been proposed in the heteroscedastic learning literature. Skafte et al. [26] introduced an alternating optimization scheme for the two (mean and variance) networks to effectively decouple the influence of variance estimation on the mean's learning. Seitzer et al. [24] proposed a surrogate loss called β-NLL, which weights each data point's contribution by the β-th power of its predicted variance in order to reduce the effect of the variance term. Stirn et al. [28] advocated for a decoupled architecture in which an encoder first produces a shared latent representation of the input, and then two separate networks use this representation to predict the mean and the covariance matrix—thereby isolating the mean estimation from direct variance interference.

2.2.2 New challenge in DAG learning under HNM. However, in the context of DAG learning under HNM, a new challenge arises when using NLL as the reconstruction loss. We focus on the balance between the NLL-based reconstruction term and the acyclicity constraint $h(W)$ that enforces a DAG structure. As Eq. (4) indicates that the predictive variance acts like an inverse weight on the gradient of the reconstruction loss with respect to the mean function—samples with higher predicted variance contribute much less to the mean updates. Consequently, during the initial phase of training, the influence of the NLL reconstruction loss on the update of W is relatively weakened, and the acyclicity constraint $h(W)$ can come to dominate the learning. In other words, the learned structure W ends up reflecting the acyclicity constraint more than the data reconstruction signal.

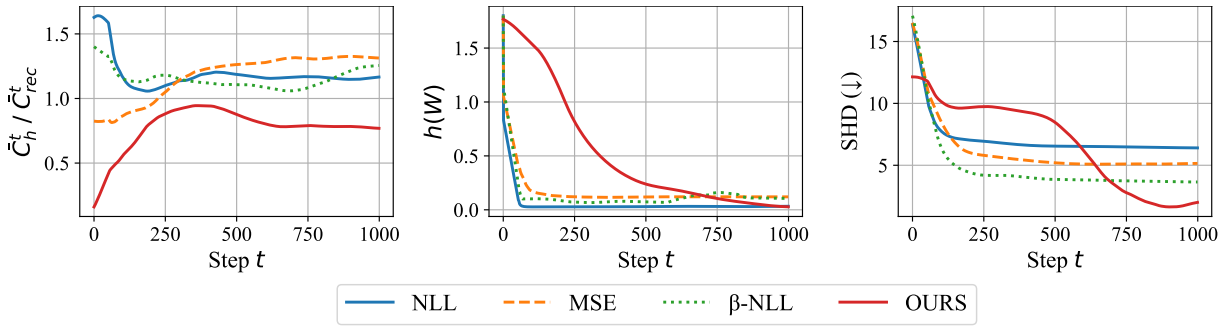
To empirically examine this phenomenon, we analyzed the contributions of the reconstruction loss and the acyclicity constraint to the updates of W throughout training. Such influence can be quantified using the contribution score of Lan et al. [15], which measures the dot product between the gradient of each loss and the corresponding parameter change at a given step. Let $\Delta W^{(s)} = W^{(s)} - W^{(s+1)}$ denote the change in W at training step s, and let $g_{\text{rec}}^{(s)}$ and $g_h^{(s)}$ be the gradients of the reconstruction loss and the acyclicity constraint at step s, respectively. We compute the dot product between them at every step, accumulate the values until t steps, and then take an average. The resulting cumulative contributions ($\bar{C}_{\text{rec}}(t)$ and $\bar{C}_h(t)$) can be expressed by:

$$\begin{aligned} \bar{C}_{\text{rec}}(t) &= \frac{1}{t} \sum_{s=0}^{t} \left| \langle -\Delta W^{(s)}, g_{\text{rec}}^{(s)} \rangle \right|, \\ \bar{C}_h(t) &= \frac{1}{t} \sum_{s=0}^{t} \left| \langle -\Delta W^{(s)}, g_h^{(s)} \rangle \right|, \end{aligned} \tag{5}$$

where $\langle \cdot \rangle$ denotes the dot product. Thus, for example, if $\bar{C}_h(t)$ exceeds $\bar{C}_{\text{rec}}(t)$, it indicates that the acyclicity constraint has been the primary driver of the learning of W.

We examined NLL, MSE [37], and β-NLL [24]; for each reconstruction loss type, we measured three metrics over training steps: (a) the ratio $\bar{C}_h(t)/\bar{C}_{\text{rec}}(t)$, (b) the value of the acyclicity constraint $h(W)$, and (c) the Structural Hamming Distance (SHD) between W and the ground-truth DAG at each step (lower SHD is better). The experiments were conducted on the synthetic data from Erdős–Rényi (ER) graph under an HNM setting with $d = 10$ (see Section G.5 of our supplementary material for detailed settings). In Figure 1(a), for MSE, we observe $\bar{C}_h(t) > C_{\text{rec}}(t)$ at around 250 steps; for NLL and β-NLL, the ratio exceeds 1, indicating that the acyclicity constraint contributed more to W updates than the reconstruction loss even from the very beginning of training. This trend persisted throughout training. In Figure 1(b), they force $h(W)$ toward zero extremely quickly at the early phase of learning. Consistent with this, Figure 1(c) shows that their SHD curves quickly flattened out and remained nearly constant thereafter, implying that the structure has ceased to change because SHD is a metric highly sensitive to edge additions, deletions, and reversals.

Remark. As a result, if the acyclicity constraint dominates too early, the structure can get prematurely *locked-in* by the constraint instead of being learned from the data. This effect may become more pronounced as the penalty on the acyclicity constraint increases. Therefore, we propose a training strategy that ensures W is guided



(a) Contribution ratio (b) DAG constraint (c) Difference with True DAG

Figure 1: (a) The cumulative contribution ratio $\bar{C}_h(t)/\bar{C}_{\text{rec}}(t)$; a value above 1 indicates that $h(W)$ contributed more to the update of W than the reconstruction loss. (b) Value of $h(W)$ during training steps. (c) Structural Hamming Distance between W and the true DAG over training steps.

by data reconstruction in the early stages, and later *progressively enforces* the acyclicity constraint. As shown in Figure 1, our method (a) pays more attention to the reconstruction loss in the early stages, guiding W to prioritize data likelihood, (b) shows higher $h(W)$ in the early epochs but gradually pushes $h(W) \to 0$ over time, and thus (c) achieves the lowest SHD.

3 Proposed Method

The optimization issue observed in the previous section arises because the variance function is overestimated before the mean and variance functions are sufficiently trained, causing the contribution of the NLL loss to updates of W to be diminished. We thus propose an optimization strategy tailored to this challenge specific to DAG learning with HNMs, mainly inspired by the homotopy-based approaches [5, 13]. In the early stages, our strategy induces stable learning of the mean and variance functions by focusing on a surrogate loss, and reduces the influence of the acyclicity constraint. Subsequently, we gradually transition from the surrogate loss to the standard NLL loss; the influence of the constraint gradually increases so that W converges to a DAG while sufficiently reflecting the data likelihood.

As noted in the previous section, directly minimizing the NLL loss in a joint optimization presents challenges; therefore, we introduce a surrogate loss that decouples the mean and variance components of the NLL loss. Let $t \in [0, T]$ denote the current training step and T be the time at which training terminates due to convergence of the penalty term. The overall loss function is defined over t as:

$$\mathcal{L}_{\text{total}}(\Theta, W) = \begin{cases} \mathcal{L}_{\text{surrogate}}(\Theta, W, t) + \frac{\rho}{2} h(W)^2 + \alpha h(W), & t < t^*, \\ \mathcal{L}_{\text{NLL}}(\Theta, W) + \frac{\rho}{2} h(W)^2 + \alpha h(W), & t \geq t^*, \end{cases} \tag{6}$$

where $\mathcal{L}_{\text{surrogate}}(\Theta; t)$ denotes the surrogate loss for the initial training phase and t^* denotes the pre-defined transition point. For $t < t^*$ (the early phase), we prioritize the surrogate loss to ensure stable convergence of the predictors, then smoothly transition to the NLL loss as t approaches t^*. In what follows, Section 3.1 details how the surrogate loss facilitates stable initial convergence of the mean and variance functions and the process of transitioning to the NLL loss, and Section 3.2 discusses the rationale behind our surrogate loss and optimization strategy.

3.1 Gradual Transition of the Objective

Our surrogate loss serves as an initial training objective to prevent W from being pushed in a wrong direction by unstable likelihood gradients, while also acting as a bridge toward the eventual NLL loss. It is defined as:

$$\mathcal{L}_{\text{surrogate}}(\Theta, W, t) = \lambda_{\text{reg}}(t) \cdot \Big((\mathcal{L}_{\text{MSE}}(\Theta_\mu, W) + \mathcal{L}_{\text{VarReg}}(\Theta, W) \Big) + \mathcal{L}_{\text{StopNLL}}(\Theta, W), \tag{7}$$

where $\lambda_{\text{reg}}(t)$ is a smooth scheduling coefficient that starts at a high value in early training, so that the loss terms aligning the mean and variance estimates are given greater weight, and gradually decays to ensure a stable transition. $\lambda_{\text{reg}}(t)$ is defined as:

$$\lambda_{\text{reg}}(t) = \lambda_{\text{reg}}(0) \cdot \exp\left(-\frac{t}{t^*/\tau}\right), \quad \lambda_{\text{reg}}(0) \gg 1, \tag{8}$$

where τ controls the rate of this decay.

3.1.1 Decoupling the NLL components. By design, when t is small ($\lambda_{\text{reg}}(t)$ is large), the loss terms $\mathcal{L}_{\text{MSE}}$ and $\mathcal{L}_{\text{VarReg}}$ dominate the objective. This phase is intended to train the mean and variance functions independently, recognizing that joint optimization at the early steps can impede the convergence of each function. Prior work [26, 28] showed that reliable learning of the variance function strongly depends on a good fit of the mean function. If the mean function has not yet sufficiently converged, the prediction residuals will not accurately reflect the data distribution and will be biased; a variance network that takes these biased residuals as input is at risk of learning a distorted signal. Therefore, in the early stage of training, we induce the mean function to converge independently via $\mathcal{L}_{\text{MSE}}$, and we accurately calibrate the variance function via residual-based regularization loss $\mathcal{L}_{\text{VarReg}}$. First, $\mathcal{L}_{\text{MSE}}$ is computed by:

$$\mathcal{L}_{\text{MSE}}(\Theta_\mu, W) = \sum_{j=1}^{d} \left(X_j - \mu_j(X; W_j, \theta_\mu)\right)^2. \tag{9}$$

Using the standard NLL loss too early (before the mean has converged) can cause the variance estimator to inflate its predictions to trivially reduce the loss [24]. To mitigate this, we introduce $\mathcal{L}_{\text{VarReg}}$, which serves to regularize the variance in the initial phase such that the variance function reflects the actual magnitude of the prediction error:

$$\mathcal{L}_{\text{VarReg}}(\Theta, W) = \sum_{j=1}^{d} \left(\sigma_j^2(X; W_j, \theta_\sigma) - (X_j - \mu_j(X; W_j, \theta_\mu))^2 \right)^2. \tag{10}$$

This regularization term encourages the predicted variance to align with the observed residual magnitude, thereby preventing the variance from growing excessively when the mean function has not yet converged. In the initial phase of training, $\mathcal{L}_{\text{MSE}}$ improves the accuracy of the mean prediction, and $\mathcal{L}_{\text{VarReg}}$ keeps the predicted variance from overestimating the mean error. Together, these two loss terms ensure that each function can converge stably. They are applied with a large initial weight $\lambda_{\text{reg}}(t)$, promoting better data reconstruction of the predictor networks.

Remark. To see how this decoupling operates during training, observe that the surrogate gradients are $\nabla_{\mu_j} \mathcal{L}_{\text{MSE}} = -2(X_j - \mu_j)$ and $\nabla_{\sigma_j^2} \mathcal{L}_{\text{VarReg}} = 2(\sigma_j^2 - (X_j - \mu_j)^2)$. The mean update no longer carries the $1/\sigma_j^2$ factor that scales $\nabla_\mu \mathcal{L}_{\text{NLL}}$, so the mean network's progress is no longer modulated by the variance estimate. Conversely, $\mathcal{L}_{\text{VarReg}}$ pulls σ_j^2 directly toward the squared residual without back-propagating through μ_j, so the variance update does not amplify errors from a still-converging mean network.

In Section 4.5 and Section B.2 of our supplementary material, we provide a justification and analysis of these two loss terms based on gradient consistency and an analysis on curvature and condition number.

3.1.2 Stop-gradient NLL. Despite the stabilizing effect of $\mathcal{L}_{\text{VarReg}}$, this term alone does not guarantee that the predictive variance reflects the true conditional variance. For instance, if we simply carry over the variance learned by $\mathcal{L}_{\text{VarReg}}$ into the NLL loss, the overall loss can still remain high – indicating that the predicted variance is not satisfying the likelihood-optimal conditions. Since the ultimate goal of structure learning is to optimize both the NLL and the acyclicity constraint simultaneously, the variance network should eventually be oriented toward minimizing the NLL as well. To achieve this, we introduce a modified NLL loss with a stop-gradient operation as follows:

$$\mathcal{L}_{\text{StopNLL}}(\Theta, W) = \sum_{j=1}^{d} \left[\frac{1}{2} \log \sigma_j^2(\lfloor X; W_j \rfloor, \theta_\sigma) + \frac{(X_j - \lfloor \mu_j(X; W_j, \theta_\mu) \rfloor)^2}{2\sigma_j^2(\lfloor X; W_j \rfloor, \theta_\sigma)} \right], \tag{11}$$

where $\lfloor \cdot \rfloor$ denotes the stop-gradient operation. Thus, $\mathcal{L}_{\text{StopNLL}}$ does not back-propagate through the mean function or the structural parameters (including W) and only updates the variance function parameters. By doing so, we design a scenario where the variance can be trained reliably according to the NLL criterion even if the mean function has not yet converged to its optimal estimate. This addresses the limitation of $\mathcal{L}_{\text{VarReg}}$: whereas $\mathcal{L}_{\text{VarReg}}$ adjusts the variance based on the current residual (which can be distorted if the mean is inaccurate), $\mathcal{L}_{\text{StopNLL}}$ blocks this indirect pathway. As a result, the variance estimation is not influenced by the mean function's biased predictions.

3.1.3 Transition to the final objective. Once the variance estimates have been sufficiently stabilized via $\mathcal{L}_{\text{StopNLL}}$, we switch the overall loss back to the original NLL loss. At this point, we jointly optimize both the mean and variance functions, and the DAG constraint is now actively imposed in the loss, guiding the model to learn an acyclic structure based on the learned representations. This transition is handled smoothly by the scheduling coefficient (i.e. by the decay of $\lambda_{\text{reg}}(t)$). Sections B.1 and H.7 of our supplementary material analyzes the convergence characteristics and time complexity of our method, respectively. Section C of our supplementary material provides the results of the contribution ratio analysis with respect to t^* and $\lambda_{\text{reg}}(0)$ and the trajectory analysis of the training process. The overall training procedure is in Section F.

3.2 Analysis of Weighted Loss Scheduling

We define the standard training objective that combines the NLL loss and the ALM-based acyclicity constraint as follows:

$$r_{\rho,n}^{\text{NLL}}(W) \;:=\; \inf_{\Theta} \left\{ \widehat{\mathbb{E}}\big[\, \mathcal{L}_{\text{NLL}}(\Theta, W) \,\big] \;+\; \tfrac{\rho}{2}\, h(W)^2 + \alpha\, h(W) \right\}, \tag{12}$$

where $\widehat{\mathbb{E}}$ denotes the empirical expectation. The challenge is the inherent instability that arises when directly optimizing $r_{\rho,n}^{\text{NLL}}(W)$. As discussed in Section 2.2, in the early stage of training, the variance-attenuation phenomenon weakens the reconstruction gradient (g_{rec}). This weakened signal is easily overwhelmed by the constraint gradient (g_h) as the penalty coefficient ρ in the ALM gradually increases, leading to a situation where the DAG structure is determined prematurely by the acyclicity constraint rather than by the data. This failure mechanism can be quantitatively characterized by a *failure certificate* (using the cumulative contribution measure defined in Eq. (5)) as follows:

Definition 3 (Failure certificate). *If there exist constants $T_0 > 0$, $\gamma > 1$, and $\varepsilon \geq 0$ such that*

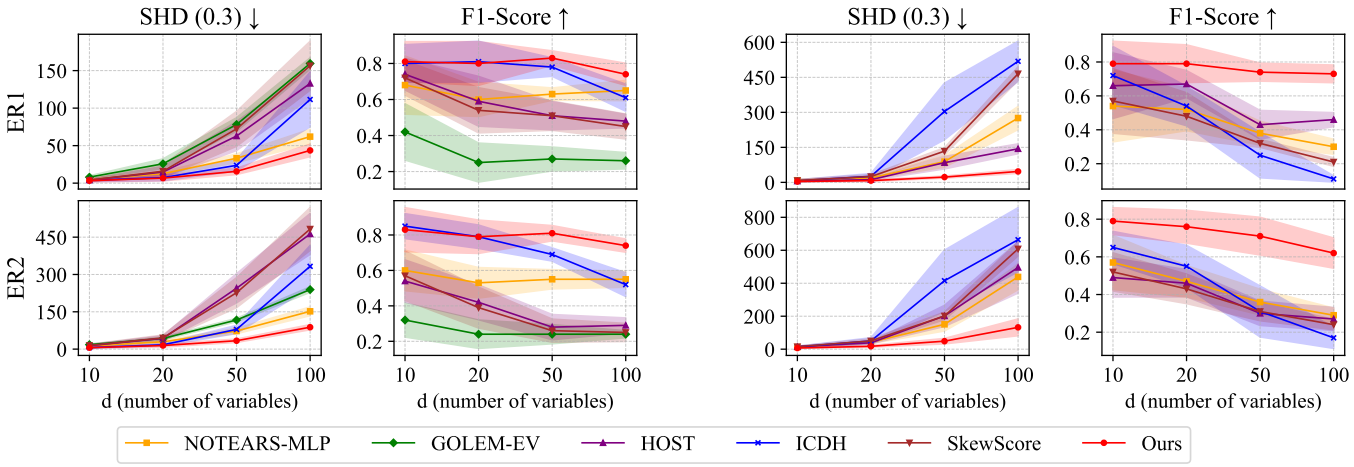
$$\frac{C_h(t)}{C_{\text{rec}}(t)} \;\geq\; \gamma, \quad (0 \leq t \leq T_0), \qquad h(W_{T_0}) \leq \varepsilon, \tag{13}$$

we call the interval $[0, T_0]$ a constraint-dominant early phase and the tuple $(T_0, \gamma, \varepsilon)$ a failure certificate.

As empirically observed in Figure 1, optimizing the standard objective $r_{\rho,n}^{\text{NLL}}(W)$ entails the aforementioned failure certificate, which becomes a fundamental cause hindering the search for the optimal DAG structure. To address this issue, we introduce a surrogate loss and a weight scheduling scheme to enforce a *reconstruction-dominant* phase in the early stage of training. The condition required for our proposed method to achieve a reconstruction-dominant phase is provided in Section C of our supplementary material.

Lemma 3.1 (Sufficient condition for an early reconstruction-dominant phase). *If, during the initial phase $t \leq T_1$ under $(0 < \delta < 1)$, the following holds:*

$$\lambda_{\text{reg}}(t) \left\| \nabla_W (\mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{VarReg}}) \right\| \;\geq\; \frac{\kappa}{1-\delta} \left\| \nabla_W \left(\tfrac{\rho}{2} h(W)^2 + \alpha h(W) \right) \right\|, \tag{14}$$



(a) Homoscedastic results (b) Heteroscedastic results

Figure 2: Comparison results on the synthetic datasets. The colored lines show the mean value across 10 runs, and the shaded areas around each curve indicate the corresponding standard error.

then the gradient contribution ratio is bounded as:

$$\frac{C_h(t)}{C_{\text{rec}}(t)} \le 1-\delta \qquad (\forall\, t \le T_1), \tag{15}$$

where $\delta \in (0,1)$ is the safety margin. This reverses the failure-certificate condition and guarantees a reconstruction-dominant phase in early training.

Its proof can be found in Section D.4 of our supplementary material, and we empirically verify the sufficient condition in Section C.1.

Lemma 3.1 formalizes the conditions under which the early training phase is reconstruction-dominant and provides a theoretical handle for understanding how our method operates. Guaranteeing such a reconstruction-dominant phase at the beginning of training has a direct impact on the quality of the final solution. We denote by ξ_{NLL} the minimal excess empirical risk achievable with the standard objective $r^{\text{NLL}}_{\rho,n}$, and by ξ_{ours} the minimal excess empirical risk achievable with our method. Based on Lemma 3.1 and on the stability of optimization analysis in Section C.2 of our supplementary material. Claim 1 below shows that, for the same number of samples, $\xi_{\text{ours}} < \xi_{\text{NLL}}$ holds.

Claim 1 (Sample-dependent sufficient condition). *Under the standard assumptions (i)–(iv) summarized in Section D.6 of our supplementary material (namely, uniform convergence of $r_{\rho,n}$ to r_ρ, a local PL-type condition along the data-induced solution path, a slow schedule $\lambda_{\text{reg}}(t)$ that maintains a stable homotopy path, and the existence of an early reconstruction-dominant phase as defined in Lemma 3.1), the proposed optimization strategy achieves a smaller excess risk than standard NLL training:*

$$\xi_{\text{ours}} \le C_1 \Delta_n + C_2 \frac{\kappa \left|\nabla_W \left(\frac{\rho}{2} h(W)^2 + \alpha h(W)\right)\right|^2}{\lambda_{\text{reg}}(t_1) \left|\nabla_W (\mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{VarReg}})\right|^2} < \xi_{\text{NLL}}. \tag{16}$$

Its assumptions and proof are discussed in detail in Section D.6 of our supplementary material. Claim 1 should be viewed not as a general guarantee but as an idealized result that provides intuition under these standard assumptions.

4 Evaluation

4.1 Experimental Settings

We evaluated the performance of the proposed method on both synthetic datasets designed to include heteroscedastic noise and four real-world datasets. We also considered the homoscedastic condition in our evaluation on synthetic data. Section G of our supplementary material reports more detailed experimental settings.

4.1.1 Datasets. We followed the settings of prior studies [14, 33, 34, 37] in generating synthetic data. The ground-truth causal graph was sampled from an Erdős–Rényi (ER) model with parameter $k \in \{1, 2\}$, which controls the average number of edges. We considered numbers of variables $d \in \{10, 20, 50, 100\}$ and generated 10 random graphs for each d. For each graph, we then generated $N = 1000$ observations, repeating this process 10 times.

For real-world datasets, we employed Sachs [23], SynTReN [31], and Causal-Assembly [9]. Sachs consists of human cell protein expression data and includes 11 variables, 17 causal edges, and 853 observations. We used only the observational data collected without intervention. SynTReN is a simulator-based pseudo-real dataset that generates synthetic transcriptional regulatory networks and gene expression data; for 20 nodes, we generated networks and data using 10 random seeds. The Causal-Assembly Station dataset is based on sensor and actuator measurements collected from a real industrial assembly line; we used the first two stations to construct a causal graph with 40 nodes and 86 directed edges, and similarly generated data with 10 random seeds.

4.1.2 Baselines and metrics. We selected as baseline methods several representative ones that support multivariate, nonlinear DAG learning. ICDH [33], HOST [7], and SkewScore [16] explicitly model

Table 1: Results on the real-world datasets.

Method	Sachs ($d = 11$)		SynTReN ($d = 20$)		CausalAssembly ($d = 40$)	
	SHD(↓)	F1-Score(↑)	SHD(↓)	F1-Score(↑)	SHD(↓)	F1-Score(↑)
NOTEARS-MLP	15	0.26	114.6	0.13	139.4	0.06
HOST	13	0.33	114.5	0.15	183.5	**0.17**
ICDH	13	0.33	128.4	0.13	199.0	0.10
Ours	**12**	**0.43**	**89.8**	**0.18**	**135.6**	0.11

observation-level heteroscedasticity; GOLEM-NV-L1 [19], and DARING [10] allow different noise variances for each variable; NOTEARS-MLP [37] and GOLEM-EV-L1 [19], which assume homoscedastic noise, are also employed to observe performance differences across heteroscedastic and homoscedastic assumptions. We used Structural Hamming distance (SHD), which measures edge differences between the estimated DAG and the ground-truth DAG, and F1-score of the edge detection, to assess their performance.

We used the NOTEARS [36] continuous optimization framework as the backbone, and all baseline methods were run with their default settings as given in the original papers. We adopted L-BFGS [3] for gradient-based training. All networks were 2-layer MLPs with 10 hidden units, and the learned weight matrix W was binarized using the standard threshold of 0.3. In all experiments, we consistently used the same values for $\lambda_{\text{reg}}(0)$, t^*, and τ.

4.2 Results on synthetic and real-world datasets

Figure 2 shows the results on the synthetic datasets generated under both homoscedastic and heteroscedastic settings. The proposed method achieves performance comparable to or better than existing methods in the homoscedastic environment, and in the heteroscedastic environment it consistently achieves the best performance on all evaluation metrics for all tested values of d (the numbers of variables). This suggests that our method attains high accuracy in structure learning while effectively suppressing spurious (false positive) edge detection.

Notably, even as the number of variables d increases, our method's performance degrades very gradually (e.g., its F1-score drops from 0.79 at $d = 10$ to 0.73 at $d = 100$ for ER1 (and to 0.62 for ER2)). In contrast, HOST's F1-score exhibits a sharp decline in F1-score as d grows (under the heteroscedastic ER2 condition with $d = 100$); ICDH's F1-score also plunges to 0.17, and SkewScore exhibits a similar degradation as d increases. We interpret these differences as follows: existing methods, due to the instability of variance estimation in heteroscedastic settings, tend to overestimate the variance in order to compensate for mean prediction errors, which in turn leads to many spurious edges. In contrast, our method effectively decouples the learning of the mean and variance functions through the surrogate loss and the graduated loss weighting strategy, preventing edge detection that is not grounded in the data during the early stages of training and enabling more reliable structure estimation. Section H.1 of our supplementary material reports the result on ER3, ER4 and ER5 (dense graphs) and Section H.2 shows the results in an extreme case, $d = 1000$, where our method also achieves the best performance.

Table 1 presents the comparative results on the real-world datasets. The proposed method achieves the best performance in terms of both SHD and F1-score, with only one exceptional case in the CausalAssembly dataset. This shows that the proposed graduated optimization strategy can effectively estimate the DAG structure on real-world scenarios. Section H.3 of our supplementary material reports the result on real-world Cause-Effect Pairs [25] dataset; we also achieve the best here.

4.3 Ablation Study

We analyzed the contribution of each component of the surrogate loss to the DAG learning performance. ER1 with $d \in \{10, 20, 50\}$ under HNM was used here. We compared the full proposed model (with all components of the surrogate loss) to ablated versions where we removed each of $\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_{\text{VarReg}}$, or $\mathcal{L}_{\text{StopNLL}}$.

Figure 3(a) indicates that removing $\mathcal{L}_{\text{MSE}}$ yields the highest SHD (worst performance). For the NLL loss to properly learn the variance, the mean function must be sufficiently converged first. If the MSE term is removed, due to a large discrepancy between the predictive mean and true values, the variance prediction becomes distorted, and overall performance degrades. The $\mathcal{L}_{\text{VarReg}}$ and $\mathcal{L}_{\text{StopNLL}}$ terms likewise play important roles in the initial stage by stabilizing variance estimation and preventing its overestimation; removing either of them also leads to performance deterioration. By contrast, with all three components, our method achieves the lowest SHD across all conditions. We performed the similar ablation study on the real-world datasets, and the results are reported in Table 2. We observe the same trend as in the experiments on the synthetic dataset: removing any single component of the surrogate loss consistently degraded performance relative to the full model. This shows that the contribution of each term is not limited to the synthetic setting. Therefore, we can confirm that each component of our surrogate loss is crucial for effective structure learning.

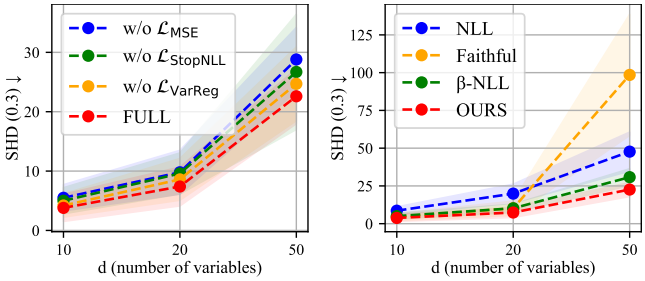
As an analysis of the effect of the graduated optimization, Section H.5 of our supplementary material shows the result of applying it to NOTEARS-MLP; this helps improve performance, but our full framework that uses the surrogate loss achieves even higher performance. In addition, Section 4.6 in the next page shows the influence of the scheduling parameters, showing that increasing t^* and $\lambda_{\text{reg}}(0)$ stabilizes the optimization and improves structure learning performance, but if their values become too large, performance deteriorates. This result is consistent with our overall design premise.

4.4 Application to Heteroscedastic Regression

In addition, we investigated whether existing optimization solutions in the context of heteroscedastic regression can be directly applied to our DAG learning problem. The first issue we identified is that naively optimizing the NLL is difficult from an optimization

Table 2: Ablation study on real-world datasets. For Sachs, SynTReN, and CausalAssembly, we report SHD and F1-score. For Cause-Effect Pairs, we report accuracy and weighted accuracy.

Variant	Sachs		SynTReN		CausalAssembly		Cause-Effect Pairs	
	SHD ↓	F1 ↑	SHD ↓	F1 ↑	SHD ↓	F1 ↑	Accuracy ↑	Weighted Acc. ↑
−w/o L_{MSE}	16	0.21	111.3	0.11	303.6	0.09	49/99	0.55
−w/o L_{StopNLL}	14	0.28	98.0	0.12	187.0	0.09	48/99	0.53
−w/o L_{VarReg}	14	0.31	99.9	0.11	198.9	0.08	44/99	0.54
Full	**12**	**0.43**	**89.8**	**0.18**	**135.6**	**0.11**	**58/99**	**0.66**



(a) Ablation study (b) Comparison with HRs

Figure 3: (a) ablation study. (b) comparison with the strategies in heteroscedastic regression (HR).

perspective under HNM. Faithful Heteroscedastic [28] and β-NLL [24] are the existing approaches to address this issue. However, according to our hypothesis, these methods do not tackle problems specific to structure learning—such as the dominance of the acyclicity constraint over the NLL loss in learning W or the initial stabilization of variance estimation—and thus their performance in DAG learning could be limited. We empirically validated this by comparing models trained with the standard NLL, with β-NLL, and with ours, under the same experimental setting.

Figure 3(b) shows the SHD results on ER1 with $d \in \{10, 20, 50\}$ under HNM. The standard NLL baseline has the highest SHD for $d = 10$ and 20, whereas at $d = 50$ Faithful Heteroscedastic performs the worst. β-NLL performs relatively well compared to the other two. While Faithful Heteroscedastic and β-NLL may stabilize variance learning, (as we expected) they do not consider the influence of the acyclicity constraint $h(W)$ during training, and consequently their DAG estimation performance is worse than that of our method. As a result, the proposed surrogate-loss-based graduated optimization strategy is specifically designed to address the challenges of DAG learning under heteroscedasticity, which sets it apart from existing heteroscedastic regression methods. Section H.4 of our supplementary material shows an experiment where our optimization strategy is applied to the existing NLL losses.

4.5 Surrogate Loss Analysis

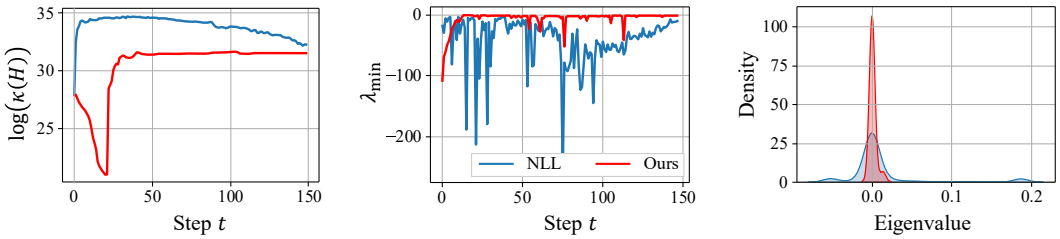
We empirically compare our surrogate loss and the NLL loss. We computed their Hessian matrices and analyzed their condition numbers, minimum eigenvalues ($\lambda_{\min}$), and eigenvalue distributions. We generated synthetic data ten times with $d = 10$ under the ER1 setting, and performed DAG learning under the heteroscedastic conditions described in Section 3.1.

Figure 4 summarizes our results. (a) The surrogate loss exhibited a lower condition number compared to the NLL loss, which indicates that it has a more well-conditioned curvature landscape. (b) The minimum eigenvalue of the NLL loss was significantly skewed towards negative values, which means the presence of saddle points in the loss surface and potentially unstable optimization behavior. Conversely, our surrogate loss maintained a minimum eigenvalue $\lambda_{\min}$ close to zero, indicating a generally flatter loss surface. (c) The eigenvalue distribution of the NLL loss exhibited a long tail, whereas the surrogate loss showed eigenvalues concentrated near zero. This implies that the surrogate loss landscape is flatter, which can lead to convergence to flatter minima and potentially better generalization performance. In conclusion, this empirical analysis supports the theoretical perspective, demonstrating that the surrogate loss indeed possesses a lower condition number compared to the NLL loss. This indicates that the surrogate loss is a more well-conditioned function with the potential for more stable convergence behavior.

4.6 Hyperparameter Analysis

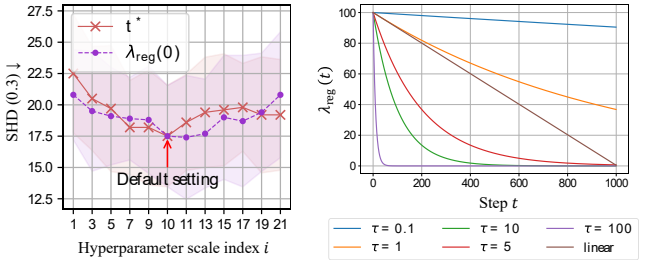
Here, we analyzed the influence of the scheduling parameters: t^* and $\lambda_{\text{reg}}(0)$. Figure 5 (a) plots SHD, where the x-axis corresponds to a scaling index i such that $t^* = 100i$ and $\lambda_{\text{reg}}(0) = 10i$ so that 20 linearly spaced values for each hyperparameter are tested (each value is repeated 10 times). This sweep is designed to empirically verify the sufficient conditions on $\lambda_{\text{reg}}(0)$ and t^*, rather than to serve as a generic sensitivity analysis. Accordingly, the range is intentionally wide enough to include failure regions in which the conditions stated in Lemma 3.1 and Section D.5 of our supplementary material break down.

We observed that as t^* increases, performance gradually improves, then beyond approximately $t^* \geq 1100$ we observed SHD tending to rise (low performance) again. This indicates that when t^* is set above a certain level, the optimization process becomes smoother and converges more stably, which aligns with the design premise of our strategy. A similar trend was observed for $\lambda_{\text{reg}}(0)$. We interpret this as follows: when $\lambda_{\text{reg}}(0)$ is sufficiently large, the influence of the acyclicity constraint term $\mathcal{L}_{\text{DAG}}$ is relatively diminished and the surrogate loss dominates the optimization, leading to better structure learning performance. However, if both schedule parameters are set too high, performance starts to deteriorate again, indicating that simply increasing these values is not always beneficial.



(a) Condition number (b) Minimum eigenvalue (c) Eigenvalue distribution

Figure 4: Hessian matrix analysis with respect to our surrogate and the NLL loss. (a) shows the condition number at each step, where a log scale is applied. (b) compares the minimum eigenvalue ($\lambda_{\min}$) of the Hessian matrix at each step. (c) draws the distribution of eigenvalues of the Hessian matrix calculated after training.



(a) Hyperparameter analysis (b) $\lambda_{\text{reg}}(t)$ trends with different scheduling

Figure 5: Hyperparameter and scheduling analysis.

Table 3: Performance comparison under different decay strategies for weighted loss scheduling.

Method	SHD ↓	F1-score ↑
linear	23.6 ± 11.842	0.69 ± 0.135
$\tau = 0.1$	26.6 ± 13.381	0.66 ± 0.093
$\tau = 1$	25.4 ± 12.052	0.68 ± 0.109
$\tau = 5$	**17.5 ± 5.408**	**0.76 ± 0.087**
$\tau = 10$	21.5 ± 8.846	0.66 ± 0.181
$\tau = 100$	28.7 ± 8.112	0.47 ± 0.266

Our weighted loss scheduling is defined in Eq. (8), where we adopted an exponential decay schedule with a decay rate of $\tau = 5$. To validate this choice, we varied τ from 0.1 to 100 and also replaced the exponential schedule with a linear schedule to compare performance. The experiments were conducted on ER2 graph with $d = 20$, repeated 10 times. Figure 5 (b) illustrates the change of $\lambda_{\text{reg}}(t)$ under each strategy, and Table 3 presents the corresponding performance.

The results show that using $\tau = 5$ yields the best performance (SHD = 17.5, F1-score = 0.76). In contrast, linear scheduling results in a noticeable performance drop (SHD = 23.6, F1-score = 0.69). The cases of $\tau = 0.1$ and $\tau = 1$ also exhibit significantly worse performance; in those cases, $\lambda_{\text{reg}}(t)$ decays more slowly than with linear scheduling and remains very high at the transition point ($t^* = 1000$). This implies that if the objective's gradual transition is too slow, or if $\lambda_{\text{reg}}(t)$ is not sufficiently low by the transition point, performance degrades. Meanwhile, for $\tau = 10$ and $\tau = 100$, $\lambda_{\text{reg}}(t)$ decays faster than for $\tau = 5$, and such faster decay leads to a larger performance drop. This suggests that $\lambda_{\text{reg}}(t)$ must decrease at an appropriate rate—neither too fast nor too slow. Consequently, we conclude that τ should be greater than 1 but less than 10, and setting $\tau = 5$ generally delivers the best performance.

We adopt this fixed schedule rather than an adaptive one, since adaptively adjusting $\lambda_{\text{reg}}(t)$ based on training statistics can introduce additional coupling with the ALM updates and early-phase variance noise, which we leave as future work.

5 Conclusions

This paper proposed a new optimization strategy for DAG learning under heteroscedastic noise. We analyzed the training instability and structure distortion issues of the existing NLL-based approach, and introduced a weighted loss scheduling-based graduated optimization scheme to mitigate these problems. In the early stage of training, the surrogate loss induces stable convergence of the mean and variance function estimators; as training progresses, we gradually transition to the standard NLL objective with the acyclicity constraint, thereby maximizing the structure learning performance. Our method achieved impressive performance on both synthetic and real datasets.

Acknowledgement

This work was supported by (1) **Samsung Electronics** Co., Ltd., (2) the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(**RS-2025-24535423**), and (3) the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(**RS-2020-II201373**, Artificial Intelligence Graduate School Program (Hanyang University)).

References

[1] Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. 2022. Dagma: Learning dags via m-matrices and a log-determinant acyclicity characterization. *Advances in Neural Information Processing Systems* 35 (2022), 8226–8239.
[2] Dimitri P Bertsekas. 2014. *Constrained optimization and Lagrange multiplier methods.* Academic press.
[3] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing* 16, 5 (1995), 1190–1208.
[4] Weilin Chen, Jie Qiao, Ruichu Cai, and Zhifeng Hao. 2023. On the role of entropy-based loss for learning causal structure with continuous optimization. *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[5] Chang Deng, Kevin Bello, Pradeep Ravikumar, and Bryon Aragam. 2023. Global optimality in bivariate gradient-based DAG learning. *Advances in Neural Information Processing Systems* 36 (2023), 17929–17968.
[6] Chang Deng, Kevin Bello, Pradeep Ravikumar, and Bryon Aragam. 2024. Markov equivalence and consistency in differentiable structure learning. *Advances in Neural Information Processing Systems* 37 (2024), 91756–91797.
[7] Bao Duong and Thin Nguyen. 2023. Heteroscedastic causal structure learning. In *ECAI 2023*. IOS Press, 598–605.
[8] Ming Gao, Yi Ding, and Bryon Aragam. 2020. A polynomial-time algorithm for learning nonparametric causal graphs. *Advances in Neural Information Processing Systems* 33 (2020), 11599–11611.
[9] Konstantin Göbler, Tobias Windisch, Mathias Drton, Tim Pychynski, Martin Roth, and Steffen Sonntag. 2024. `causalAssembly`: Generating Realistic Production Data for Benchmarking Causal Discovery. In *Causal Learning and Reasoning*. PMLR, 609–642.
[10] Yue He, Peng Cui, Zheyan Shen, Renzhe Xu, Furui Liu, and Yong Jiang. 2021. Daring: Differentiable causal discovery with residual independence. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 596–605.
[11] Alexander Immer, Christoph Schultheiss, Julia E Vogt, Bernhard Schölkopf, Peter Bühlmann, and Alexander Marx. 2023. On the identifiability and estimation of causal location-scale noise models. In *International Conference on Machine Learning*. PMLR, 14316–14332.
[12] Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. 2021. Causal autoregressive flows. In *International conference on artificial intelligence and statistics*. PMLR, 3520–3528.
[13] Joon-Hyuk Ko, Hankyul Koh, Nojun Park, and Wonho Jhe. 2023. Homotopy-based training of neuralodes for accurate dynamics discovery. *Advances in Neural Information Processing Systems* 36 (2023), 64725–64752.
[14] Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. 2020. Gradient-Based Neural DAG Learning. In *International Conference on Learning Representations*.
[15] Janice Lan, Rosanne Liu, Hattie Zhou, and Jason Yosinski. 2019. Lca: Loss change allocation for neural network training. *Advances in neural information processing systems* 32 (2019).
[16] Yingyu Lin, Yuxing Huang, Wenqin Liu, Haoran Deng, Ignavier Ng, Kun Zhang, Mingming Gong, Yian Ma, and Biwei Huang. 2025. A skewness-based criterion for addressing heteroscedastic noise in causal discovery. In *International Conference on Learning Representations*, Vol. 2025. 89283–89310.
[17] Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. 2016. Distinguishing cause from effect using observational data: methods and benchmarks. *Journal of Machine Learning Research* 17, 32 (2016), 1–102.
[18] Leland Gerson Neuberg. 2003. Causality: models, reasoning, and inference, by judea pearl, cambridge university press, 2000. *Econometric Theory* 19, 4 (2003), 675–685.
[19] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. 2020. On the role of sparsity and dag constraints for learning linear dags. *Advances in Neural Information Processing Systems* 33 (2020), 17943–17954.
[20] Rainer Opgen-Rhein and Korbinian Strimmer. 2007. From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC systems biology* 1 (2007), 1–10.
[21] J Peters, J Mooij, D Janzing, and B Schölkopf. 2011. Identifiability of causal graphs using functional models. In *27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*. 589–598.
[22] Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. 2014. Causal discovery with continuous additive noise models. *The Journal of Machine Learning Research* 15, 1 (2014), 2009–2053.
[23] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. 2005. Causal protein-signaling networks derived from multiparameter single-cell data. *Science* 308, 5721 (2005), 523–529.
[24] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. 2022. On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks. In *International Conference on Learning Representations*.
[25] Eleni Sgouritsa, Dominik Janzing, Philipp Hennig, and Bernhard Schölkopf. 2015. Inference of cause and effect with unsupervised inverse regression. In *Artificial intelligence and statistics*. PMLR, 847–855.
[26] Nicki Skafte, Martin Jørgensen, and Søren Hauberg. 2019. Reliable training and estimation of variance networks. *Advances in Neural Information Processing Systems* 32 (2019).
[27] Peter Spirtes, Clark N Glymour, and Richard Scheines. 2000. *Causation, prediction, and search*. MIT press.
[28] Andrew Stirn, Harm Wessels, Megan Schertzer, Laura Pereira, Neville Sanjana, and David Knowles. 2023. Faithful heteroscedastic regression with neural networks. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 5593–5613.
[29] Eric V Strobl and Thomas A Lasko. 2023. Identifying patient-specific root causes with the heteroscedastic noise model. *Journal of Computational Science* 72 (2023), 102099.
[30] Hiroshi Takahashi, Tomoharu Iwata, Yuki Yamanaka, Masanori Yamada, and Satoshi Yagi. 2018. Student-t Variational Autoencoder for Robust Density Estimation.. In *IJCAI*. 2696–2702.
[31] Tim Van den Bulcke, Koenraad Van Leemput, Bart Naudts, Piet van Remortel, Hongwu Ma, Alain Verschoren, Bart De Moor, and Kathleen Marchal. 2006. SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC bioinformatics* 7, 1 (2006), 43.
[32] Sascha Xu, Alexander Marx, Osman Mian, and Jilles Vreeken. 2022. Causal inference with heteroscedastic noise models. In *Proceedings of the AAAI Workshop on Information Theoretic Causal Inference and Discovery*.
[33] Naiyu Yin, Tian Gao, Yue Yu, and Qiang Ji. 2024. Effective causal discovery under identifiable heteroscedastic noise model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 16486–16494.
[34] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. 2019. DAG-GNN: DAG structure learning with graph neural networks. In *International conference on machine learning*. PMLR, 7154–7163.
[35] Bin Zhang, Chris Gaiteri, Liviu-Gabriel Bodea, Zhi Wang, Joshua McElwee, Alexei A Podtelezhnikov, Chunsheng Zhang, Tao Xie, Linh Tran, Radu Dobrin, et al. 2013. Integrated systems approach identifies genetic nodes and networks in late-onset Alzheimer's disease. *Cell* 153, 3 (2013), 707–720.
[36] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. 2018. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems* 31 (2018).
[37] Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. 2020. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*. Pmlr, 3414–3425.

Appendix

Code and Supplementary Material

To support reproducibility, we publicly release our source code, datasets, and scripts used in the main experiments. In addition, all detailed derivations, proofs, additional experiments, and extended analyses are provided in a separate Supplementary Material:

- **Code repository:** https://github.com/Sinegi/HNM/
- **Supplementary Material:** https://github.com/Sinegi/HNM/blob/main/Supplementary_Material.pdf

Overview of the Supplementary Material

To help readers locate specific material of interest, we summarize the contents of the Supplementary Material below.

Section A — Related Work. We provide an extended discussion of (i) classical causal discovery methods, (ii) continuous optimization-based DAG learning methods, and (iii) prior work under different noise variance assumptions. A detailed comparison with ICDH [33] is also included, together with a clarification of the distinction between heteroscedastic noise and heterogeneous data.

Section B — Details of Our Surrogate Loss. We analyze the convergence characteristics of the surrogate loss under a smoothness assumption on the scheduling coefficient $\lambda_{\text{reg}}(t)$, and further justify the surrogate by examining its gradient consistency with the NLL at stationarity. A Hessian-based analysis on the curvature and condition number is also presented, along with a toy heteroscedastic regression example that empirically supports the stability of the surrogate.

Section C — Details of Graduated Optimization. We connect our optimization strategy to homotopy optimization and graduated non-convexity, and state the assumptions used in Section 3.2 of the main paper. We also provide an empirical analysis on the conditions for the schedule parameters $\lambda_{\text{reg}}(0)$ and t^*, and visualize the

optimization trajectories projected onto a two-dimensional space via PCA.

Section D — Details for Section 3.2. We provide the formal notation, the failure certificate, auxiliary inequalities, and the full proof of emma 3.1. The path stability under slow scheduling is also discussed, and the proof of Claim 1 is presented in detail under the standard assumptions (uniform convergence, local PL-type condition, slow schedule, and reconstruction-dominant phase).

Section E — Identifiability of Multivariate HNM.. We restate the identifiability conditions of HNMs in the bivariate setting and extend the argument to the multivariate case, following prior studies [7, 22, 33]. We also discuss how our method preserves population-level identifiability, since it ultimately optimizes the same NLL-based objective.

Section F — Algorithm. We provide the pseudocode of the proposed graduated optimization based on L-BFGS, and the overall training procedure based on the augmented Lagrangian method.

Section G — Experiment Settings. We report the detailed settings for synthetic data generation under both homoscedastic and heteroscedastic conditions, descriptions of the baselines, the evaluation metrics, implementation details, and the configuration used for the motivating analysis in Section 2.2.

Section H — Additional Experiments. We report additional empirical results that could not be included in the main paper, including: performance on denser graphs (ER3, ER4, ER5); performance on a larger graph with $d = 1{,}000$; results on the Cause-Effect Pairs dataset; further analysis of the surrogate loss and the weighted loss scheduling; an application of the weighted loss scheduling to NOTEARS-MLP (a constant-variance model); comparison with methods designed for heterogeneous data; time complexity analysis and an empirical comparison of computational cost; and a comprehensive comparison with additional variance estimation methods (GraN-DAG [14], NPVAR [8], NOTEARS-ENT [4], DAGMA [1], and MCP [6]).

Section I — Supplementary Results for Figure 2. We report the full numerical results (SHD, F1-score, TPR, and FDR, with means and standard deviations over 10 trials) corresponding to the synthetic-data results in Figure 2 of the main paper, for both heteroscedastic and homoscedastic settings.

Section J — Supplementary Results for Table 1. We additionally report TPR and FDR on the real-world datasets (Sachs, SynTReN, and CausalAssembly), complementing the SHD and F1-score results in Table 1 of the main paper.