



Universidade Federal do Amazonas
Instituto de Computação
Programa de Pós-Graduação em
Informática



Richard Hada Degaki

Fighting against data scarcity: a Cold Diffusion Framework for Sparse Covariance Matrix Estimation

Manaus - AM
April 2026

Richard Hada Degaki

Fighting against data scarcity: a Cold Diffusion Framework for Sparse Covariance Matrix Estimation

Master's thesis submitted to the Graduate Program
in Computer Science of the Institute of Computing at
the Federal University of Amazonas (ICOMP-UFAM),
as partial fulfillment of the requirements for the de-
gree of Master of Science (MSc) in Computer Science.

Supervisor: *Prof. Dr. Juan Gabriel Colonna*

Manaus - AM
2026

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

D317f Degaki, Richard Hada
 Fighting against data scarcity: a Cold Diffusion Framework for
 Sparse Covariance Matrix Estimation / Richard Hada Degaki. -
 2026.
 98 f. : il., color. ; 31 cm.

 Orientador(a): Juan Gabriel Colonna.
 Dissertação (mestrado) - Universidade Federal do Amazonas,
 Programa de Pós-Graduação em Informática, Manaus, 2026.

 1. Cold diffusion. 2. Covariance estimation. 3. Machine learning.
 4. Generative models. 5. Noise reversal. I. Colonna, Juan Gabriel.
 II. Universidade Federal do Amazonas. Programa de Pós-
 Graduação em Informática. III. Título



Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Informática

FOLHA DE APROVAÇÃO

"FIGHTING AGAINST DATA SCARCITY: A COLD DIFFUSION FRAMEWORK FOR SPARSE COVARIANCE MATRIX ESTIMATION"

RICHARD HADA DEGAKI

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Dr. Juan Gabriel Colonna - **Presidente**

Prof. Dr. Marco Antônio Pinheiro Cristo - **Membro Interno**

Prof. Dr. Sandro Marcelo Ruiz - **Membro Externo**

Manaus, 30 de abril de 2026.



Documento assinado eletronicamente por **Juan Gabriel Colonna, Professor do Magistério Superior**, em 05/05/2026, às 01:34, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Marco Antônio Pinheiro de Cristo, Professor do Magistério Superior**, em 05/05/2026, às 08:58, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3113102** e o código CRC **6C1AD7D5**.

Referência: Processo nº 23105.014743/2026-31

SEI nº 3113102

Acknowledgments

First and foremost, I would like to thank my parents, Christhina and Carlos, for being my foundation and my safe harbor. Your unconditional support and sacrifices were essential for me to pursue this path and reach this milestone.

To my fiancée, Mariane, for your love, patience, and tireless companionship throughout this journey. Thank you for believing in me, for listening to my doubts, and for celebrating every small victory.

To my advisor, Prof. Juan, for the invaluable guidance, trust, and precise feedback that made this work possible.

To my colleagues, for the technical and academic support that enriched this study.

Abstract

Accurately estimating covariance matrices from limited samples is a fundamental challenge in high-dimensional statistics. Traditional methods attempt to mitigate the effects of data scarcity through fixed statistical priors, such as regularization or shrinkage, which may not capture the true underlying data structure. This thesis introduces a new paradigm that reframes covariance estimation as a generative denoising problem. We conceptualize the noisy Sample Covariance Matrix ($\hat{\Sigma}$) as the result of a non-Gaussian statistical degradation process and develop a framework that learns to reverse this process.

Our primary contribution is ColdCov, a novel deep generative model based on the Cold Diffusion framework. ColdCov is uniquely suited to this problem for two reasons. First, it is designed to reverse arbitrary non-Gaussian degradations, aligning perfectly with the statistical properties of the degradation process. Second, its output architecture enforces the symmetric positive semi-definite (SPSD) property by construction, guaranteeing that every output is a mathematically valid covariance matrix.

Through comprehensive experiments, we show that ColdCov provides strong advantages in covariance reconstruction, particularly in low-sample regimes, achieving lower Frobenius reconstruction error and competitive precision-matrix quality relative to established baselines. However, the results also reveal a task-dependent trade-off: sparsity-enforcing methods such as Graphical Lasso and Ledoit-Wolf remain stronger for explicit graph-structure recovery, while ColdCov produces richer denoised covariance representations. The framework’s practical utility is further validated on the ABIDE neuroimaging dataset, where precision matrix features extracted by ColdCov show best observed mean performance on downstream discriminative classification, with modest gains over baselines in a Control vs. Autism subject-level classification task.

We conclude that modeling covariance estimation as a generative denoising task is a promising complement to traditional regularization: it is particularly effective for covariance reconstruction and downstream discriminative feature extraction, while sparse regularization remains preferable when exact graph topology recovery is the primary objective.

Resumo

Estimar com precisão matrizes de covariância a partir de amostras limitadas é um desafio fundamental na estatística de altas dimensões. Métodos tradicionais tentam mitigar os efeitos da escassez de dados por meio de *priors* estatísticos fixos, como regularização ou *shrinkage*, que podem não capturar a verdadeira estrutura subjacente dos dados. Esta dissertação introduz um novo paradigma que reformula a estimativa de covariância como um problema de remoção de ruído generativo. Conceituamos a Matriz de Covariância Amostral ruidosa ($\hat{\Sigma}$) como o resultado de um processo de degradação estatística não gaussiana e desenvolvemos um *framework* que aprende a reverter esse processo.

Nossa contribuição principal é o ColdCov, um novo modelo generativo profundo baseado no *framework* de Cold Diffusion. O ColdCov é particularmente adequado a esse problema por dois motivos. Primeiro, ele é projetado para reverter degradações arbitrárias não gaussianas, alinhando-se perfeitamente às propriedades estatísticas do processo de degradação. Segundo, sua arquitetura de saída impõe a propriedade simétrica positiva semi-definida (SPSD) por construção, garantindo que toda saída seja uma matriz de covariância matematicamente válida.

Por meio de experimentos abrangentes, demonstramos que o ColdCov proporciona vantagens significativas na reconstrução de covariância, particularmente em regimes de poucas amostras, alcançando menor erro de reconstrução de Frobenius e qualidade competitiva da matriz de precisão em relação a *baselines* estabelecidos. No entanto, os resultados também revelam uma compensação dependente da tarefa: métodos que impõem esparsidade, como o Graphical Lasso e Ledoit-Wolf, permanecem mais fortes para a recuperação explícita de estrutura de grafos, enquanto o ColdCov produz representações de covariância enriquecidas por remoção de ruído. A utilidade prática do *framework* é ainda validada no conjunto de dados de neuroimagem ABIDE, onde características de matriz de precisão extraídas pelo ColdCov apresentam o melhor desempenho médio observado na classificação discriminativa *downstream*, com ganhos modestos sobre os *baselines* em uma tarefa de classificação ao nível de sujeito de Controle *vs.* Autismo.

Concluimos que modelar a estimativa de covariância como uma tarefa de remoção de ruído generativa é um complemento promissor à regularização tradicional: é particularmente efi-

caz para reconstrução de covariância e extração de características discriminativas *downstream*, enquanto a regularização esparsa permanece preferível quando a recuperação exata da topologia do grafo é o objetivo primário.

Nomenclature

Variables and Symbols

n	Number of observations in a sample
p	Number of variables (dimensions)
x_i	A single data observation vector ($x_i \in \mathbb{R}^p$)
$\mathbf{x}_t$	The diffusion state matrix at timestep t
Σ	True covariance matrix
$\hat{\Sigma}$	Sample Covariance Matrix (SCM)
Θ	True precision matrix (Σ^{-1})
λ	Regularization parameter (e.g., in Graphical Lasso)
β	Coefficients of the Ordinary Least Squares
D	Statistical degradation operator

Acronyms

SCM	Sample Covariance Matrix
GGM	Gaussian Graphical Model
PDF	Probability Density Function
SPSD	Symmetric Positive Semi-Definite
OLS	Ordinary Least Squares
fMRI	Functional Magnetic Resonance Imaging
ABIDE	Autism Brain Imaging Data Exchange
GSP	Graph Signal Processing
HDLSS	High-Dimensional, Low Sample Size
ERM	Empirical Risk Minimization
ADNI	Alzheimer's Disease Neuroimaging Initiative
AP	Average Precision
AUC	Area Under the Curve
BIC	Bayesian Information Criterion
CNN	Convolutional Neural Network
DDPM	Denoising Diffusion Probabilistic Model
LDA	Linear Discriminant Analysis
MLE	Maximum Likelihood Estimation
NKL	Normalized Kullback-Leibler Divergence
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic
ROI	Region of Interest

List of Figures

1.1	The estimation pipeline explored in this thesis, illustrated for functional brain connectivity. Multivariate time series from p functional brain regions over n observations (left) are processed by a Cold Diffusion framework to recover the precision matrix Σ^{-1} (center), from which the conditional dependency graph G over brain regions (right) is inferred. Accurately recovering the true connectivity structure (top, marked “?”) from limited samples is the central challenge addressed by this work.	2
1.2	The statistical degradation process. Each panel shows a covariance matrix estimate $\hat{\Sigma}$ computed from n samples drawn from the same population. Reading left to right, the sample size n decreases while the number of variables p remains fixed. In the leftmost panel ($n \gg p$), the estimate closely resembles the true covariance: off-diagonal block structure is clearly visible and diagonal entries are well-calibrated. As n decreases, off-diagonal correlations fade, diagonal dominance increases, and the characteristic block patterns dissolve into noise. In the rightmost panel ($n \ll p$), the matrix is rank-deficient and virtually all structural information has been lost. Our method learns to reverse this trajectory.	4
2.1	Example of spurious correlation (Laska e Narayan, 2017): A fire causes both smoke and heat. While smoke and heat are strongly correlated, neither directly causes the other; they are conditionally independent given the presence of the fire.	8
2.2	An illustration of the Pairwise, Local, and Global Markov properties on a 5-node graph. From Laska e Narayan (2017).	8
2.3	The equivalence between zero entries in the precision matrix Θ and the absence of edges in the corresponding graph, indicating conditional independence (Laska e Narayan, 2017).	9

2.4	The visual duality of the Graph Learning problem. Identifying edges in the graph (left) is structurally equivalent to identifying non-zero off-diagonal entries in the Precision Matrix (right). Missing edges correspond to zeros (conditional independence).	11
2.5	Representative domains where Graphical Models and sparse covariance estimation find application. From top to bottom: genomics (DNA sequences), proteomics (protein structures), financial markets, and neuroscience (cortical parcellation). In each domain, p interacting variables produce multivariate time series of n observations (center), from which a dependency network (right) can be inferred via the estimated precision matrix. Image credits: biomolecule by D. Goodsell (CC BY 4.0); financial data by L. Villa del Campo (CC BY 2.0); brain parcellation courtesy of the Human Connectome Project.	12
2.6	Functional brain connectivity as a graphical model. Brain regions ($R_1, \dots, R_7$) are represented as nodes, with edges encoding conditional dependencies estimated from fMRI time series via the precision matrix $\hat{\Theta}$. Solid lines indicate strong conditional dependencies ($\hat{\Theta}_{ij} \neq 0$); dashed lines indicate weaker connections. Zeros in $\hat{\Theta}$ correspond to conditionally independent region pairs. .	13
2.7	The DeepGraph architecture, a supervised CNN that maps a sample covariance matrix to an estimated graph structure (Belilovsky et al., 2017).	15
2.8	Schematic of the U-Net architecture (Ronneberger et al., 2015). The encoder (left, blue) progressively downsamples the input through convolutional and pooling layers, capturing increasingly abstract features. The decoder (right, red) upsamples back to the original resolution via transposed convolutions. Skip connections (dashed arrows) concatenate encoder feature maps to corresponding decoder layers, preserving fine-grained spatial details lost during downsampling.	16
2.9	The DDPM framework (Ho et al., 2020). The forward process (blue, top) gradually corrupts a clean data sample $\mathbf{x}_0$ by adding Gaussian noise over T timesteps until only isotropic noise remains at $\mathbf{x}_T$. The learned reverse process (red, bottom) uses a neural network to iteratively denoise, recovering the original data from noise.	18
2.10	The Cold Diffusion framework (Bansal et al., 2023). The forward degradation (orange, top) applies an arbitrary, potentially non-Gaussian operator D that progressively corrupts a clean sample $\mathbf{x}_0$. The learned restoration (teal, bottom) iteratively recovers the original data using a neural network R_θ , without requiring Gaussian noise assumptions.	19
3.1	Regularization path for Graphical Lasso: as λ increases, the estimated precision matrix becomes progressively sparser (Laskas e Narayan, 2017).	29

3.2	The Step-Graph algorithm progressively refines the graph structure through local updates (Zamar et al., 2021).	31
3.3	Taxonomy of covariance estimation methods reviewed in this chapter. Statistical methods impose fixed mathematical priors, while machine learning methods learn from data. The generative denoising approach (this thesis) fills the identified gap.	33
3.4	Schematic illustration of how different estimation paradigms affect precision matrix structure. (a) Shrinkage produces a dense, regularized matrix. (b) Sparsity-based methods produce matrices with explicit zero entries. (c) Learned approaches produce data-adaptive structures that can capture complex patterns without imposing a rigid prior.	33
4.1	The synthetic data generation and evaluation pipeline. A sparse precision matrix Θ generates a true covariance Σ_{true} . Data X is sampled from $\mathcal{N}(0, \Sigma_{true})$, used to compute estimates $\hat{\Sigma}$, which are then quantitatively compared against the ground truth.	39
4.2	The Cold Diffusion sampling schedule $n(t)$ over $T = 1,000$ diffusion steps. The geometric decay from $N_{\max} = 32p$ to $N_{\min} = \lfloor p/4 \rfloor$ distributes timesteps uniformly on the logarithmic scale. The orange marker at $t = 0$ denotes the population limit ($n = \infty$), where the degradation operator returns the true covariance Σ_{true} .	44
4.3	The deterministic statistical degradation process. The operator $D(\Sigma_{true}, t)$ generates a degraded matrix by simulating the sampling process ($X \sim \mathcal{N}(0, \Sigma_{true})$) with sample size n_t .	44
4.4	The architectural SPSD constraint block. This differentiable module transforms the raw U-Net output into a guaranteed valid symmetric positive semi-definite correlation matrix.	45
4.5	The iterative training loop detailed in a horizontal flow. The model learns to map degraded matrices $\hat{\Sigma}_t$ back to the clean ground truth Σ_{true} by minimizing the empirical risk on synthetic data.	45

5.1	Impact of the logarithmic sampling ratio $k = \log_2(N/p)$ on covariance estimation quality. Left: Frobenius norm error between the sample covariance matrix and the true covariance, showing exponential decay with increasing k . Right: ROC AUC for graph structure recovery (link prediction), saturating rapidly beyond the invertibility limit. Vertical dashed lines mark the three critical thresholds: the Wishart invertibility boundary ($k = 0$, $N = p$), the Pareto knee ($k = 3$, $N = 8p$), and the selected operating point ($k = 5$, $N = 32p$). Results are consistent across all tested dimensions; each curve in the legend corresponds to a different matrix dimension ($p = 40, 80, 120, 160$).	52
5.2	Frobenius Norm reconstruction error as a function of sample size. ColdCov achieves the lowest error from $n = 20$ onward, maintaining its advantage across both the sub-sampled and over-sampled regimes.	55
5.3	Normalized KL Divergence as a function of sample size. ColdCov achieves the lowest divergence from $n = 20$ through $n = 160$, indicating strong precision matrix quality across the sub-sampled and moderate regimes.	56
5.4	ROC AUC for graph structure recovery via partial correlations. Sparse-enforcing methods (Graphical Lasso, Ledoit-Wolf) lead at all sample sizes, with ColdCov consistently outperforming StepGraph and DeepGraph.	57
5.5	Average Precision (AP) for graph structure recovery via partial correlations. AP is sensitive to performance on the sparse set of true edges, confirming the advantage of methods that produce well-calibrated precision matrices.	58
5.6	Qualitative comparison of precision matrix reconstruction across four graph topologies (rows: Random, Small-World, Lattice, Preferential Attachment) and five estimation methods (columns: Empirical, Ledoit-Wolf, Graphical Lasso, ColdCov, True). Experimental conditions: $p = 40$, $n = 40$, seed = 42. The Empirical sample inverse suffers from pervasive noise at this sample-to-dimension ratio ($n/p = 1$). Ledoit-Wolf applies shrinkage toward the diagonal, dampening off-diagonal noise but losing structural detail. Graphical Lasso promotes sparsity via ℓ_1 regularization, recovering some off-diagonal entries while suppressing weaker edges. ColdCov applies the SPSPD-constrained cold diffusion model to denoise the sample covariance, recovering off-diagonal sparsity patterns that visually approximate the true precision matrix.	60
5.7	Confusion matrices for the Control vs. ASD classification task across four estimation methods. ColdCov shows the most balanced classification in this representative fold, with higher true positive rates for both diagnostic groups. StepGraph, by contrast, shows strong bias toward the majority class.	64

5.8	Estimated functional brain connectomes for a representative subject, visualized using the Nilearn glass brain projection. Each method reveals different topological characteristics: Graphical Lasso produces a sparse network with few strong connections, Ledoit-Wolf yields a dense network, StepGraph identifies a minimal set of edges, and ColdCov recovers a rich yet structured connectivity pattern with distributed inter-regional connections.	65
5.9	Pareto efficiency of connectivity estimators on the ABIDE dataset, plotting sparsity (fraction of near-zero precision matrix entries, higher is better) against log-likelihood (higher is better). Graphical Lasso achieves the highest sparsity (≈ 0.51) with moderate log-likelihood; Ledoit-Wolf attains the best log-likelihood (≈ -45) but near-zero sparsity; ColdCov occupies an intermediate position (≈ 0.13 sparsity, ≈ -70 log-likelihood), demonstrating that its precision matrices are not fully dense but rather exhibit moderate sparsity while retaining rich conditional dependency structure.	66
5.10	Visual analysis of the degradation residuals for $p = 40$ and $n/p = 2$. While visually Gaussian, statistical tests confirm latent non-Gaussianity.	67
5.11	Convergence of p-values from three normality tests (Kolmogorov-Smirnov, D'Agostino-Pearson, Shapiro-Wilk) as the number of residual elements sampled increases from 100 to 10,000.	68
5.12	Ablation study: Frobenius Norm reconstruction error. The Unconstrained variant produces catastrophically invalid outputs ($> 1000\times$ error), confirming that the SPSD constraint is essential, not merely beneficial.	69
5.13	Ablation study: Normalized KL Divergence. The Unconstrained variant remains near total divergence (≈ 0.98) at all sample sizes, while both SPSD variants improve steadily as data increases, with the Custom Loss variant marginally ahead.	70
5.14	Ablation study: ROC AUC for graph structure recovery. Both SPSD variants achieve meaningful link prediction, while the Unconstrained variant performs at random-chance level (≈ 0.50).	71
B.1	Precision matrix heatmaps for a representative ABIDE subject across four estimation methods. Graphical Lasso produces a sparse matrix with concentrated strong entries, Ledoit-Wolf yields a dense matrix with attenuated off-diagonal values, StepGraph identifies a minimal set of non-zero entries, and ColdCov produces a structured matrix with distributed conditional dependencies. . . .	91

B.2 PCA+LDA 2D projections of precision matrix features for the ABIDE dataset. Each point represents a subject (blue: Control, orange: ASD), with covariance ellipses showing within-class distributions. ColdCov features exhibit the greatest inter-class separation, consistent with its best observed mean classification performance.	92
--	----

List of Tables

3.1	Application domains where covariance estimation plays a critical role. Brain connectivity is selected as the primary domain for this thesis due to the inherent $p \approx n$ regime and clinical significance.	27
3.2	Comparison of covariance and precision matrix estimation methods reviewed in this chapter. The final row highlights the gap addressed by the proposed ColdCov framework.	34
3.3	Feature comparison of estimation methods. The proposed ColdCov framework is the only method to combine data-driven degradation modeling with full matrix estimation and an SPSD guarantee, though it shares DeepGraph's reliance on synthetic training data.	35
3.4	Software packages commonly used for covariance and precision matrix estimation. Packages marked with $\dagger$ are used in this thesis.	36
5.1	Critical thresholds identified from the sampling ratio analysis. These thresholds are dimension-independent, observed consistently across all tested values of p	52
5.2	Frobenius Norm ($\ \hat{\Sigma} - \Sigma\ _F$) Comparison (Lower is better)	55
5.3	Normalized KL Divergence (Lower is better)	56
5.4	Link Prediction Performance (ROC AUC and Average Precision) via Partial Correlations across Sample Sizes ($n \in \{10, 40, 160\}$)	57
5.5	Subject-level classification performance on the ABIDE dataset (870 subjects, 39 ROIs). All metrics are computed via 5×5 Repeated Stratified Cross-Validation (25 folds) with L2 Logistic Regression. Higher is better. Best values in bold	62

5.6	Wilcoxon signed-rank test results comparing ColdCov (Optimized) against each baseline across repeated 5×5 stratified cross-validation (25 paired observations per comparison). p -values are Holm–Bonferroni corrected for 9 simultaneous tests. Significance: $**p < 0.01$, $*p < 0.05$, ns = not significant. . . .	63
5.7	Mean and standard deviation of p -values from normality tests at representative residual sample sizes (100 and 10,000 elements).	68
5.8	Ablation Study: Key Metrics at Representative Sample Sizes	69
B.1	Best Estimators for Frobenius Norm Reconstruction Error across Topologies and Sample Sizes.	89
B.2	Best Estimators for Link Prediction Metrics (Recall vs Precision) across Sample Sizes (n).	90

Sumário

Abstract	v
Resumo	vii
Nomenclature	ix
List of Figures	xi
List of Tables	xvii
1 Introduction	1
1.1 The Curse of Dimensionality and the Role of Sparsity	2
1.2 From HDLSS to Statistical Degradation	3
1.3 Covariance Estimation as a Generative Process	4
1.4 Research Goals	5
1.5 Organization of the Thesis	6
2 Background	7
2.1 Foundations of Graphical Models	7
2.1.1 The interpretation of Θ	9
2.2 Graph Signal Processing and Graph Learning	10
2.3 Applications of Sparse Covariance Estimation	11
2.4 The Challenge of High-Dimensional Covariance Estimation	13

2.4.1	The Sample Covariance Matrix and Its Limitations	13
2.5	Deep Learning Architectures for Denoising	14
2.5.1	Supervised Learning with Convolutional Neural Networks	15
2.5.2	The U-Net Architecture for Structured Denoising	15
2.6	Generative Denoising with Diffusion	16
2.6.1	Denoising Diffusion Probabilistic Models (DDPMs)	17
2.6.2	Cold Diffusion: A Generalization for Arbitrary Degradations	18
2.7	Assessing Data Normality	19
2.7.1	Graphical Analysis	20
2.7.2	Statistical Hypothesis Testing	20
2.8	Metrics for Evaluating Covariance Estimation	21
2.8.1	Quality of Covariance and Precision Matrix Estimates	21
2.8.2	Metrics for Graph Structure Recovery	22
2.9	Chapter Summary	24
3	Literature Review	25
3.1	Application Domain: Functional Brain Connectivity	25
3.1.1	Alternative Application Domains	26
3.2	A Review of Estimation Methods	27
3.2.1	Paradigm 1: Statistical Regularization Methods	28
3.2.2	Alternative Machine Learning Approaches	30
3.2.3	Comparative Synthesis of Estimation Methods	33
3.3	Identifying the Research Gap: From Regularization to Generative Denoising .	35
3.4	Chapter Summary	37
4	A Generative Denoising Framework for Covariance Estimation	38
4.1	Data Generation and Experimental Pipeline	38
4.1.1	Synthetic Data Generation	39
4.1.2	Evaluation Strategy for Synthetic Data	39
4.1.3	Real-World Datasets and Preprocessing	40

4.1.4	Evaluation Strategy for Real-World Data	40
4.2	The ColdCov Framework	41
4.2.1	Modeling Statistical Degradation for Cold Diffusion	42
4.2.2	Constrained Architecture for Valid Covariance Matrices	44
4.2.3	Model Training	45
4.2.4	Denoising and Inference	46
4.2.5	Computational Complexity	46
4.3	Baseline Method Implementation	47
4.3.1	Ledoit-Wolf Shrinkage	47
4.3.2	Graphical Lasso	48
4.3.3	StepGraph	48
4.3.4	DeepGraph	48
4.4	Chapter Summary	48
5	Experiments and Results	50
5.1	Experimental Setup	50
5.1.1	Datasets and Degradation Schedule	50
5.1.2	Implementation Details of Proposed Models	53
5.1.3	Implementation Details of Baseline Methods	53
5.2	Quantitative Analysis on Synthetic Data	54
5.2.1	Covariance Reconstruction Fidelity	54
5.2.2	Analysis of Precision Matrix Quality	55
5.2.3	Graph Topology and Link Prediction	57
5.2.4	Topology-Specific Performance Analysis	59
5.3	Qualitative Analysis of Matrix Reconstruction	59
5.4	Application to Real-World Data: A Neuroimaging Case Study	61
5.4.1	Discriminative Classification Evaluation	62
5.4.2	Classification Results	62
5.4.3	Visual Analysis of Estimated Brain Connectivity	64

5.5	Validating the Non-Gaussian Degradation Assumption	67
5.6	Ablation Studies: Impact of Model Components	69
5.7	Discussion of Results	71
6	Conclusion	73
6.1	Conceptual, Methodological, and Empirical Contributions	73
6.2	Revisiting the Central Hypothesis	74
6.3	Limitations and Challenges	75
6.3.1	Dependency on the Training Prior	75
6.3.2	Computational Cost of Inference	76
6.3.3	Dependency on Fixed Input Dimensions	76
6.3.4	The “Black Box” Nature of Deep Learning	77
6.3.5	Scope of the Current Evaluation	77
6.4	Future Research	78
6.4.1	Advanced Iterative Inference Strategies	78
6.4.2	Validation on Non-Gaussian Data Distributions	79
6.4.3	Flexible Architectures for Arbitrary Input Dimensions	80
6.4.4	Scalability to Ultra-High Dimensions ($p > 1000$)	80
6.4.5	Theoretical Guarantees and Stability Analysis	81
6.5	Concluding Remarks	81
A	Mathematical Foundations	83
A.1	Geometry of the SPSD Cone	83
A.1.1	Cholesky Construction Validity	83
A.1.2	Convexity of the Cone	84
A.2	Statistical Properties of the Degradation Process	84
A.2.1	The Wishart Distribution	84
A.2.2	Marchenko-Pastur Law (Eigenvalue Spreading)	84
A.3	Derivation of Evaluation Metrics	85
A.3.1	KL Divergence for Multivariate Gaussians	85

A.3.2 Gaussian Log-Likelihood	85
A.4 Computational Complexity of Attention Mechanisms	86
A.4.1 Quadratic Sequence Length and the $O(p^4)$ Bound	86
A.4.2 Linear Attention and Mitigation Strategies	86
B Supplementary Experimental Results	88
B.1 Reconstruction Performance by Topology	88
B.2 Link Prediction Performance	89
B.3 Supplementary ABIDE Visualizations	90
B.3.1 Precision Matrix Heatmaps	90
B.3.2 PCA+LDA 2D Projections	91
C Core Algorithms and Pseudo-Code	93
C.1 Statistical Degradation (Wishart Sampling)	93
C.2 Cold Diffusion Training Loop	93

Introduction

Graphical Models (GMs) provide a powerful framework for understanding complex systems by combining graph theory and probability theory to model probabilistic relationships. GMs find broad application across diverse scientific domains, including systems biology, neuroscience, finance, and statistical signal processing ([Diskin e Wiesel, 2024](#); [Manolakis et al., 2013](#); [Nasrabadi, 2013](#)). In many of these domains, models operate in high-dimensional settings where the number of variables is large relative to the available data, making accurate covariance estimation both central and challenging. At the heart of such models lies the covariance matrix, a fundamental statistical tool that quantifies the structure of dependencies between variables, and its inverse, the precision matrix $\Theta = \Sigma^{-1}$, which directly encodes conditional independence under the multivariate Gaussian assumption.

Among these applications, functional brain connectivity provides a particularly compelling illustration of this challenge. As depicted in [Figure 1.1](#), neuroscientists measure activity across p functional brain regions and seek to infer the underlying network of conditional dependencies between them. This inference requires estimating the precision matrix Σ^{-1} from limited observations and then extracting the connectivity graph G . Accurate covariance estimates are therefore critical, forming the foundation on which such essential downstream analyses are built.

However, in all real-world applications, the true covariance matrix Σ of a multivariate random variable X is unknown and must be estimated from a finite number of samples. This estimation step represents the single most critical challenge in the entire analysis pipeline, as the quality of all subsequent inferences, from network discovery to predictive modeling, fundamentally depends on the accuracy of this initial estimate.

In classical statistical settings, where the number of observations n vastly exceeds the number of variables p ($n \gg p$), the Sample Covariance Matrix (SCM) acts as an effective, unbiased, and well-behaved estimator. By the Law of Large Numbers, the SCM converges to the true population covariance as $n \rightarrow \infty$.

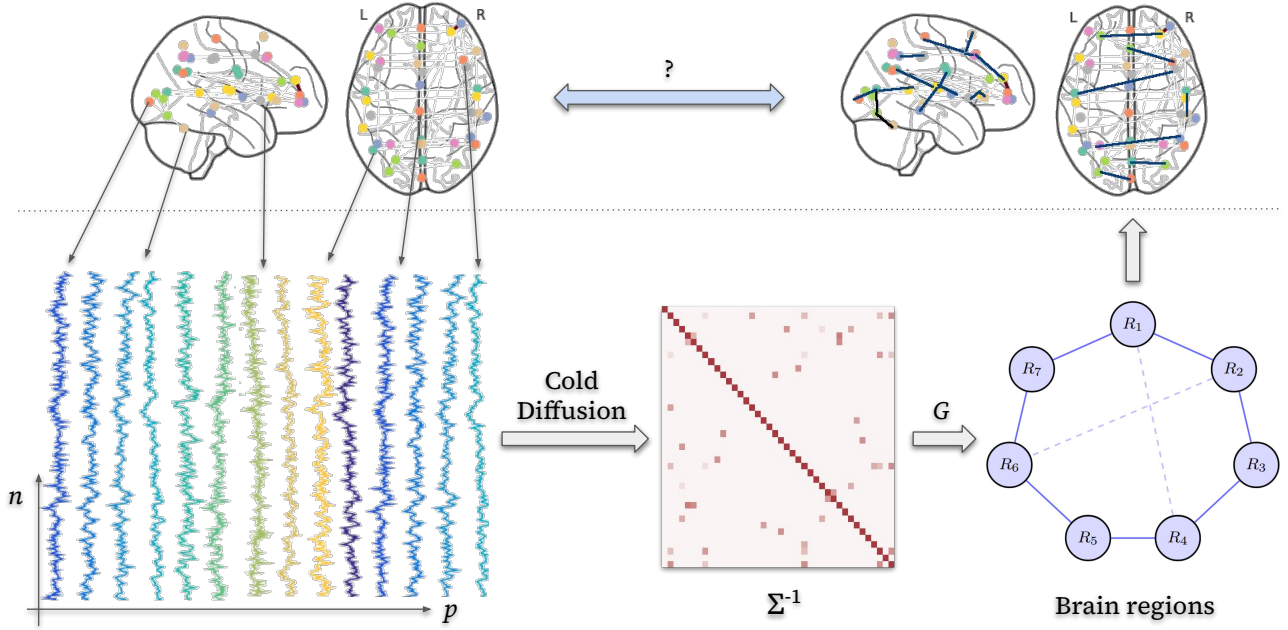


Figure 1.1: The estimation pipeline explored in this thesis, illustrated for functional brain connectivity. Multivariate time series from p functional brain regions over n observations (left) are processed by a Cold Diffusion framework to recover the precision matrix Σ^{-1} (center), from which the conditional dependency graph G over brain regions (right) is inferred. Accurately recovering the true connectivity structure (top, marked "?") from limited samples is the central challenge addressed by this work.

1.1 The Curse of Dimensionality and the Role of Sparsity

In modern statistical learning problems, it is increasingly common for the number of variables to greatly exceed the number of observations, rendering classical estimators unreliable or ill-posed (Donoho, 2000; Hastie et al., 2009). In fields ranging from genomics (where p may be 20,000 genes) to neuroimaging (where p may be hundreds of brain regions), the number of variables often approaches or far exceeds the number of available samples ($n \leq p$, then $\hat{\Sigma} \neq \Sigma$). In this regime, the SCM fails catastrophically due to two interconnected phenomena. These phenomena are a manifestation of the well-known curse of dimensionality (Bellman, 1961).

First, singularity arises when $n < p$, causing the SCM to be rank-deficient (rank at most $n - 1$) and thus singular (Anderson, 2003). This rank deficiency renders standard graphical modeling techniques mathematically impossible without regularization, as the matrix cannot be inverted to obtain the precision matrix Θ . Second, even when $p \approx n$, the SCM ($\hat{\Sigma}$) suffers from severe noise-induced distortion known as eigenvalue spreading. Random Matrix Theory, specifically the Marchenko-Pastur law (Marchenko e Pastur, 1967), dictates that the eigenvalues of the SCM will spread out, systematically underestimating small eigenvalues and overestimating large ones, thereby obscuring the true eigenstructure of the data. Consequently, the high-dimensional SCM is not just a noisy estimate; it is a fundamentally unreliable estimator that requires sophisticated restoration to become usable.

The scenario in which $n \ll p$ is formally recognized in statistics and machine learning as the High-Dimensional, Low Sample Size (HDLSS) problem (Hall et al., 2005). In this context, sparsity is not merely a theoretical construct but a mathematical necessity to ensure model identifiability and interpretability. Within this framework, a parameter vector is considered sparse if the majority of its elements are zero.

When the dimensionality of the feature space exceeds the sample size, the underlying linear system is underdetermined. This leads to two critical issues: the existence of infinite solutions and extreme overfitting. Regarding the former, traditional estimation methods, such as Ordinary Least Squares (OLS), fail because the empirical covariance matrix is singular and non-invertible. Consequently, there are infinite combinations of coefficients (β) that can achieve zero training error, rendering the solution unstable. Regarding the latter, in an HDLSS setting, the model possesses enough degrees of freedom to perfectly interpolate the noise within the training data. This results in poor generalization performance, as the system cannot distinguish between true underlying patterns and random fluctuations.

The sparsity hypothesis emerges as a solution to render the $n \ll p$ problem tractable. This hypothesis assumes that although the data resides in a high-dimensional space (p), the output is governed by only a small subset of active predictors ($s \ll p$), where $s < n$. By imposing sparsity, the effective dimensionality of the problem is reduced from p to s , transforming an ill-posed problem into a well-posed one. Through techniques such as L_1 regularization (LASSO (Tibshirani, 1996)), the model performs simultaneous feature selection and estimation, identifying the *true* variables within the high-dimensional noise.

Sparsity thus represents the classical approach to taming the HDLSS problem: it constrains the solution space by assuming that most interactions are absent. This thesis asks a different question. Rather than prescribing a fixed structural prior, can we learn the inverse of the statistical degradation directly from data, without imposing assumptions about which entries should be zero? The following sections develop this alternative perspective.

1.2 From HDLSS to Statistical Degradation

The HDLSS problem described above is conventionally treated as an estimation error to be corrected through regularization. This thesis proposes a more powerful perspective: to view the phenomenon not merely as an estimation error, but as a statistical degradation process. In this framework, the true, clean covariance matrix undergoes a systematic corruption when observed through the lens of HDLSS. This process is visualized in Figure 1.2.

This degradation can be formally understood as an ill-posed inverse problem. The *forward* process is the generation of a rank-deficient empirical covariance matrix from a full-rank population covariance matrix via random sampling. Concretely, a small number of observations are drawn from a distribution with true covariance Σ , and the covariance matrix is then estimated from this limited sample. This process destroys information, specifically the

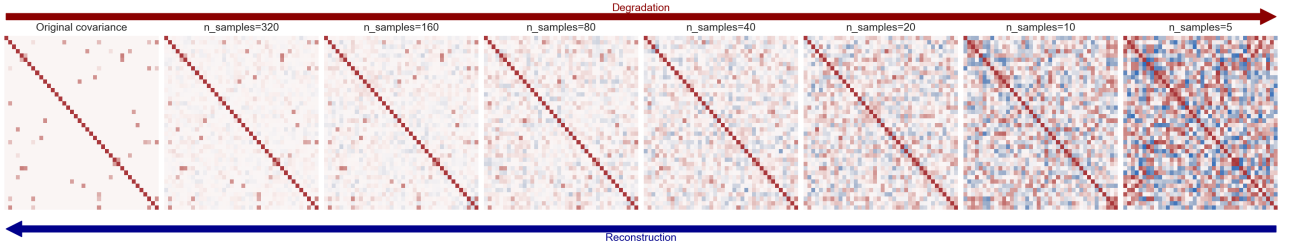


Figure 1.2: The statistical degradation process. Each panel shows a covariance matrix estimate $\hat{\Sigma}$ computed from n samples drawn from the same population. Reading left to right, the sample size n decreases while the number of variables p remains fixed. In the leftmost panel ($n \gg p$), the estimate closely resembles the true covariance: off-diagonal block structure is clearly visible and diagonal entries are well-calibrated. As n decreases, off-diagonal correlations fade, diagonal dominance increases, and the characteristic block patterns dissolve into noise. In the rightmost panel ($n \ll p$), the matrix is rank-deficient and virtually all structural information has been lost. Our method learns to reverse this trajectory.

variance associated with directions orthogonal to the low-dimensional subspace spanned by the n samples. The corresponding inverse problem, recovering the full-rank covariance structure from the singular empirical estimate, is theoretically impossible without the incorporation of prior knowledge or structural assumptions.

The degradation follows a well-defined statistical law (the Wishart distribution for Gaussian data). This insight motivates our novel solution approach: instead of using generic mathematical regularizers (like L1 penalties) that impose rigid, handcrafted priors, we can use deep generative models to learn the inverse mapping of this degradation process. By training a model to reverse the specific structural damage induced by sampling under severe data scarcity (i.e., $n \ll p$), we hypothesize that it is possible to recover a plausible, high-fidelity covariance matrix that preserves the complex dependency structure of real-world data variables.

1.3 Covariance Estimation as a Generative Process

Existing methods attempt to mitigate the effects of this statistical degradation through regularization. Techniques like Graphical Lasso impose sparsity assumptions to reduce complexity, while shrinkage estimators pull the SCM towards a more stable target (Friedman et al., 2008). While effective, these methods rely on fixed statistical priors that may not capture the true, complex structure of the data. Supervised deep learning approaches, such as DeepGraph (Belilovsky et al., 2017), have also been explored but face key limitations: their convolution-based architectures implicitly assume a local or spatial ordering among variables, and they are inherently limited to estimating the graph adjacency matrix rather than the full covariance or precision matrices required for many downstream tasks.

This thesis introduces a fundamentally different paradigm that represents a conceptual departure from traditional approaches. Rather than indirectly compensating for data scarcity through regularization or shrinkage, we tackle the problem directly by reframing covari-

ance estimation as an explicit generative denoising task. We conceptualize the empirical SCM as a *noisy observation* of the true covariance matrix, analogous to a corrupted image, and develop a deep generative model specifically designed to reverse the statistical degradation process and recover the clean, underlying Symmetric Positive Semi-Definite (SPSD) structure.

This research is motivated by the central hypothesis:

By explicitly modeling finite-sample covariance estimation as a statistical degradation process, a deep generative framework can learn a data-driven prior over valid covariance matrices and improve covariance reconstruction and downstream discriminative performance in high-dimensional, low-sample-size regimes, while providing competitive graph-structure recovery compared with traditional regularization-based estimators.

Here, a *powerful prior* denotes a data-driven probabilistic model that captures the statistical structure and constraints of valid covariance matrices, learned from representative data rather than imposed through simple parametric assumptions. Unlike classical priors or regularization schemes, such a prior encodes rich, high-order dependencies and provides strong guidance in data-scarce, high-dimensional settings.

1.4 Research Goals

The primary objective of this research is to develop and validate a novel deep generative framework for high-dimensional covariance estimation designed to explicitly reverse the effects of data scarcity.

The specific objectives are fourfold:

1. To demonstrate the viability of framing high-dimensional covariance estimation as a denoising task by developing a direct supervised model based on a U-Net architecture.
2. To formally characterize the statistical degradation process, analytically showing that degrading noise is inherently non-Gaussian, thereby motivating the requirement for advanced generative modeling.
3. To design and implement ColdCov, a generative model that utilizes a Constrained Cold Diffusion framework to learn the iterative reversal of this non-Gaussian degradation.
4. To conduct a rigorous quantitative and qualitative evaluation comparing the proposed generative denoising approach with established baselines, such as Graphical Lasso, across three distinct objectives: covariance reconstruction, graph-structure recovery, and downstream classification. This evaluation aims to assess the performance of ColdCov particularly in challenging HDLSS regimes.

1.5 Organization of the Thesis

The remainder of this thesis is organized to build the case for our generative denoising approach systematically.

Chapter 2 establishes the theoretical foundations by reviewing Gaussian Graphical Models, the mathematical properties of covariance matrices, and the deep learning architectures that enable our framework.

Chapter 3 critically examines the existing literature, identifying the conceptual gap that our work addresses and positioning our contribution within the broader landscape of covariance estimation methods.

Chapter 4 presents our methodological contributions, detailing both the direct denoising approach and our principal innovation, the ColdCov framework, which leverages Constrained Cold Diffusion to reverse non-Gaussian statistical degradation.

Chapter 5 provides comprehensive empirical validation through experiments on synthetic and real-world data, evaluating the strengths and limitations of our approach across covariance reconstruction, graph-structure recovery, and downstream classification, particularly in the challenging low-sample regime.

Finally, Chapter 6 synthesizes our findings, discusses their implications for the field, and charts promising avenues for future research. The appendices provide supporting material: Appendix A contains the formal mathematical proofs underpinning our architectural and evaluation choices, Appendix B provides supplementary experimental results including additional visualizations, and Appendix C details the core algorithms in pseudo-code form. Through this progression, we build a compelling case that generative denoising represents not just an alternative, but a promising and effective paradigm for high-dimensional covariance estimation.

Background

This chapter establishes the theoretical foundations necessary to understand the generative denoising approach to covariance estimation developed in this thesis. We begin with Gaussian Graphical Models (GGMs) and the role of the covariance and precision matrices in encoding conditional dependencies (Section 2.1), followed by the complementary perspective of Graph Signal Processing and Graph Learning (Section 2.2). We then present representative applications that motivate the need for accurate estimation in high-dimensional settings (Section 2.3). Next, we formally define the Sample Covariance Matrix, derive its rank-deficiency in the high-dimensional regime, and characterize the resulting estimation challenge (Section 2.4). The chapter then introduces the deep learning architectures that enable our solution: the U-Net for structured denoising and diffusion models, including the generalized Cold Diffusion framework for non-Gaussian degradations (Sections 2.5–2.6). We also present the statistical tools for assessing data normality (Section 2.7). Finally, we define the evaluation metrics used throughout this work (Section 2.8). Specific estimation methods and their detailed comparisons are reserved for the literature review in Chapter 3.

2.1 Foundations of Graphical Models

Graphical models (GM) combine graph theory and probability theory to create networks that model complex probabilistic relationships, finding applications in diverse domains such as systems biology, neuroscience, psychometrics, and finance (Laska e Narayan, 2017). Estimating the structure of these networks poses a significant statistical challenge. This section lays the groundwork for understanding sparse covariance estimation by defining Gaussian Graphical Models (GGMs), a framework for representing dependencies between variables in high-dimensional data. Central to this framework is the precision matrix Θ , the inverse of the covariance matrix ($\Theta = \Sigma^{-1}$), which encodes the network structure and encourages sparsity, a desirable property in real-world scenarios.

Fire

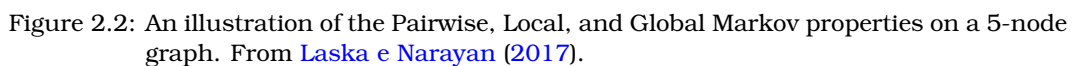
Smoke

Heat

..... "indirect" connection

— "direct" connection

This rationale underpins the use of Markov networks, which, in the context of Gaussian distributed data, are referred to as Gaussian Graphical Models. Formally, a graph $G = (V, E)$ consists of a set of vertices $V = \{1, \dots, p\}$ representing random variables and a set of edges $E \subset V \times V$. A probability distribution $\mathbb{P}_X$ forms a Markov network with a graph G if it satisfies the Markov properties, which link the graph’s structure to conditional independencies in the data (Lauritzen, 1996; Maathuis et al., 2018), as shown in Figure 2.2.



For multivariate Gaussian distributions, these properties are equivalent. The crucial connection to covariance estimation is established through the Hammersley-Clifford theorem (Maathuis et al., 2018), which reveals a fundamental equivalence: for a set of variables X following a multivariate Gaussian distribution $\mathcal{N}_p(0, \Sigma)$, a zero entry in the precision matrix, $\Theta_{ij} = 0$, is equivalent to the conditional independence of variables X_i and X_j given all

other variables. This means the sparsity pattern of the precision matrix directly reveals the absence of edges in the graphical model.

$$\Theta_{ij} = 0 \iff (i, j) \notin E \iff X_i \perp X_j \mid X_{V \setminus \{i, j\}} \quad (2.1)$$

In Equation 2.1, the symbol $\perp$ (inverted T) denotes conditional independence: $X_i \perp X_j \mid X_{V \setminus \{i, j\}}$ reads as "variable X_i is conditionally independent of variable X_j given all other variables in the system."

As illustrated in Figure 2.3, the structure of the precision matrix provides an efficient and powerful means of determining the structure of the underlying Markov network. This makes the accurate estimation of a sparse precision matrix Θ the primary goal for structure discovery in high-dimensional systems.

It is important to note that the equivalence between zero entries in the precision matrix and conditional independence holds strictly under the multivariate Gaussian assumption. For non-Gaussian data, the precision matrix should be interpreted as encoding conditional linear-dependence summaries rather than full conditional independence: a zero entry indicates the absence of a linear relationship after controlling for other variables, but nonlinear dependencies may still exist.

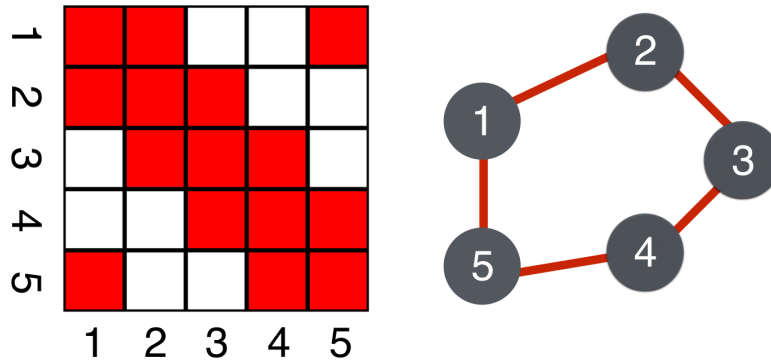


Figure 2.3: The equivalence between zero entries in the precision matrix Θ and the absence of edges in the corresponding graph, indicating conditional independence (Lasko e Narayan, 2017).

2.1.1 The interpretation of Θ

The interpretation of the precision matrix (Θ) is centered on conditional relationships rather than marginal ones¹ (Lauritzen, 1996; Hastie et al., 2009). The off-diagonal elements (θ_{ij} , where $i \neq j$) represent the conditional dependence between variables X_i and X_j , assuming all other variables in the system remain constant:

- If $\theta_{ij} = 0$, the variables are *conditionally independent* given the rest of the set. In a graphical model G , this implies the absence of a direct edge between nodes i and j .

¹Marginal relationships refer to the correlation or covariance between two variables calculated in isolation, disregarding the presence or influence of other variables in the system.

- If $\theta_{ij} \neq 0$, a direct relationship exists that cannot be explained by other variables.
- Due to the structure of the multivariate normal distribution, a negative θ_{ij} typically indicates a positive conditional correlation, and vice versa.

The precision matrix coefficients are also directly linked to the partial correlation ($\rho_{ij|rest}$), which quantifies the association between X_i and X_j after removing the linear effects of all other variables:

$$\rho_{ij|rest} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}}.$$

This provides an intuitive interpretation: a high absolute value indicates a strong connection between the two variables within the model. Similarly, the diagonal elements θ_{ii} correspond to the residual variance of variable X_i that remains unexplained by a linear combination of the other variables. Specifically, θ_{ii} is the reciprocal of the conditional variance:

$$\theta_{ii} = \frac{1}{\sigma^2(X_i | X_{rest})}.$$

Consequently, a high value of θ_{ii} suggests that X_i has a small conditional variance and is highly predictable based on the other variables in the system.

2.2 Graph Signal Processing and Graph Learning

While Gaussian Graphical Models provide the statistical framework for modeling dependencies, the field of **Graph Signal Processing (GSP)** offers a complementary perspective rooted in signal theory. GSP extends classical signal processing tools, such as Fourier transforms, filtering, and sampling, to data residing on irregular, non-Euclidean domains represented by graphs (Shuman et al., 2013; Leus et al., 2023).

In the standard GSP paradigm, the underlying graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ is typically assumed to be known *a priori*. For instance, in sensor networks, the edges are defined by physical proximity; in transportation networks, they follow physical routes. The goal is then to analyze signals $\mathbf{x} \in \mathbb{R}^N$ defined on the nodes $\mathcal{V}$ of this graph. In the GSP framework, the data values observed at graph nodes are referred to as *signals*; these correspond to the random variables of the GGM framework introduced in Section 2.1. However, a more challenging and increasingly relevant problem arises when the graph structure itself is unknown and must be inferred from the observed signals. This inverse problem is known as Graph Learning or Topology Inference (Mateos et al., 2019; Leus et al., 2023).

Graph Learning seeks to identify a graph structure (typically represented by a weighted adjacency matrix $\mathbf{W}$ or a Laplacian $\mathbf{L}$) that best explains the observed data $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_M]$. A fundamental assumption in this domain is that the signals are *smooth* with respect to the underlying graph, meaning that values at connected nodes are likely to be similar. This

smoothness can be quantified using the graph Laplacian quadratic form:

$$\text{Smoothness}(\mathbf{x}) = \mathbf{x}^\top \mathbf{L} \mathbf{x} = \sum_{(i,j) \in \mathcal{E}} \mathbf{W}_{ij} (x_i - x_j)^2. \quad (2.2)$$

Minimizing this quantity leads to graph structures where connected nodes have correlated signal values, creating a direct bridge to covariance estimation.

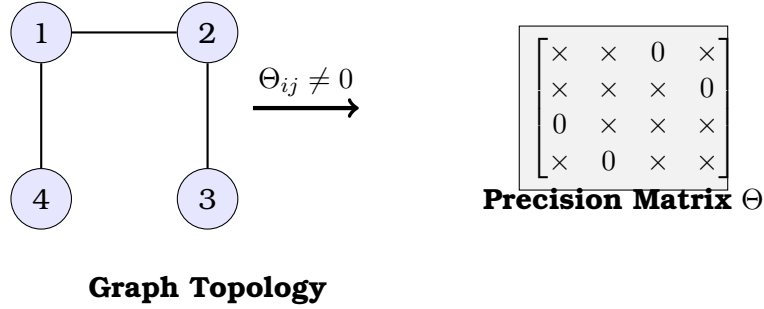


Figure 2.4: The visual duality of the Graph Learning problem. Identifying edges in the graph (left) is structurally equivalent to identifying non-zero off-diagonal entries in the Precision Matrix (right). Missing edges correspond to zeros (conditional independence).

The connection between Graph Learning and Gaussian Graphical Models is profound. As noted in the comprehensive review by [Leus et al. \(2023\)](#), many modern approaches to graph learning can be viewed as identifying a mapping from the data covariance to a graph shift operator. Specifically, for Gaussian Markov Random Fields (GMRFs), the problem of learning the graph topology is mathematically equivalent to estimating the sparsity pattern of the precision matrix $\Theta = \Sigma^{-1}$.

In this context, the precision matrix acts as a generalized Laplacian. Non-zero off-diagonal entries $\Theta_{ij} \neq 0$ correspond directly to edges in the graph, representing conditional dependencies (partial correlations) given all other nodes. Thus, the statistical task of Sparse Inverse Covariance Estimation, the focus of this thesis, is fundamentally a Graph Learning problem. By accurately estimating Θ from limited data, we are effectively learning the underlying graph topology that governs the generation of the observed signals.

This GSP perspective reinforces the importance of the generative denoising approach adopted in this thesis. Traditional Graph Learning methods often rely on convex optimization with smoothness priors (like Graphical Lasso). By viewing the problem instead as learning a generative prior over valid graph topologies (via Cold Diffusion), we can potentially capture more complex structural dependencies than simple smoothness assumptions allow.

2.3 Applications of Sparse Covariance Estimation

The practical value of Gaussian Graphical Models becomes especially evident in high-dimensional scientific domains such as functional brain connectivity and genetic analysis

(Maathuis et al., 2018; Laska e Narayan, 2017; Watts e Strogatz, 1998). These fields are characterized by complex systems with a vast number of interacting components, making the accurate estimation of sparse dependency structures a critical objective. As illustrated in Figure 2.5, each of these domains produces high-dimensional multivariate data from which dependency networks can be inferred via the estimated precision matrix.

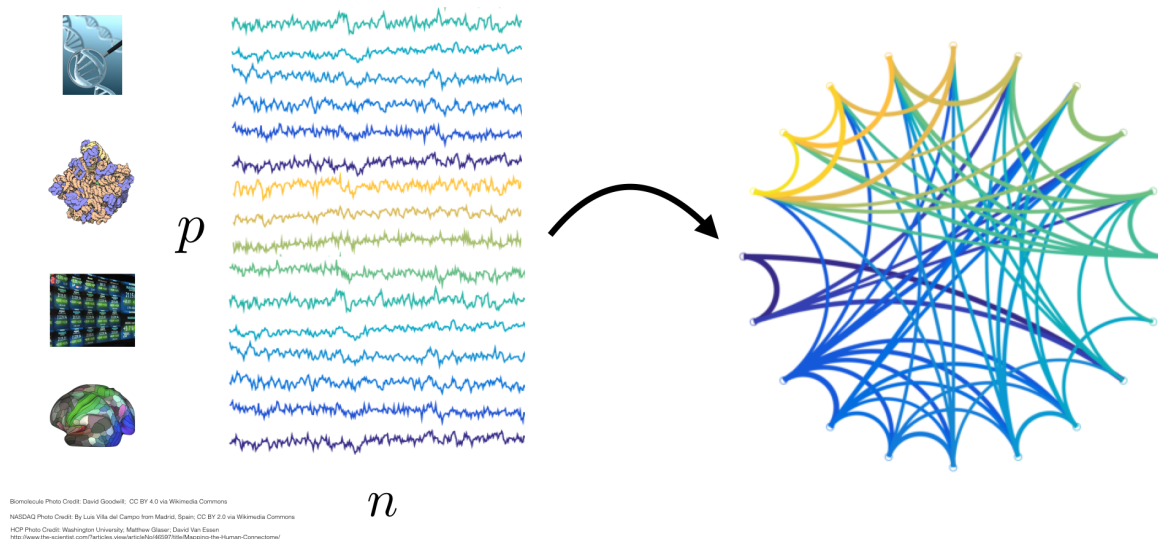


Figure 2.5: Representative domains where Graphical Models and sparse covariance estimation find application. From top to bottom: genomics (DNA sequences), proteomics (protein structures), financial markets, and neuroscience (cortical parcellation). In each domain, p interacting variables produce multivariate time series of n observations (center), from which a dependency network (right) can be inferred via the estimated precision matrix. Image credits: biomolecule by D. Goodsell (CC BY 4.0); financial data by L. Villa del Campo (CC BY 2.0); brain parcellation courtesy of the Human Connectome Project.

In neuroscience, GGMs are a valuable tool for understanding the intricate network of functional connections within the human brain. Functional Magnetic Resonance Imaging (fMRI) provides a noninvasive method for measuring blood oxygen level-dependent (BOLD) signals, which act as a proxy for neuronal activity. By applying GGMs to fMRI data, researchers can estimate the precision matrix to capture the conditional dependencies between different brain regions (Bullmore e Sporns, 2009). This process is illustrated in Figure 2.6: brain regions are parcellated into nodes, and the precision matrix estimated from their time series reveals the functional connectivity network. This enables the identification of large-scale functional networks and allows for the study of how these networks are altered in neurological disorders such as Alzheimer’s disease or autism (Sporns e Betzel, 2016). Prominent datasets like the Autism Brain Imaging Data Exchange (ABIDE) have been instrumental in facilitating this line of research (Di Martino et al., 2014).

In the field of genetics, GGMs play a crucial role in deciphering the complex regulatory relationships between genes. Gene expression data, obtained from technologies like RNA sequencing, provide a snapshot of the activity levels of thousands of genes simultaneously. By estimating the precision matrix from this data, researchers can construct gene regulatory networks that illustrate how genes interact to control biological processes (Emmert-Streib et al., 2014). Identifying key hubs and pathways in these networks can provide

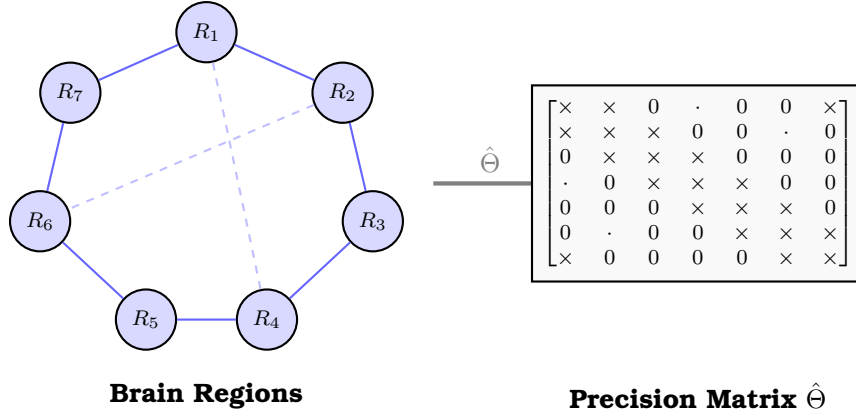


Figure 2.6: Functional brain connectivity as a graphical model. Brain regions $(R_1, \dots, R_7)$ are represented as nodes, with edges encoding conditional dependencies estimated from fMRI time series via the precision matrix $\hat{\Theta}$. Solid lines indicate strong conditional dependencies ($\hat{\Theta}_{ij} \neq 0$); dashed lines indicate weaker connections. Zeros in $\hat{\Theta}$ correspond to conditionally independent region pairs.

insight into the mechanisms of disease and suggest potential therapeutic targets (Zhang e Horvath, 2005). Large-scale datasets such as The Cancer Genome Atlas (TCGA) and the PAM50 breast cancer dataset provide rich resources for applying these methods to unravel the complex interplay of genes in cancer biology (Weinstein et al., 2013; Picornell et al., 2019).

In the field of remote sensing, hyperspectral imaging captures data across a wide spectrum of light, resulting in high-dimensional datasets where each pixel contains hundreds of spectral bands. Covariance estimation is fundamental to many analysis tasks in this domain, particularly for anomaly and target detection algorithms (Manolakis et al., 2013). These algorithms often rely on characterizing the statistical properties of the background clutter, which is modeled by the covariance matrix. An accurate estimate of this matrix allows for the effective differentiation of targets of interest from the background, even when they are subpixel-sized or camouflaged (Nasrabadi, 2013). Therefore, robust covariance estimation is critical for unlocking the full potential of hyperspectral data in applications ranging from defense and security to environmental monitoring.

2.4 The Challenge of High-Dimensional Covariance Estimation

2.4.1 The Sample Covariance Matrix and Its Limitations

Given a set of n data samples $(x_1, x_2, \dots, x_n)^\top = X$, where each sample $x_i \in \mathbb{R}^p$, the most direct way to estimate the covariance matrix is by using the Maximum Likelihood Estimation (MLE) framework. For data assumed to follow a multivariate Gaussian distribution, the MLE is the Sample Covariance Matrix (SCM), $\hat{\Sigma}$, formulated as:

$$\hat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^\top \quad (2.3)$$

where $\bar{x}$ denotes the sample mean. The SCM is an unbiased estimator, meaning that on average, it accurately reflects the true covariance structure when the number of samples n is significantly larger than the number of variables p .

However, in the high-dimensional settings common to modern data analysis ($n \leq p$), the SCM becomes a fundamentally flawed estimator. To see why, consider the centered data matrix $\tilde{X} = X - \mathbf{1}\bar{x}^\top \in \mathbb{R}^{n \times p}$, where each row has had the sample mean subtracted. The SCM can be written as:

$$\hat{\Sigma} = \frac{1}{n-1} \tilde{X}^\top \tilde{X}. \quad (2.4)$$

Since the rows of $\tilde{X}$ are constrained to sum to zero, the matrix $\tilde{X}$ has at most $\min(n-1, p)$ linearly independent rows, and therefore $\text{rank}(\tilde{X}) \leq \min(n-1, p)$. Because the rank of a product cannot exceed the rank of either factor, it follows that:

$$\text{rank}(\hat{\Sigma}) = \text{rank}(\tilde{X}^\top \tilde{X}) = \text{rank}(\tilde{X}) \leq \min(n-1, p). \quad (2.5)$$

When $n \leq p$, this bound yields $\text{rank}(\hat{\Sigma}) \leq n-1 < p$, so the SCM is singular. This rank deficiency has a direct practical consequence: the precision matrix $\hat{\Theta} = \hat{\Sigma}^{-1}$ cannot be computed by direct inversion, making it impossible to discover the conditional independence structure of the underlying GGM without additional regularization or structural assumptions.

Furthermore, even when the SCM is not strictly singular, it suffers from severe statistical instability in the high-dimensional regime. The matrix is corrupted by the effects of data scarcity: random fluctuations that obscure the true underlying structure. This phenomenon can be understood through the lens of random matrix theory: when n and p are comparable, the eigenvalues of the SCM deviate systematically from those of the true covariance matrix, with small eigenvalues being underestimated and large eigenvalues being overestimated (Ledoit e Wolf, 2004). The formal distributions governing this process, including the Wishart distribution and the Marchenko-Pastur law, are derived in Appendix A.2.

The formal framework for viewing this estimation challenge as a statistical degradation process, and the approach to reversing it, is developed in Chapter 4.

2.5 Deep Learning Architectures for Denoising

Recent advances in machine learning have inspired the exploration of deep learning techniques for estimating covariance matrices. These methods learn complex, non-linear map-

pings directly from data, offering a powerful alternative to the statistically-principled but more rigid baseline methods.

2.5.1 Supervised Learning with Convolutional Neural Networks

Supervised deep learning approaches treat covariance estimation as a regression or classification problem, requiring a labeled dataset where the true target (e.g., the covariance matrix or graph structure) is known for each input sample (Diskin e Wiesel, 2024).

A prominent example is DeepGraph, which uses a deep Convolutional Neural Network (CNN) to predict the presence or absence of edges in a GGM (Belilovsky et al., 2017). As shown in Figure 2.7, the model takes an empirical covariance matrix $\hat{\Sigma}$ as a 2D image-like input and learns a function $f_w(\hat{\Sigma})$ that outputs the corresponding graph adjacency matrix. The CNN architecture is specifically chosen for its ability to capture local spatial patterns and hierarchical features within the input matrix. The network is trained on synthetic data by minimizing an empirical risk function (e.g., cross-entropy loss) between the predicted and true graph structures.

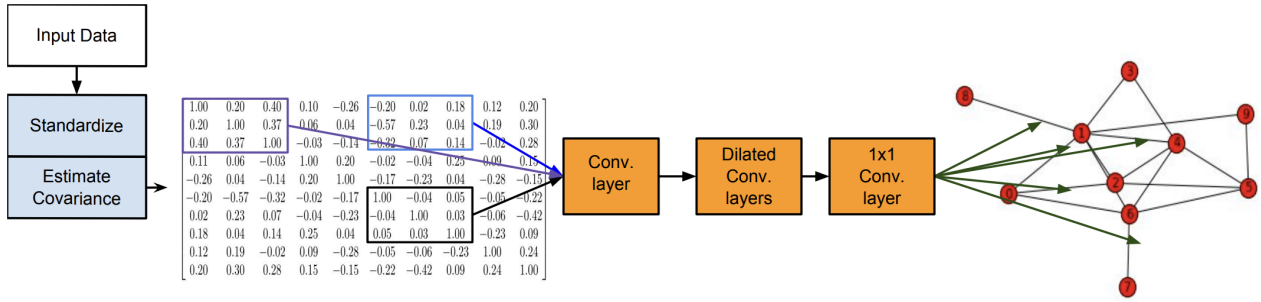


Figure 2.7: The DeepGraph architecture, a supervised CNN that maps a sample covariance matrix to an estimated graph structure (Belilovsky et al., 2017).

While powerful, the primary limitation of supervised methods like DeepGraph is their reliance on large, accurately labeled datasets, which are often unavailable. Nonetheless, they established the viability of using CNNs to interpret and process covariance matrices as structured inputs.

2.5.2 The U-Net Architecture for Structured Denoising

A particularly successful evolution of the CNN for image-to-image translation tasks is the U-Net architecture (Ronneberger et al., 2015). The U-Net is designed to process an input image and produce a new image of the same size, making it exceptionally well-suited for tasks like segmentation and, crucially for this thesis, denoising.

The U-Net consists of two main parts. First, a Contracting Path (Encoder), which is a series of convolutional and max-pooling layers that progressively downsample the input, capturing high-level, contextual features while reducing spatial dimensions. Second, an

Expansive Path (Decoder), specifically designed to symmetrically mirror the encoder, using transposed convolutions to progressively upsample the feature maps back to the original input resolution.

The defining innovation of the U-Net is the use of skip connections, which concatenate the feature maps from the encoder path directly to the corresponding layers in the decoder path. These connections allow the network to combine high-level contextual information from the decoder with the fine-grained, high-resolution spatial details from the encoder. This mechanism is critical for precise reconstruction and is a key reason for the U-Net’s success in tasks that require preserving fine-grained structural details. The overall architecture is illustrated in Figure 2.8.

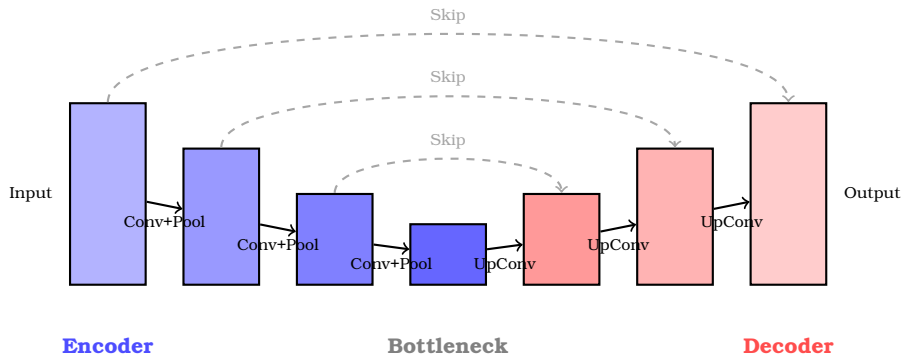


Figure 2.8: Schematic of the U-Net architecture (Ronneberger et al., 2015). The encoder (left, blue) progressively downsamples the input through convolutional and pooling layers, capturing increasingly abstract features. The decoder (right, red) upsamples back to the original resolution via transposed convolutions. Skip connections (dashed arrows) concatenate encoder feature maps to corresponding decoder layers, preserving fine-grained spatial details lost during downsampling.

The U-Net’s success in image denoising tasks suggests a natural connection to other domains where structured inputs are corrupted by noise. The encoder-decoder architecture with skip connections is well-suited for learning mappings that must preserve both coarse-scale and element-level dependencies in the input. The application of the U-Net architecture to covariance estimation is developed in Chapter 4.

2.6 Generative Denoising with Diffusion

While the U-Net provides a powerful architecture for direct denoising, generative models offer a more principled approach by learning the underlying data distribution itself. Denoising diffusion models have recently emerged as a state-of-the-art class of generative models capable of producing remarkably high-fidelity results (Ho et al., 2020).

2.6.1 Denoising Diffusion Probabilistic Models (DDPMs)

Denoising Diffusion Probabilistic Models are a class of generative models that learn to produce data by reversing a gradual noising process. The framework is defined by two opposing processes. The first is a fixed forward process, a predefined Markov chain that incrementally adds Gaussian noise to an input data sample over a series of T timesteps. This process is designed to be gradual enough that it slowly destroys the structure in the data, such that at the final timestep T , the sample is transformed into an instance of pure, unstructured isotropic Gaussian noise.

The second, and central, part of the framework is the learned reverse process. This process is a parameterized Markov chain, typically implemented with a neural network, that learns to reverse the diffusion. The network is trained to predict the noise that was added at any given timestep t , effectively acting as a denoiser. For generation, the model begins with a sample of pure Gaussian noise and iteratively applies the learned denoiser for T steps, progressively removing noise to transform the unstructured input into a clean, coherent sample from the learned data distribution. The primary strength of this framework is its ability to learn a highly flexible and expressive model. However, its core theoretical foundation relies on the properties of Gaussian noise.

Formally, the forward process defines a Markov chain over a sequence of latent variables. To prevent notational overload with the data vectors $x_i \in \mathbb{R}^p$ defined in Section 2.4.1, we denote the multidimensional diffusion variables (e.g., matrices or images) by $\mathbf{x}$, which distinguishes them from the regular vector representations².

Let $\mathbf{x}_0 \in \mathbb{R}^{d \times d}$ denote the original clean data sample and $\mathbf{x}_t$ the corrupted version obtained at timestep t . The forward transition distribution q , which maps a sample at timestep $t - 1$ to a noisier version at timestep t , progressively adds Gaussian noise according to a variance schedule $\{\beta_t\}_{t=1}^T$, with $\mathbf{I}$ denoting the d -dimensional identity matrix:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right). \quad (2.6)$$

A key property of this formulation is that the corrupted sample at any arbitrary timestep t can be expressed in closed form. Defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, one obtains:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}\right), \quad (2.7)$$

which allows direct sampling of $\mathbf{x}_t$ from $\mathbf{x}_0$ without iterating through intermediate steps, since each transition in the forward chain is Gaussian and Gaussian distributions are closed under composition (Ho et al., 2020). The reverse process p_θ , a learned Markov chain parameterized by neural network weights θ , is defined as:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I}\right), \quad (2.8)$$

²The original DDPM formulation (Ho et al., 2020) uses $x_0, \dots, x_T$ for the diffusion chain. We adopt bold $\mathbf{x}_0, \dots, \mathbf{x}_T$ to distinguish these quantities from the observation vectors x_i used throughout this thesis.

where μ_θ is a neural network that predicts the denoised mean and σ_t^2 is a predetermined variance (set to β_t in the original formulation). Ho et al. (2020) showed that reparameterizing the network to predict the noise ϵ rather than the mean leads to a simplified training objective:

$$L_{\text{simple}} = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2 \right], \quad (2.9)$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the noise sampled during the forward process and ϵ_θ is the neural network's prediction of that noise. The complete forward and reverse processes are illustrated in Figure 2.9.

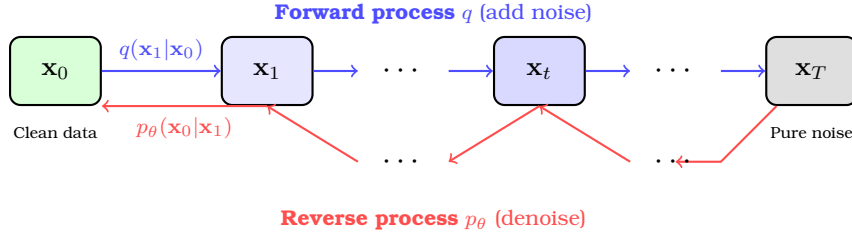


Figure 2.9: The DDPM framework (Ho et al., 2020). The forward process (blue, top) gradually corrupts a clean data sample $\mathbf{x}_0$ by adding Gaussian noise over T timesteps until only isotropic noise remains at $\mathbf{x}_T$. The learned reverse process (red, bottom) uses a neural network to iteratively denoise, recovering the original data from noise.

The DDPM framework is theoretically grounded in the Gaussian assumption: both the forward and reverse transitions are modelled as Gaussian distributions. This assumption is well-suited for image generation, where additive Gaussian noise is a reasonable corruption model. However, not all degradation processes conform to this assumption. As demonstrated empirically in Chapter 5, the residuals produced by the statistical degradation of covariance matrices are fundamentally non-Gaussian. This observation motivates the need for a more general diffusion framework that can accommodate arbitrary degradation operators.

2.6.2 Cold Diffusion: A Generalization for Arbitrary Degradations

To address the Gaussianity limitation identified above, Cold Diffusion was introduced as a generalization of the diffusion framework that removes the requirement for Gaussian noise (Bansal et al., 2023). The core innovation of Cold Diffusion is that it replaces the fixed Gaussian noising process with an arbitrary, and potentially non-Gaussian, degradation operator. This allows the framework to be tailored to specific, real-world corruption phenomena.

The Cold Diffusion framework is built around two key components:

- **The Degradation Operator (D):** This is a deterministic function, $\mathbf{x}_t = D(\mathbf{x}_0, t)$, that transforms a clean data sample $\mathbf{x}_0$ into a degraded version $\mathbf{x}_t$ at timestep t . The

severity of the degradation increases with t . Unlike DDPMs, the degradation need not be Gaussian; it can represent any domain-specific corruption process.

- **The Restoration Operator (R):** This is a neural network, $\hat{\mathbf{x}}_0 = R(\mathbf{x}_t, t)$, trained to be the inverse of the degradation operator. Its goal is to predict the original clean data $\mathbf{x}_0$ given a degraded version $\mathbf{x}_t$. The network is trained by minimizing a reconstruction loss, such as the L1 norm, over all timesteps:

$$\min_{\theta} \mathbb{E}_{\mathbf{x}, t} \|R_{\theta}(D(\mathbf{x}, t), t) - \mathbf{x}\|_1, \quad (2.10)$$

where R_{θ} is the network parameterized by weights θ .

During inference, the model starts with a highly degraded sample $\mathbf{x}_T$ and iteratively refines it. At each step s , the restoration network R predicts the clean data, $\hat{\mathbf{x}}_0 = R(\mathbf{x}_s, s)$. This prediction is then used to compute a less degraded sample $\mathbf{x}_{s-1}$ using a specialized update rule, effectively taking a small step backward through the degradation process. This iterative refinement is crucial for generating high-quality, clean data from a severely corrupted input. The general framework is illustrated in Figure 2.10. The flexibility to define a custom degradation process makes Cold Diffusion applicable to a wide range of domain-specific restoration problems. The application of this framework to covariance estimation is detailed in Chapter 4.

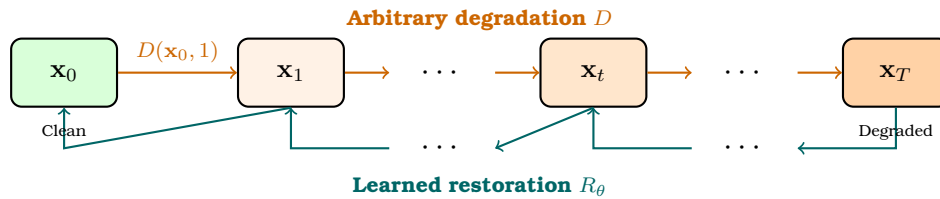


Figure 2.10: The Cold Diffusion framework (Bansal et al., 2023). The forward degradation (orange, top) applies an arbitrary, potentially non-Gaussian operator D that progressively corrupts a clean sample $\mathbf{x}_0$. The learned restoration (teal, bottom) iteratively recovers the original data using a neural network R_{θ} , without requiring Gaussian noise assumptions.

2.7 Assessing Data Normality

The theoretical foundation of Denoising Diffusion Probabilistic Models (DDPMs) is contingent upon the assumption that the underlying noise process is Gaussian. A departure from this assumption can compromise model performance and theoretical guarantees. Therefore, it is essential to employ rigorous methods to assess the normality of a given data distribution before applying such models. This assessment is typically performed using a combination of qualitative graphical methods and quantitative statistical hypothesis tests.

2.7.1 Graphical Analysis

Visual inspection provides an intuitive first pass at evaluating normality. A histogram of the data can reveal the general shape of the distribution; for normally distributed data, it should approximate a symmetric, bell-shaped curve. A more powerful graphical tool is the Quantile-Quantile (Q-Q) plot, which plots the quantiles of the empirical data against the theoretical quantiles of a standard normal distribution. If the data is normally distributed, the points on the Q-Q plot will form an approximately straight line. Significant deviations from this line, particularly at the tails, suggest that the data is not normal.

2.7.2 Statistical Hypothesis Testing

For a formal, quantitative assessment, statistical hypothesis tests are used. These tests evaluate a null hypothesis (H_0) that the data sample is drawn from a normally distributed population, against an alternative hypothesis (H_a) that it is not. The outcome is a p-value, which represents the probability of observing the given data if the null hypothesis were true. A small p-value (typically $p < 0.05$) provides evidence to reject the null hypothesis and conclude that the data is non-Gaussian. Two widely used tests for this purpose are the Shapiro-Wilk and D'Agostino's K-squared tests.

The Shapiro-Wilk test ([Shapiro e Wilk, 1965](#)) is considered one of the most powerful normality tests. It calculates a test statistic, W , that essentially measures the correlation between the sample data and the expected values for a normal distribution. The statistic is defined as:

$$W = \frac{(\sum_{i=1}^n a_i x_{(i)})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.11)$$

where $x_{(i)}$ are the ordered sample values, $\bar{x}$ is the sample mean, and the coefficients a_i are constants derived from the order statistics of a sample from a normal distribution. Values of W close to 1 indicate a good fit to normality.

The D'Agostino's K-squared test ([D'Agostino, 1971](#); [D'Agostino e Pearson, 1973](#)) is another robust method that assesses normality by jointly considering the sample's skewness and kurtosis. Skewness measures the asymmetry of the distribution, while kurtosis measures the heaviness of its tails. The test statistic, K^2 , is computed as:

$$K^2 = Z_1(\sqrt{b_1})^2 + Z_2(b_2)^2 \quad (2.12)$$

where $\sqrt{b_1}$ is the sample skewness and b_2 is the sample kurtosis. Z_1 and Z_2 are transformations that convert the skewness and kurtosis values to be approximately standard normal. Under the null hypothesis, the K^2 statistic follows a chi-squared distribution with two degrees of freedom. A high value of K^2 indicates significant deviation from normality.

As will be demonstrated empirically in Chapter 5, applying these methods to the statistical degradation process reveals that it is fundamentally non-Gaussian. This finding is critical

as it invalidates the core assumption of standard DDPMs and provides a strong justification for adopting a more generalized framework.

2.8 Metrics for Evaluating Covariance Estimation

Evaluating covariance estimation methods requires metrics that capture complementary aspects of estimator quality. We organize our evaluation strategy into two categories. The first addresses the intrinsic quality of the estimated covariance ($\hat{\Sigma}$) and precision ($\hat{\Theta}$) matrices, comprising reconstruction quality metrics (Frobenius norm), probabilistic fidelity metrics (KL divergence, log-likelihood), and measures of how faithfully the matrices capture the true dependence structure. The second category focuses on graph structure recovery, assessing the ability to correctly identify the conditional independence pattern (sparsity pattern of $\hat{\Theta}$) through classification metrics applied at the edge level.

2.8.1 Quality of Covariance and Precision Matrix Estimates

The primary measure of success is the fidelity of the estimated matrices. This reflects how faithfully an estimator captures the true underlying dependence structure of the data, which is encapsulated by the population covariance matrix (Σ) and its inverse, the precision matrix (Θ). High-quality estimates are crucial as they enable accurate inference of conditional independence relationships and facilitate reliable downstream analyses. We evaluate this fidelity using the following metrics.

Frobenius Norm (Reconstruction Error)

The Frobenius norm provides a direct measure of the element-wise closeness of the estimated covariance matrix $\hat{\Sigma}$ to the ground truth Σ . It essentially quantifies the Euclidean distance between the matrices and is defined as:

$$\|\hat{\Sigma} - \Sigma\|_F = \sqrt{\sum_{i,j} (\hat{\Sigma}_{ij} - \Sigma_{ij})^2} \quad (2.13)$$

A low Frobenius norm indicates that the estimator has accurately recovered the magnitude and sign of the covariance entries. Unlike Mean Squared Error (MSE), which averages the error, the Frobenius norm represents the total error magnitude, identifying global structural deviations.

Normalized Kullback-Leibler (KL) Divergence

Following the definition in [Zamar et al. \(2021\)](#), we measure the discrepancy between the estimated precision matrix $\hat{\Theta}$ and the true precision matrix Θ using the Kullback-Leibler

divergence. This metric is preferred over element-wise errors for precision matrices as it accounts for the geometric structure of the positive definite cone. The standard KL divergence is given by:

$$D_{KL} = \frac{1}{2} \left(\text{tr}(\hat{\Theta}\Theta^{-1}) - \log \det(\hat{\Theta}\Theta^{-1}) - p \right) \quad (2.14)$$

where $\text{tr}(\cdot)$ denotes the matrix trace (sum of diagonal elements), $\det(\cdot)$ denotes the matrix determinant, and p is the dimensionality (number of variables). To facilitate comparison across different dimensions and bound the metric for easier interpretation, we employ the normalized form:

$$\text{NKL} = \frac{D_{KL}}{1 + D_{KL}} \quad (2.15)$$

The NKL provides a bounded measure of how much information is lost when approximating the true graphical structure with the estimated one. An NKL close to 0 indicates near-perfect recovery of the probabilistic structure.

Log-Likelihood (Generalization)

To assess the generalization capability of the model, particularly on real-world data where the ground truth is unknown, we evaluate the Log-Likelihood of unseen holdout data. For a set of n test observations with test sample covariance $\hat{\Sigma}_{test}$, the log-likelihood under the estimated model $\hat{\Sigma}$ is:

$$\mathcal{L}(\hat{\Sigma}) = -\frac{n}{2} \ln |\hat{\Sigma}| - \frac{1}{2} \text{tr}(\hat{\Sigma}^{-1} \hat{\Sigma}_{test}) \quad (2.16)$$

This metric is the gold standard for evaluating generative models. A higher log-likelihood implies that the model assigns a higher probability to the valid, unseen data, indicating it has learned a robust representation of the underlying distribution rather than overfitting to the training samples. Full derivations of the KL Divergence and Log-Likelihood from first principles are provided in Appendix A.3.

2.8.2 Metrics for Graph Structure Recovery

Beyond numerical accuracy, a critical objective is to recover the topology of the underlying Gaussian Graphical Model. This involves correctly identifying the non-zero entries in the precision matrix $\Theta = \Sigma^{-1}$, which correspond to edges in the graph. We treat this as a binary classification problem (edge vs. no-edge) and evaluate it using the following metrics.

ROC AUC (Graph Topology)

The Area Under the Receiver Operating Characteristic Curve (ROC AUC) is our primary metric for evaluating the global quality of the estimated graph topology. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across all possible threshold set-

tings:

$$\text{ROC AUC} = \int_0^1 \text{TPR}(\text{FPR}) d\text{FPR} \quad (2.17)$$

This metric is crucial for tasks like connectome discovery, where identifying the correct network connections (sparsity pattern) is often more important than the exact numerical values of the edge weights (Belilovsky et al., 2017). It provides a robust measure of a model's ability to recover the true graph topology from limited data, independent of the specific threshold chosen to binarize the matrix.

Precision, Recall, Accuracy, and F1-Score

To provide a comprehensive view of classification performance, we report standard metrics. Precision measures the reliability of predicted edges:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2.18)$$

Recall (Sensitivity) measures the proportion of true edges effectively recovered:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2.19)$$

Accuracy measures the overall correctness of the graph structure:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2.20)$$

The F1-Score is the harmonic mean of Precision and Recall, offering a balanced metric, particularly useful when the graph sparsity leads to class imbalance:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.21)$$

AUC of PR and ROC Curves

We utilize both the Area Under the ROC Curve (ROC AUC) and the Area Under the Precision-Recall Curve (PR AUC). ROC AUC evaluates the trade-off between True Positive Rate and False Positive Rate. PR AUC is often preferred in sparse settings (like brain networks) as it focuses on the performance of the positive class (edges), ignoring the large number of true negatives (non-edges).

Note on metric reuse in downstream classification

The metrics defined above (Accuracy, ROC-AUC, and Average Precision) are presented here in the context of graph structure recovery, where they evaluate *edge-level* predictions (edge

vs. no-edge). In Chapter 5, we also employ these same metrics for a *subject-level* downstream classification task on real-world neuroimaging data, where they evaluate the ability to distinguish diagnostic groups (e.g., Control vs. Autism) using features extracted from the estimated precision matrix. The mathematical definitions remain identical; only the level of analysis changes from edges to subjects.

The Graph Signal Processing and Applications material presented in Sections 2.2 and 2.3 provides essential foundational knowledge, specifically the formal connection between graph topology and covariance structure, that is prerequisite for understanding the degradation operator and restoration framework developed in Chapter 4.

2.9 Chapter Summary

This chapter has established the theoretical foundations required for the remainder of this thesis. We introduced Gaussian Graphical Models and showed how the precision matrix encodes conditional independence structure, making accurate covariance estimation essential for graph discovery. The complementary perspective of Graph Signal Processing reinforced this connection by framing topology inference as a covariance estimation problem. We then characterized the core challenge: the Sample Covariance Matrix becomes rank-deficient and singular when the number of samples does not exceed the number of variables, rendering direct precision matrix estimation impossible.

On the methodological side, we presented the U-Net architecture for structured denoising and the Denoising Diffusion Probabilistic Models framework, including its formal mathematical formulation. We identified the Gaussianity assumption as a key limitation of standard DDPMs and introduced Cold Diffusion as a generalization that accommodates arbitrary degradation operators. We also presented the statistical tools for assessing data normality, which will be used in Chapter 5 to empirically verify the non-Gaussian nature of the degradation process. Finally, we defined the evaluation metrics used throughout this work. With these foundations in place, Chapter 3 surveys the existing literature on covariance estimation methods and positions our contribution within the broader research landscape.

Literature Review

This chapter positions our generative denoising framework within the broader landscape of covariance estimation research. Building on the theoretical foundations established in Chapter 2, we examine how existing methods have attempted to address the challenge of high-dimensional covariance estimation. We begin by exploring functional brain connectivity as a key application domain, reviewing concrete studies that have applied covariance and precision estimation to neuroimaging data, and briefly surveying alternative domains to justify this focus. We then provide a comprehensive review of estimation methods organized into two paradigms: statistical regularization and machine learning. For each, we report key results, evaluation metrics, computational costs, and baselines. Comparison tables and conceptual figures synthesize these findings. Through this critical analysis, we identify a crucial conceptual gap: the absence of methods that explicitly model and reverse the statistical degradation process inherent in finite-sample covariance estimation. This gap motivates the novel framework introduced in Chapter 4.

3.1 Application Domain: Functional Brain Connectivity

The human brain operates as a complex, interconnected system whose intricate communication patterns give rise to cognition and behavior. The concept of functional connectivity provides a powerful framework for studying these patterns by measuring the statistical dependencies between the activities of different brain regions over time (Bear et al., 2017). This perspective treats the brain as a network, where regions are nodes and functional relationships are edges. The complete map of these connections is often referred to as the “connectome,” which serves as a unique fingerprint of an individual’s neural architecture (Seung, 2018). Understanding this connectome is a central goal of modern neuroscience, offering insights into the fundamental principles of brain organization.

Gaussian Graphical Models (GGMs) have become an indispensable tool for estimating these functional networks from neuroimaging data, such as Functional Magnetic Resonance

Imaging (fMRI). While a simple correlation matrix can reveal pairwise associations between brain regions, it cannot distinguish direct connections from spurious ones arising from a common cause. GGMs overcome this limitation by estimating the precision matrix, which encodes the conditional independence structure between variables. The non-zero entries in the precision matrix correspond to direct functional connections, forming the edges of the brain graph and providing a more accurate representation of the underlying network topology (Bullmore e Sporns, 2009).

Several landmark studies have demonstrated the effectiveness of GGM-based approaches for mapping brain connectivity. Varoquaux et al. (2010) proposed a group-sparse Graphical Lasso with a population prior for multi-subject fMRI analysis, showing that leveraging shared structure across subjects significantly improves individual connectivity estimates as measured by held-out log-likelihood. In a comprehensive comparison of network modelling methods, Smith et al. (2011) evaluated multiple approaches, including partial correlations, Tikhonov-regularized precision matrices, Bayesian networks, and the Graphical Lasso, on both simulated and resting-state fMRI data, finding that partial correlation methods consistently outperformed full correlation for edge detection as measured by ROC AUC. Huang et al. (2010) applied sparse inverse covariance estimation to the Alzheimer’s Disease Neuroimaging Initiative (ADNI) dataset, identifying disrupted connectivity patterns in patients with Alzheimer’s disease relative to healthy controls and achieving 88% sensitivity and 88% specificity in discriminating the two groups. These studies collectively demonstrate that precision matrix estimation is not merely an academic exercise but a practical tool for extracting clinically meaningful information from neuroimaging data.

The ability to accurately map the connectome has profound clinical implications, particularly in the study of neurodevelopmental disorders. By applying GGM-based approaches to datasets from different populations, researchers can identify systematic alterations in brain connectivity associated with conditions like autism spectrum disorder. Studies using large-scale datasets such as the Autism Brain Imaging Data Exchange (ABIDE) have revealed atypical patterns of both local and long-range connectivity in individuals with autism, offering potential biomarkers and shedding light on the neurobiological basis of the condition (Di Martino et al., 2014; Sporns e Betzel, 2016). Critically, these neuroimaging datasets exemplify the high-dimensional estimation challenge: a typical fMRI study parcellates the brain into $p = 50\text{--}400$ regions of interest (ROIs) but obtains only $n = 100\text{--}300$ time points per subject, placing the problem squarely in the $p \geq n$ regime where the sample covariance matrix becomes poorly conditioned or singular.

3.1.1 Alternative Application Domains

While this thesis focuses on functional brain connectivity, covariance estimation is a fundamental challenge across numerous scientific disciplines. Table 3.1 summarizes four representative domains where the quality of the covariance or precision estimate directly impacts downstream analysis.

Table 3.1: Application domains where covariance estimation plays a critical role. Brain connectivity is selected as the primary domain for this thesis due to the inherent $p \approx n$ regime and clinical significance.

Domain	Typical p	Typical n	Key Challenge	Representative Study
Brain Connectivity	50–400	100–300	$p \approx n$; non-stationary	Varoquaux et al. (2010)
Gene Networks	1k–20k	50–500	$p \gg n$; ultra-high dim.	Schäfer e Strimmer (2005)
Portfolio Optim.	100–5k	250–2.5k	Heavy tails; non-stationary	Markowitz (1952)
Hyperspectral	100–300	Varies	Spatial correlation	Manolakis et al. (2013)

In gene regulatory network inference, researchers must estimate dependencies among thousands or even tens of thousands of genes from a few hundred biological samples, making this an extreme instance of the $p \gg n$ problem ([Schäfer e Strimmer, 2005](#); [Emmert-Streib et al., 2014](#)). In financial portfolio optimization, the covariance matrix of asset returns governs risk allocation; estimation errors propagate directly into suboptimal portfolios, particularly under the heavy-tailed, non-stationary distributions typical of financial markets ([Markowitz, 1952](#); [Ledoit e Wolf, 2004](#)). In hyperspectral imaging, accurate estimation of spectral covariance is essential for target detection algorithms that must distinguish faint signals from complex background clutter ([Manolakis et al., 2013](#)).

We select functional brain connectivity as the primary application domain for three reasons: (1) it naturally falls in the challenging $p \approx n$ regime where the sample covariance matrix is near-singular, making it an ideal testbed for improved estimators; (2) it carries direct clinical significance, with potential to improve diagnostic tools for neurological and psychiatric conditions; and (3) large-scale public datasets such as ABIDE ([Di Martino et al., 2014](#)) and ADNI ([Huang et al., 2010](#)) provide standardized benchmarks for reproducible evaluation.

3.2 A Review of Estimation Methods

The challenge of accurately estimating covariance matrices from noisy, high-dimensional data has given rise to a variety of methodological approaches. This section reviews the dominant paradigms in the field, which serve as the context and primary benchmarks for the novel framework proposed in this thesis. The selection of methods reviewed here was guided by three criteria: (1) relevance to covariance estimation in high-dimensional settings, (2) methodological diversity covering both classical statistical approaches and mod-

ern machine learning techniques, and (3) coverage of both regularization-based estimators and deep learning architectures to provide a comprehensive landscape of the field.

3.2.1 Paradigm 1: Statistical Regularization Methods

The first major paradigm for addressing the limitations of the SCM relies on statistical regularization and optimization principles. These methods impose mathematical constraints or penalties to stabilize the estimation process.

Shrinkage Estimation: The Ledoit-Wolf Estimator

One of the most widely adopted approaches is shrinkage estimation, which improves the conditioning of the SCM by regularizing it toward a structured target. The Ledoit-Wolf estimator (Ledoit e Wolf, 2004) represents the canonical implementation of this principle. It computes an optimal convex combination of the SCM and a well-conditioned target matrix (typically the identity or diagonal matrix):

$$\hat{\Sigma}_{LW} = \rho \cdot T + (1 - \rho) \cdot \hat{\Sigma} \quad (3.1)$$

where T is the target matrix and $\rho \in [0, 1]$ is the shrinkage intensity, determined analytically to minimize the expected squared error. This “shrinking” of the empirical eigenvalues toward a central value results in a well-conditioned, invertible matrix that is particularly robust in the $p > n$ regime.

In their original Monte Carlo experiments with $p = 30$ features and $n = 35$ samples, Ledoit e Wolf (2004) demonstrated approximately 40% reduction in mean squared error (MSE) relative to the unregularized SCM. The method’s computational cost is dominated by forming the SCM itself, $O(p^2n)$, since the optimal shrinkage intensity admits a closed-form solution requiring no iterative optimization. This analytical tractability has made Ledoit-Wolf the default baseline in virtually all subsequent covariance estimation studies and a standard component of statistical software packages such as scikit-learn (Pedregosa et al., 2011). The closely related Oracle Approximating Shrinkage (OAS) estimator further refines the shrinkage intensity through an improved approximation of the oracle formula. However, while effective at improving numerical stability, shrinkage methods produce dense matrices and do not exploit the sparsity often present in real-world networks.

Sparsity-Based Regularization: Graphical Lasso

The Graphical Lasso (Friedman et al., 2008) has emerged as the standard method for sparse covariance estimation. It leverages the assumption that many real-world systems exhibit

sparse conditional independence structures. The method estimates a sparse precision matrix Θ by solving the following L1-penalized maximum likelihood problem:

$$\hat{\Theta} = \arg \min_{\Theta \succ 0} -\log \det \Theta + \text{tr}(\hat{\Sigma}\Theta) + \lambda \|\Theta\|_1 \quad (\text{Graphical Lasso})$$

where $\lambda \geq 0$ is a regularization parameter controlling sparsity. The L1 penalty shrinks many off-diagonal elements to exactly zero, simultaneously performing estimation and variable selection. Figure 3.1 illustrates how increasing λ produces progressively sparser estimates.

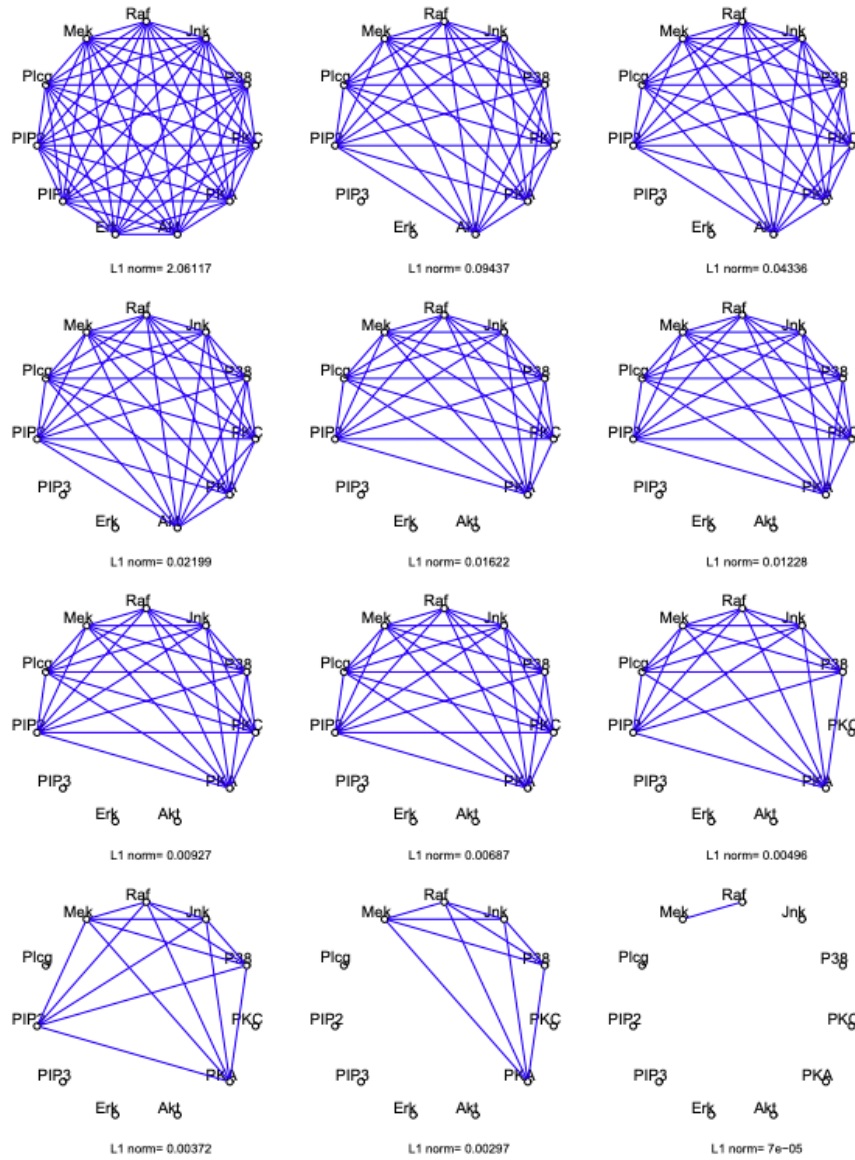


Figure 3.1: Regularization path for Graphical Lasso: as λ increases, the estimated precision matrix becomes progressively sparser (Laska e Narayan, 2017).

While powerful and widely adopted, Graphical Lasso has notable limitations. Its performance is highly sensitive to the choice of λ , typically requiring computationally expensive cross-validation. Moreover, the element-wise L1 penalty may not effectively capture struc-

tured sparsity patterns common in real networks, such as hub structures or community patterns.

The key computational contribution of [Friedman et al. \(2008\)](#) was a block coordinate descent algorithm that converges roughly $1000\times$ faster than the interior-point methods previously used for the same problem. Each iteration requires $O(p^3)$ operations due to the underlying Cholesky factorization, with convergence typically achieved in 5–50 iterations. In practice, however, selecting λ via k -fold cross-validation multiplies the total cost by $k \times |\Lambda|$, where $|\Lambda|$ is the size of the regularization grid. Graphical Lasso was benchmarked against earlier approaches including [Meinshausen e Bühlmann \(2006\)](#) neighborhood selection and [Yuan e Lin \(2007\)](#) penalized likelihood, establishing itself as the standard method that subsequent work must outperform. The `skggm` package ([Laska e Narayan, 2017](#)) extends the basic Graphical Lasso with adaptive and group-penalty variants.

Iterative Graph Construction: Step-Graph

The Step-Graph algorithm ([Zamar et al., 2021](#)) represents a fundamentally different approach through iterative, local graph construction. Rather than global optimization, it employs a forward-backward selection procedure. The algorithm starts with an empty graph containing no edges. It then iteratively adds the edges that most improve the model fit (forward step) and subsequently removes any edges that contribute least to the model (backward step). This process continues until convergence or a predefined stopping criterion is met.

Figure 3.2 visualizes this iterative refinement process. By making local decisions based on partial correlations, Step-Graph achieves computational efficiency that scales well to high-dimensional problems, making it particularly suitable when Graphical Lasso with full cross-validation becomes prohibitive.

[Zamar et al. \(2021\)](#) reported comparable or superior edge detection accuracy relative to Graphical Lasso while achieving significantly lower computational cost in high-dimensional settings. Each forward or backward step requires $O(p^2)$ operations, avoiding the $O(p^3)$ bottleneck of Graphical Lasso iterations. The method was benchmarked against Graphical Lasso (with BIC and cross-validation model selection), CLIME, and TIGER. However, the greedy, local nature of the forward-backward decisions means Step-Graph offers no theoretical guarantees on global optimality and may miss complex dependency structures that require coordinated edge additions.

3.2.2 Alternative Machine Learning Approaches

The limitations of hand-crafted statistical priors have motivated a recent shift toward data-driven approaches. Deep learning, with its capacity to approximate complex, high-

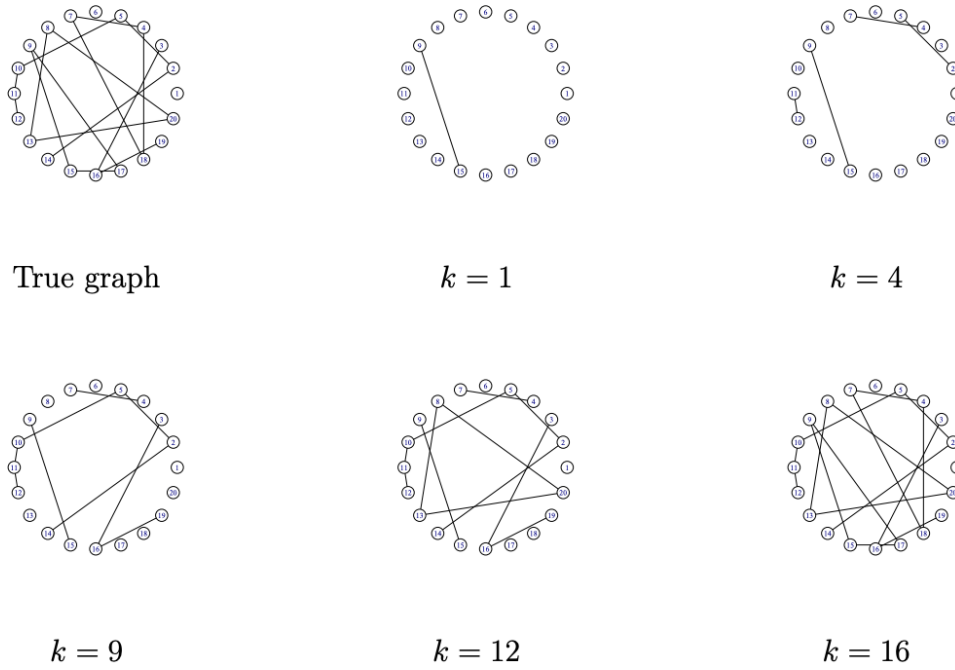


Figure 3.2: The Step-Graph algorithm progressively refines the graph structure through local updates (Zamar et al., 2021).

dimensional functions, offers a promising path to learn the estimation prior directly from data.

Supervised Graph Discovery: DeepGraph

DeepGraph (Belilovsky et al., 2017) represents a pioneering effort to apply supervised deep learning to the problem of graphical model selection. The core insight of DeepGraph is to treat the empirical covariance matrix $\hat{\Sigma}$ not merely as a set of statistics, but as a structured 2D input tensor, analogous to an image with a single channel.

The architecture employs a multi-layer Convolutional Neural Network (CNN) with filters of varying sizes (e.g., 3×3 , 5×5) to extract local features from the covariance matrix. These convolutional layers are designed to capture local dependency patterns among variables. The network is trained in a fully supervised manner: it takes empirical covariance matrices (generated from synthetic data with known ground truth) as input and is penalized on the reconstruction of the true graph structure.

Mathematically, DeepGraph learns a mapping function $f_\theta : \mathbb{R}^{p \times p} \rightarrow \{0, 1\}^{p \times p}$ that predicts the adjacency matrix A of the underlying graph:

$$\hat{A} = f_\theta(\hat{\Sigma}) \quad (3.2)$$

On synthetic data with hub-graph and scale-free topologies, [Belilovsky et al. \(2017\)](#) reported F1 scores of approximately 0.85 compared to Graphical Lasso’s 0.55 at a sample-to-dimension ratio of $n/p = 0.5$. This substantial improvement in the low-sample regime ($n < p$) validates the hypothesis that CNN-based architectures are effective for processing covariance matrices and can learn priors that outperform fixed mathematical assumptions. At inference time, the method requires only a single forward pass through the network. However, DeepGraph’s reliance on supervised training is a major bottleneck: it requires generating vast amounts of synthetic data whose graph topology and sampling distribution must statistically resemble the unknown real-world target. This non-trivial challenge, known as the “simulation-to-reality gap,” causes performance to degrade when the test distribution differs from the training distribution. Furthermore, DeepGraph inherently limits its output to estimating only the adjacency matrix, failing to provide the full covariance or precision matrices that are critical for many downstream applications.

A critical limitation of CNN-based approaches like DeepGraph is their sensitivity to variable ordering. Convolutional filters assume adjacent matrix entries correspond to semantically related variables. In domains without a natural spatial ordering (e.g., gene expression), this can lead to learning artifacts of the variable permutation. This limitation applies to the current ColdCov implementation, which uses a U-Net backbone. However, the ColdCov framework is architecture-agnostic: its degradation operator and SPSP constraint are independent of the neural backbone. Future work could adopt permutation-invariant architectures, such as Transformers, to overcome this ordering sensitivity while preserving the generative denoising benefits.

Self-Supervised Learning: SSCE

SSCE ([Diskin e Wiesel, 2024](#)) takes a fundamentally different approach by removing the need for ground-truth covariance matrices entirely. The method employs a “masked token prediction” strategy adapted for covariance estimation: a subset of the observed features (columns) in the input data matrix X is randomly masked, and a neural network, such as a Transformer or Graph Neural Network, is trained to reconstruct the masked features from the remaining context. By solving this pretext task, the model implicitly learns the internal covariance structure of the data. [Diskin e Wiesel \(2024\)](#) establish a theoretical connection between this self-supervised objective and maximum likelihood estimation under certain regularity conditions.

While conceptually elegant, SSCE treats covariance estimation as a feature completion task in the data domain, rather than explicitly modelling the sampling process in the covariance domain. This distinction is subtle but important: SSCE learns to “fill in blanks” in the observed data, whereas the framework proposed in this thesis learns to “reverse noise” in the estimated covariance matrix. The degradation caused by computing a covariance matrix from finite samples is a specific, deterministic transformation (as detailed in Chapter 2) that SSCE’s generic masking objective does not directly address.

A further practical limitation is that the authors of SSCE did not release a public implementation. Our own reimplementation revealed two significant issues: the method produces dense covariance estimates (offering no sparsity) and incurs prohibitive training times due to the Transformer-based architecture. These practical limitations, combined with the conceptual mismatch between masking-based pretext tasks and the structured nature of the statistical degradation, provide additional motivation for seeking a purpose-built generative framework.

3.2.3 Comparative Synthesis of Estimation Methods

The preceding subsections have reviewed the formulation, reported results, and limitations of individual covariance estimation methods. In this subsection, we synthesize these findings by comparing the methods across key conceptual and practical dimensions.

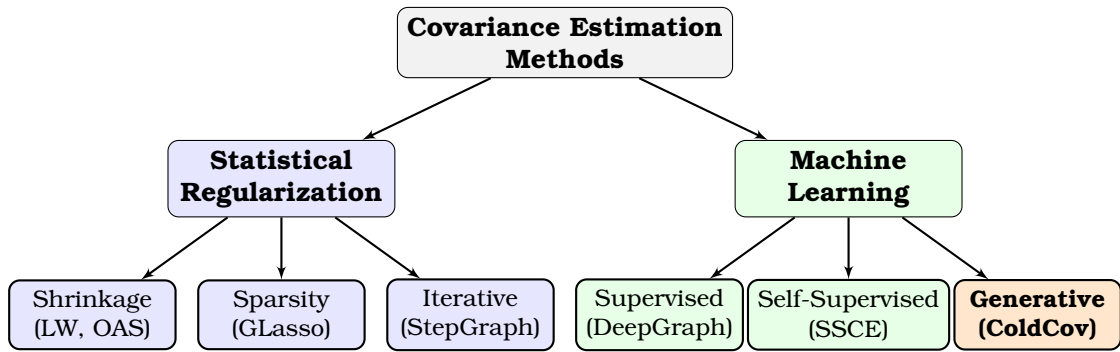


Figure 3.3: Taxonomy of covariance estimation methods reviewed in this chapter. Statistical methods impose fixed mathematical priors, while machine learning methods learn from data. The generative denoising approach (this thesis) fills the identified gap.

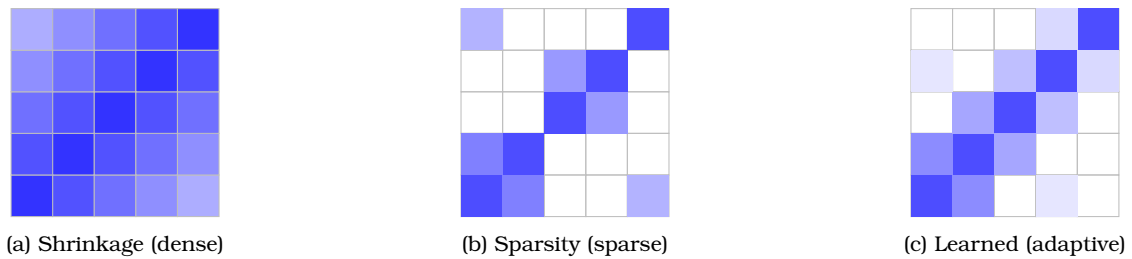


Figure 3.4: Schematic illustration of how different estimation paradigms affect precision matrix structure. (a) Shrinkage produces a dense, regularized matrix. (b) Sparsity-based methods produce matrices with explicit zero entries. (c) Learned approaches produce data-adaptive structures that can capture complex patterns without imposing a rigid prior.

The most fundamental divide between the reviewed methods is their paradigmatic approach to the estimation problem, as illustrated in the taxonomy in Figure 3.3. Statistical regularization methods (Ledoit-Wolf, Graphical Lasso, StepGraph) attempt to manage the symptoms of data scarcity by imposing fixed mathematical priors, such as uniform shrinkage or element-wise sparsity. In contrast, machine learning methods (DeepGraph, SSCE) treat covariance estimation as a data-driven learning problem, using neural architectures to dis-

cover adaptive priors from observed data. The generative denoising approach proposed in this thesis sits within the machine learning paradigm but differs by explicitly modeling the statistical degradation process.

These differing paradigms lead to distinct output properties, which are schematically contrasted in Figure 3.4. Shrinkage methods produce dense matrices that improve conditioning but obscure conditional independence structures. Sparsity-based regularizers successfully identify zero entries but struggle with complex, non-random network topologies. Data-driven approaches have the potential to produce adaptive structures that capture realistic patterns without imposing rigid priors, provided they are trained effectively.

Table 3.2: Comparison of covariance and precision matrix estimation methods reviewed in this chapter. The final row highlights the gap addressed by the proposed ColdCov framework.

Method	Paradigm	Key Assumption	Reported Performance	Complexity	Main Limitation
Ledoit-Wolf	Statistical	Shrinkage toward target	$\sim 40\%$ MSE reduction vs. SCM	$O(p^2n)$	Dense output; no sparsity
Graphical Lasso	Statistical	Element-wise sparsity (L1)	$1000\times$ faster than IP methods	$O(p^3)/\text{iter}$	Sensitive to λ ; requires CV
StepGraph	Statistical	Local graph construction	Comparable to GLasso; lower cost	$O(p^2)/\text{step}$	Greedy; no global optimality
DeepGraph	ML (Supervised)	CNN on SCM as image	$F1 \approx 0.85$ vs. GLasso 0.55 ($n < p$)	Forward pass	Estimates adjacency only
SSCE	ML (Self-Sup.)	Masked feature prediction	Theoretical MLE connection	High (Transformer)	Dense output; slow training
ColdCov	ML (Generative)	Reverse degradation	See Chapter 5	$O(p^2)$ (Sec. 4.2.5)	Synthetic data reliance

Beyond conceptual differences, Table 3.2 highlights significant practical trade-offs across the reviewed methods. The methods diverge substantially in their computational complexity, ranging from efficient analytical solutions to demanding iterative optimization or slow neural network training. Furthermore, they differ critically in their data requirements, relying either on unsupervised analytical forms, large-scale synthetic labels, or self-supervised pretext tasks. Finally, the table reveals that only a subset of these methods provides an architectural guarantee that the final output is a valid, symmetric positive semi-definite (SPSD) covariance matrix.

Having mapped this landscape, we can now synthesize these dimensions to identify a critical conceptual void. This comparative overview sets the stage for defining the specific research gap that motivates the development of the ColdCov framework.

3.3 Identifying the Research Gap: From Regularization to Generative Denoising

Our review reveals a landscape of increasingly sophisticated but fundamentally limited approaches to high-dimensional covariance estimation. Despite the diversity of methods surveyed, a critical conceptual gap persists, one that this thesis directly addresses.

Traditional statistical methods achieve reliable results within their design assumptions. Ledoit-Wolf shrinkage reduces MSE by approximately 40% relative to the SCM but produces dense matrices that obscure the sparse conditional independence structures present in real-world networks. Graphical Lasso recovers sparse graphs efficiently, yet its performance hinges on the choice of regularization parameter λ and the validity of the element-wise sparsity assumption. StepGraph offers computational savings but sacrifices global optimality guarantees. In essence, these methods manage the symptoms of data scarcity through fixed mathematical priors (shrinkage, sparsity, or iterative construction) without explicitly modelling the degradation process itself.

Recent machine learning approaches have demonstrated the power of learning data-driven priors. DeepGraph achieves F1 scores of 0.85 where Graphical Lasso reaches only 0.55 in the low-sample regime, validating that neural networks can learn estimation priors superior to hand-crafted ones. However, this performance comes at the cost of requiring large-scale labeled synthetic datasets, a requirement that is seldom met in practice and introduces the simulation-to-reality gap. SSCE removes the labelling requirement through self-supervised training but relies on a generic masking pretext task that does not target the specific structure of the statistical degradation, and our own reimplementation confirmed that it produces dense outputs with prohibitive training times.

The critical research gap lies at the intersection of these paradigms. Table 3.3 summarizes the desirable properties of each method, revealing that no existing approach combines all four.

Table 3.3: Feature comparison of estimation methods. The proposed ColdCov framework is the only method to combine data-driven degradation modeling with full matrix estimation and an SPSD guarantee, though it shares DeepGraph’s reliance on synthetic training data.

Method	Data-Driven	Models Degradation	Needs Synthetic Data	Full Matrix	SPSD Guarantee
Ledoit-Wolf	×	×	×	✓	✓
Graphical Lasso	×	×	×	✓	✓
StepGraph	×	×	×	✓	×
DeepGraph	✓	×	✓	×	×
SSCE	✓	×	×	✓	×
ColdCov (Ours)	✓	✓	✓	✓	✓

What is missing is a method that combines four critical properties: data-driven learning capable of learning complex, flexible priors directly from data; principled degradation

modelling that explicitly models the statistical degradation process caused by finite sampling; full matrix estimation that outputs the complete covariance structure rather than just an adjacency graph; and an architectural guarantee that the output is always a valid (symmetric positive semi-definite) covariance matrix. No existing work has conceptualized covariance estimation as a generative denoising task aimed at reversing a known, deterministic, but non-Gaussian corruption process. This represents not just a methodological gap but a distinct conceptual reframing: from compensating for noise through regularization to actively learning to reverse its effects.

Like DeepGraph, our proposed ColdCov method requires a large amount of synthetic data to learn the generative prior. While it avoids the need for ground-truth labels during real-world inference, training on synthetic covariance structures minimizes empirical risk at the cost of introducing a potential simulation-to-reality gap.

It is worth noting that the self-supervised learning paradigm pioneered by SSCE, despite its current practical limitations, points toward a promising direction for future work. The ability to adapt a generative model to unseen real-world data through self-supervised fine-tuning could address the simulation-to-reality gap that limits supervised approaches. We revisit this opportunity in Chapter 6.

Table 3.4 lists the principal software packages available for covariance estimation, several of which are used in the experimental evaluation presented in Chapter 5.

Table 3.4: Software packages commonly used for covariance and precision matrix estimation. Packages marked with † are used in this thesis.

Package	Language	Method(s)	Reference
scikit-learn [†]	Python	LedoitWolf, GraphicalLassoCV, OAS, EmpiricalCovariance	Pedregosa et al. (2011)
skggm	Python	Extended Graphical Lasso (adaptive, group penalties)	Laska e Narayan (2017)
FastStepGraph [†]	R	Step-Graph (forward-backward selection)	Colonna (2023)
nilearn	Python	Connectivity estimation, brain atlas tools	Abraham et al. (2014)
DeepGraph [†]	Python/PyTorch	CNN-based graph discovery (custom impl.)	Belilovsky et al. (2017)
ColdCov [†]	Python/PyTorch	Cold Diffusion denoising (UNet2DModel, SPSD constraint)	This thesis

3.4 Chapter Summary

This literature review has systematically examined the current state of covariance estimation research, from traditional statistical regularization methods to modern machine learning approaches, reporting concrete results, computational costs, and baselines for each. The comparison tables (Tables 3.2 and 3.3) and method taxonomy (Figure 3.3) synthesize these findings, making the research gap visible at a glance: no existing method combines data-driven learning, explicit degradation modelling, and guaranteed SPSD output for full matrix estimation. This gap motivates our novel contribution: a generative denoising framework that learns to reverse the statistical degradation, utilizing synthetic data to capture complex priors while avoiding the need for ground-truth labels during real-world inference.

It is important to clarify that this work is not a mere application of Cold Diffusion to covariance matrices. It adapts the Cold Diffusion paradigm to a statistically structured inverse problem by defining finite-sample covariance estimation itself as the **degradation process**. In standard Cold Diffusion, the degradation may be blur, masking, downsampling, or similar transformations. Here, the degradation is not arbitrary: it is the sampling-induced transformation from a true covariance matrix to a noisy sample covariance matrix, governed by finite-sample statistics. This makes the method a domain-specific generative estimator rather than a direct reuse of an existing image-restoration framework.

Having established both the theoretical foundations (Chapter 2) and the research context (this chapter), we now turn to Chapter 4, where we present our methodology in detail, showing how Cold Diffusion can be adapted to solve this fundamental statistical problem.

A Generative Denoising Framework for Covariance Estimation

This chapter presents the core methodological contribution of this thesis, the ColdCov framework, a generative denoising approach to covariance estimation. Building on the theoretical foundations of Chapter 2 and the research gap identified in Chapter 3, we develop a principled solution that explicitly models and reverses the statistical degradation process inherent in finite sampling.

We adopt an Empirical Risk Minimization (ERM) approach by training our model on synthetic data where the ground truth graph structure is known, allowing us to directly minimize the error against the true distribution. This learnt model is then transferred to real-world data, under the assumption that the fundamental structure of covariance degradation is consistent across domains.

We begin by outlining our experimental pipeline for rigorous evaluation. We then present our main contribution: a Constrained Cold Diffusion framework that learns to reverse non-Gaussian statistical degradation while guaranteeing mathematical validity of the output. Throughout, we emphasize the key innovations that distinguish our approach from existing methods.

4.1 Data Generation and Experimental Pipeline

To rigorously evaluate our proposed methods, we establish a comprehensive experimental pipeline utilizing both synthetic and real-world data. Synthetic data provides a controlled environment with a known ground truth, essential for quantitative validation, while real-world data allows us to assess the practical applicability and robustness of our framework. The complete implementation of this pipeline, including data generation scripts, model training configurations, and evaluation code, is available in the public repository

at https://github.com/xrhd/cold_cov_diffusion. The high-level pipeline consists of four stages: (1) synthetic data generation with controlled properties, (2) statistical degradation via finite sampling, (3) model training to reverse the degradation, and (4) evaluation against established baselines across multiple metrics.

4.1.1 Synthetic Data Generation

The generation of synthetic datasets with controlled properties serves as the cornerstone of our methodological approach. The process is designed to create realistic, sparse, symmetric positive-definite (SPSD) covariance matrices that mimic structures found in real-world systems. This pipeline, illustrated in Figure 4.1, allows for a systematic and objective evaluation of all estimation methods.

The core of our data generation is the `CovarianceMatrixDataset` class, which leverages the `make_sparse_spd_matrix` function from the `scikit-learn` library. For each sample, a sparse precision matrix is first generated with a high degree of sparsity (specified by $\alpha=0.98$). This precision matrix is then inverted to obtain the ground truth covariance matrix, Σ_{true} . To ensure all variables are on a comparable scale, this Σ_{true} matrix is subsequently normalized to have a unit diagonal, effectively converting it into a true correlation matrix. This serves as the "clean" ground truth for our experiments.

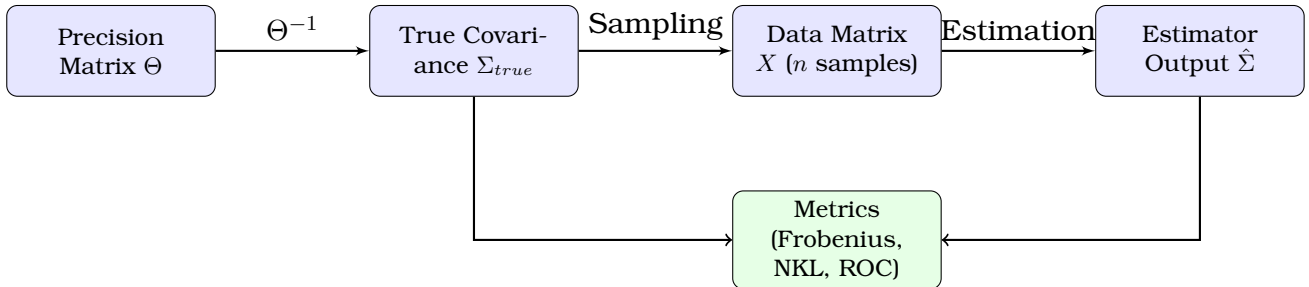


Figure 4.1: The synthetic data generation and evaluation pipeline. A sparse precision matrix Θ generates a true covariance Σ_{true} . Data X is sampled from $\mathcal{N}(0, \Sigma_{true})$, used to compute estimates $\hat{\Sigma}$, which are then quantitatively compared against the ground truth.

4.1.2 Evaluation Strategy for Synthetic Data

For experiments conducted on synthetic data, we have the distinct advantage of knowing the Σ_{true} matrix. This allows for a direct and quantitative evaluation of each estimation method's accuracy. Our primary evaluation strategy is to measure the element-wise discrepancy between the estimated covariance matrix, $\hat{\Sigma}$, and the true matrix.

We employ two principal metrics for this purpose. First, the Mean Absolute Error (L1 Error) calculates the average absolute difference between the elements of the estimated and true matrices. It is a robust measure of overall reconstruction error and is particularly sensitive to sparse structures. Second, the Frobenius Norm computes the square root of the sum

of the absolute squares of the element-wise differences, providing a measure of the total magnitude of the error between the two matrices.

By using these direct comparison metrics, we can rigorously and objectively quantify the performance of our ColdCov framework against baseline methods in recovering the true underlying covariance structure from noisy samples.

4.1.3 Real-World Datasets and Preprocessing

To validate our framework’s performance in practical scenarios, we utilize a real-world dataset from the field of neuroscience, as previously introduced in Section 2.3. The primary dataset for this purpose is the ABIDE dataset, providing resting-state fMRI data for brain connectivity analysis.

Prior to use in our models, this dataset undergoes a standardized preprocessing pipeline. This includes motion correction, normalization, time-series extraction from predefined regions of interest (ROIs), exclusion criteria for subjects with excessive motion artifacts, and leakage control procedures to prevent information flow between training and test sets. The complete preprocessing pipeline code and configuration is available in the public repository at https://github.com/xrhd/cold_cov_diffusion. The final step is the computation of the empirical covariance matrix, which serves as the noisy input to both the baseline and our proposed denoising models.

4.1.4 Evaluation Strategy for Real-World Data

Since the true population covariance is unknown for real-world data, we cannot use direct comparison metrics like L1 or Frobenius error. Instead, our evaluation strategy shifts to measuring the downstream discriminative utility of the estimated models: how well the structural features encoded in each estimator’s precision matrix support a clinically relevant classification task.

Feature Extraction from Precision Matrices

For each subject, we first compute the precision matrix $\hat{\Theta} = \hat{\Sigma}^{-1}$ from the estimated covariance matrix. We then extract the upper triangular elements of $\hat{\Theta}$ (excluding the diagonal) as a feature vector. These $\binom{p}{2}$ features encode the pairwise conditional dependencies between brain regions and serve as the input representation for classification. This approach treats covariance estimation as an upstream feature extraction step, directly testing whether better estimation translates to more informative features.

Discriminative Classification Protocol

We evaluate feature quality through a subject-level binary classification task, distinguishing Control subjects from those with Autism Spectrum Disorder (ASD), using 5-fold Stratified Cross-Validation with an L2-regularized Logistic Regression classifier. Stratification ensures that both diagnostic groups are proportionally represented in each fold, preventing evaluation bias. Performance is measured using three complementary metrics:

- **Accuracy:** the proportion of correctly classified subjects, providing an intuitive overall measure.
- **ROC-AUC:** the Area Under the Receiver Operating Characteristic Curve, measuring the classifier’s ability to discriminate between groups across all decision thresholds.
- **Average Precision (AP):** the area under the Precision-Recall curve, which is particularly informative when diagnostic groups are imbalanced.

It is important to distinguish these metrics from the identically named metrics used in the synthetic evaluation (Section 5.2). In the synthetic setting, ROC-AUC and AP measure edge-level link prediction performance: how well the estimated precision matrix recovers the true graph topology. In the real-world evaluation, the same metrics measure subject-level classification performance: how well precision matrix features distinguish diagnostic groups. Despite sharing names, these metrics operate at fundamentally different levels of analysis.

ColdCov Threshold Optimization

ColdCov produces comparatively dense precision matrix estimates, which may include weak conditional dependencies that add noise to the feature representation. To address this, we apply a sparsity threshold τ that sets entries of the precision matrix with absolute value below τ to zero. The optimal threshold is selected via a nested cross-validation procedure: for each outer fold in the evaluation, an inner grid search over candidate thresholds is performed using only the training data, selecting the value that maximizes Average Precision; the selected threshold is then fixed and applied once to the held-out test fold. This nested procedure prevents information leakage and ensures that reported ColdCov (Optimized) results reflect generalization to unseen subjects.

4.2 The ColdCov Framework

The central insight motivating this framework is to view the relationship between the true covariance matrix Σ and its empirical estimate $\hat{\Sigma}$ through the lens of a statistical degradation process. Rather than treating the Sample Covariance Matrix merely as a

noisy estimate, we model it as the output of a deterministic transformation that corrupts the true matrix through finite sampling. This degradation can be formally expressed as $\hat{\Sigma} = D(\Sigma, n) + \epsilon$, where $D(\cdot, n)$ represents the degradation operator parameterized by sample size. As demonstrated empirically in Chapter 5, this process produces non-Gaussian residuals, motivating the use of the Cold Diffusion framework over standard Gaussian diffusion models.

Building on this perspective, the ColdCov framework applies Constrained Cold Diffusion to the problem of covariance estimation. This approach addresses the fundamental limitations of traditional methods by explicitly modeling the statistical degradation process and learning to reverse it through a generative framework. Unlike regularization-based approaches that indirectly compensate for noise, ColdCov directly learns the inverse of the degradation operator while guaranteeing mathematical validity of the output.

4.2.1 Modeling Statistical Degradation for Cold Diffusion

The core innovation of ColdCov is to formally define the creation of the SCM as a deterministic degradation process suitable for the Cold Diffusion framework. Rather than treating the statistical degradation as an abstract corruption, we model the concrete physical process of statistical sampling.

We define a series of discrete timesteps $t \in \{0, 1, \dots, T\}$, where $t = 0$ represents the clean ground truth Σ_{true} and $t = T$ represents the most severely degraded version. The degradation operator $D(\Sigma_{true}, t)$ modulates the number of samples n_t used to compute the SCM, with n_t decreasing as t increases.

Our degradation schedule uses a geometric progression that provides finer resolution in the low-sample regime, precisely where estimation is most challenging. This creates a principled, non-Gaussian degradation process that accurately reflects real-world covariance estimation from limited data, distinguishing our approach from standard Gaussian diffusion models. The complete pseudo-code for this operator is given in Appendix C.1.

The Sample Schedule Function

To bridge the abstract Cold Diffusion formulation with the concrete statistical sampling process, we define a sample schedule function $n(t)$ that maps each diffusion timestep t to a specific number of observations. This function makes the degradation operator $D(\Sigma, t)$ equivalent to $D(\Sigma, n(t))$, where $n(t)$ determines the sample size used to generate the empirical covariance matrix at step t .

We define $n(t)$ as the following piecewise function:

$$n(t) = \begin{cases} \infty & \text{if } t = 0 \quad (\text{population covariance } \Sigma_{true}) \\ \lfloor N_{\max} \cdot \gamma^t \rfloor & \text{if } t \in [1, T] \end{cases} \quad (4.1)$$

The boundary parameters are set as follows. The upper bound $N_{\max} = 32p$ corresponds to a logarithmic sampling ratio of $k = \log_2(N/p) = 5$, which we identify as the optimal operating point where the Frobenius estimation error is minimized (the empirical justification for this choice is presented in Section 5.1.1). The lower bound $N_{\min} = \lfloor p/4 \rfloor$ corresponds to $k = -2$, placing the most degraded state deep within the rank-deficient regime where $n \ll p$ and the sample covariance matrix is singular. For the illustrative dimension $p = 120$, these yield $N_{\max} = 3,840$ and $N_{\min} = 30$.

We note that the primary experiments in this thesis use $p = 40$ rather than $p = 120$. This choice reflects three considerations: (1) $p = 40$ corresponds to the dimensionality of standard functional brain atlases used in neuroimaging studies, making it a representative testbed for the target application domain; (2) the moderate dimensionality allows thorough exploration of the n/p ratio across distinct statistical regimes while keeping generative model training computationally efficient; and (3) the sampling ratio analysis in Section 5.1.1 demonstrates that the critical thresholds ($k = 0, 3, 5$) are dimension-independent, observed consistently across $p \in \{40, 80, 120, 160\}$. The scalability of ColdCov to higher dimensions is discussed as a key future research direction in Sections 6.4.3 and 6.4.

The geometric decay rate γ is derived by requiring that $n(T) = N_{\min}$, giving the closed-form expression:

$$\gamma = \left(\frac{N_{\min}}{N_{\max}} \right)^{\frac{1}{T-1}} \quad (4.2)$$

For $T = 1,000$ timesteps, this yields $\gamma \approx 0.995$, producing a smooth geometric decay from $n(1) = 3,821$ to $n(1,000) = 30$.¹

The choice of geometric (rather than linear or polynomial) decay is principled: since the variance of the Wishart-distributed sample covariance matrix scales as $\mathcal{O}(1/n)$, the estimation error exhibits a multiplicative structure across sample sizes. A geometric schedule distributes timesteps uniformly on the logarithmic scale, ensuring that the model encounters equally informative degradation gradients throughout training. A linear schedule would disproportionately allocate timesteps to the high-sample regime, where estimation is already accurate, while under-representing the low-sample regime where the model’s learned prior is most critical. This sampling schedule is visualized in Figure 4.2.

The complete degradation process is illustrated in Figure 4.3.

¹In the algorithmic implementation, the schedule is computed via `numpy.geomspace` to generate T logarithmically spaced values between $N_{\min}$ and $N_{\max}$, which are then reversed and cast to integers. The closed-form expression presented here is the mathematical equivalent; integer rounding may produce at-most-unit differences at individual timesteps.

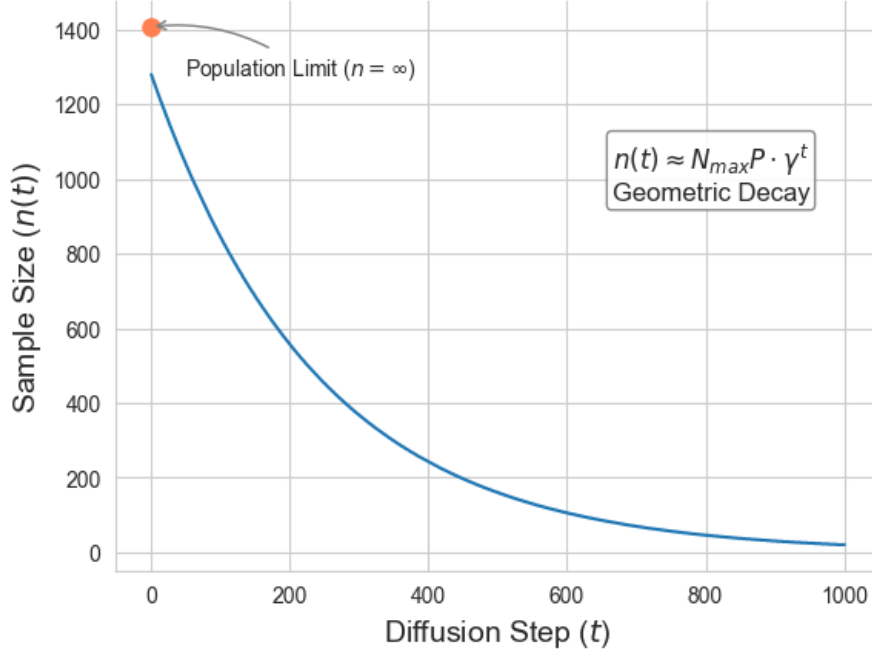


Figure 4.2: The Cold Diffusion sampling schedule $n(t)$ over $T = 1,000$ diffusion steps. The geometric decay from $N_{\max} = 32p$ to $N_{\min} = \lfloor p/4 \rfloor$ distributes timesteps uniformly on the logarithmic scale. The orange marker at $t = 0$ denotes the population limit ($n = \infty$), where the degradation operator returns the true covariance Σ_{true} .

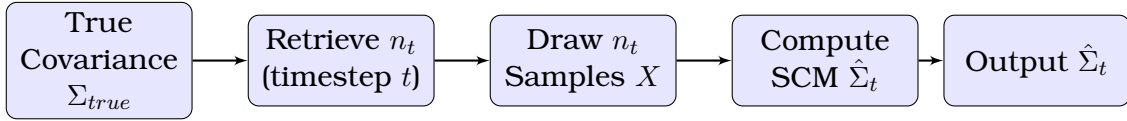


Figure 4.3: The deterministic statistical degradation process. The operator $D(\Sigma_{true}, t)$ generates a degraded matrix by simulating the sampling process ($X \sim \mathcal{N}(0, \Sigma_{true})$) with sample size n_t .

4.2.2 Constrained Architecture for Valid Covariance Matrices

The ColdCov model builds upon the U-Net architecture but introduces a crucial innovation: an architectural constraint that guarantees the output is always a mathematically valid covariance matrix. While the U-Net processes the degraded input matrix and timestep information through its encoder-decoder structure, our key contribution lies in the final output layer that enforces the symmetric positive semi-definite (SPSD) property by construction.

The SPSP Constraint

Rather than hoping the network learns to produce valid covariance matrices implicitly, we enforce this constraint through architectural design, as visualized in Figure 4.4.

This architectural constraint represents a powerful form of inductive bias. In machine learning, inductive bias refers to the set of assumptions that the learner uses to predict

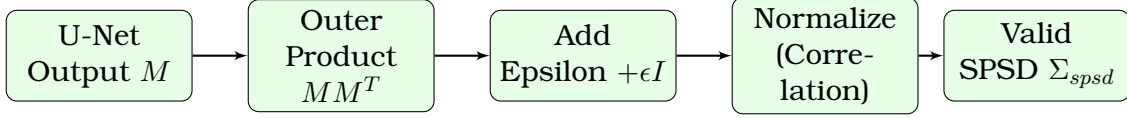


Figure 4.4: The architectural SPSP constraint block. This differentiable module transforms the raw U-Net output into a guaranteed valid symmetric positive semi-definite correlation matrix.

outputs for inputs it has not encountered. By strictly enforcing the SPSP property, we restrict the hypothesis space to the manifold of valid correlation matrices. A formal proof that the Cholesky construction guarantees positive semi-definiteness, together with the convexity properties of the SPSP cone, is provided in Appendix A.1. This explicit bias, combined with the implicit bias towards sparsity encouraged by our loss function (Section 4.2.3), dramatically improves learning efficiency and eliminates the risk of generating invalid covariance matrices.

4.2.3 Model Training

The training procedure for ColdCov follows the Cold Diffusion framework with adaptations for our statistical setting. The process learns to reverse a Markov chain of degradations, as shown in Figure 4.5.

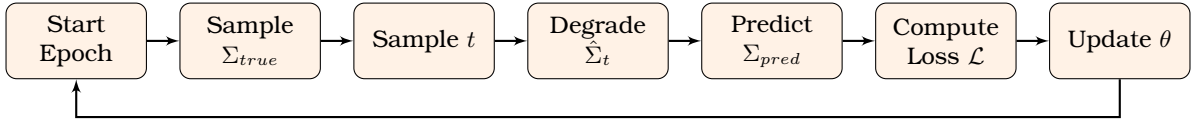


Figure 4.5: The iterative training loop detailed in a horizontal flow. The model learns to map degraded matrices $\hat{\Sigma}_t$ back to the clean ground truth Σ_{true} by minimizing the empirical risk on synthetic data.

A key contribution of this work is the design of a Composite Loss Function that jointly optimizes for accurate covariance estimation and precise structural recovery. We define the loss function $\mathcal{L}$ as:

$$\mathcal{L} = \underbrace{\|\Sigma_{pred} - \Sigma_{true}\|_1}_{\text{Covariance Loss}} + \underbrace{\text{clip}\left(\|\Sigma_{pred}^\dagger - \Sigma_{true}^\dagger\|_1, 0, 1\right)}_{\text{Precision Loss}} \quad (4.3)$$

This composite loss addresses two critical needs. First, the Covariance Loss minimizes the L1 error of the covariance matrix itself, promoting element-wise accuracy and sparsity in the covariance domain. Second, the Precision Loss explicitly targets the precision matrix ($\Theta = \Sigma^{-1}$), which encodes the graph structure. Since Σ_{pred} typically becomes full rank only as training progresses, we use the Moore-Penrose pseudo-inverse ($\dagger$) to ensure numerical stability during early training phases. The gradient clipping prevents loss explosion when the estimate is ill-conditioned.

This dual objective ensures that the model not only reconstructs the covariance values but also recovers the underlying conditional independence structure essential for graphical modeling. The full training algorithm, including the degrade-restore cycle and loss computation, is detailed in Appendix C.2.

4.2.4 Denoising and Inference

During inference, ColdCov reverses the degradation process to recover clean covariance structure from noisy empirical matrices. While the model is trained on varying degradation levels, inference typically begins with the most degraded state corresponding to the actual sample size.

For computational efficiency, we primarily employ single-step inference:

$$\Sigma_{restored} = R_{\theta}(\hat{\Sigma}_{emp}, t_{max}) \quad (4.4)$$

where $\hat{\Sigma}_{emp}$ is the empirical covariance matrix and t_{max} corresponds to the highest degradation level. This approach achieves high-quality reconstructions in a single forward pass, though the framework also supports iterative refinement for potentially higher quality reconstructions.

4.2.5 Computational Complexity

The computational cost of ColdCov inference is dominated by a single forward pass through its purely convolutional U-Net backbone (Ronneberger et al., 2015). We can formalize this using standard CNN time complexity analysis (Shah e Bhavsar, 2022). For a network with L layers, the total number of operations scales as:

$$O\left(\sum_{l=1}^L c_{l-1} \cdot s_l^2 \cdot c_l \cdot m_l^2\right) \quad (4.5)$$

where c_{l-1} is the number of input channels, s_l is the spatial size of the convolution kernel, c_l is the number of output channels, and m_l is the spatial dimension of the output feature map at layer l . Because ColdCov processes a covariance matrix of dimension p as a $p \times p$ grid, the maximum spatial dimension at any layer is bounded by $m_l \leq p$, meaning $m_l^2 \leq p^2$. Since c_{l-1} , s_l , c_l , and the total number of layers L are architecture-specific constants fixed at design time, they do not scale with the data dimension. The inference complexity of the current ColdCov architecture therefore simplifies to $O(p^2)$. Alternative architectures that incorporate global self-attention mechanisms would raise this cost substantially; we derive the resulting $O(p^4)$ bound and discuss mitigation strategies in Appendix A.4.

To place this cost in context, consider the baseline estimators evaluated in this thesis. Ledoit-Wolf shrinkage admits a closed-form solution whose cost is dominated by the com-

putation of the sample covariance matrix itself, scaling as $O(p^2n)$ where n is the number of observations. Graphical Lasso, by contrast, solves an iterative optimization problem: each block coordinate descent iteration requires a Cholesky factorization costing $O(p^3)$ (Friedman et al., 2008), and convergence typically requires 5 to 50 iterations. In practice, the regularization parameter λ must be selected via k -fold cross-validation over a grid of $|\Lambda|$ candidates, multiplying the total cost by $k \times |\Lambda|$. For moderate dimensions and standard grids, this cross-validation overhead can dominate the overall wall-clock time of Graphical Lasso.

ColdCov rebalances this computational profile by shifting the heavy computation to the offline training phase. Training the U-Net on synthetic data requires GPU resources and multiple epochs, but this cost is incurred once and amortized over all subsequent applications; a formal analysis of training cost is beyond the scope of this section, as the key observation is that it is paid independently of the number of subsequent inference queries. At inference time, ColdCov produces an estimate in a single, deterministic forward pass with no iterative convergence loop, no cross-validation search, and no runtime hyperparameter tuning. This predictable, constant-time inference contrasts with the variable runtime of iterative methods, whose wall-clock cost depends on convergence behavior and grid-search configuration. For applications where many covariance estimates must be produced from new data (e.g., analyzing multiple subjects in a neuroimaging study), the amortized cost of ColdCov can therefore be competitive with, or lower than, that of cross-validated iterative methods.

4.3 Baseline Method Implementation

To rigorously evaluate the performance of our proposed denoising frameworks, we compare them against four diverse baseline methods. These baselines encompass standard approaches from the statistical literature, including both shrinkage and iterative structural learning methods, as well as a pioneering deep learning approach. The statistical baselines utilize robust implementations from the `scikit-learn` library and the `FastStepGraph` package in R, while the deep learning baseline relies on a custom PyTorch implementation.

4.3.1 Ledoit-Wolf Shrinkage

The Ledoit-Wolf estimator is a leading shrinkage method that provides a well-conditioned, dense estimate of the covariance matrix. It operates by optimally shrinking the noisy sample covariance matrix towards the identity matrix, effectively regularizing its eigenvalue structure. This produces an invertible and more stable estimate, particularly in $n \leq p$ scenarios. We use the `sklearn.covariance.LedoitWolf` implementation directly on the empirical covariance matrix. This baseline serves to evaluate the performance of our methods against a non-sparse, regularization-based approach.

4.3.2 Graphical Lasso

The Graphical Lasso is the standard benchmark for sparse precision matrix estimation. As detailed in Section 3.2.1, it encourages sparsity by applying an L1 penalty to the maximum likelihood objective. We use the `sklearn.covariance.GraphicalLassoCV` implementation. The regularization parameter, α , which controls the degree of sparsity, is a critical hyperparameter. For our experiments, this parameter is selected via data-driven cross-validation (`GraphicalLassoCV`) rather than manual configuration, ensuring a robust and fair baseline. This baseline is crucial for assessing how well our denoising frameworks recover sparse structures compared to the industry-standard method designed for that exact purpose.

4.3.3 StepGraph

To include a modern, stepwise iterative approach, we implemented StepGraph using the `FastStepGraph` algorithm implemented by Colonna (2023). This method constructs the precision matrix by iteratively adding and removing edges to maximize a score function, offering an alternative to the L1-regularized global optimization of GLasso. StepGraph is particularly relevant as a rigorous baseline for graph structure recovery, allowing us to compare our generative approach against a dedicated structural learning algorithm.

4.3.4 DeepGraph

Complementing the statistical methods, we also compare against DeepGraph (Belilovsky et al., 2017), a pioneering deep learning approach for graphical model selection. We implemented the network architecture as described in the original work: a multi-layer Convolutional Neural Network (CNN) that treats the empirical covariance matrix as a 2D image input. The network is trained in a supervised manner to predict the conditional independence structure (adjacency matrix). This baseline is essential for distinguishing the benefits of our generative diffusion-based approach from standard discriminative deep learning methods applied to the same problem domain.

4.4 Chapter Summary

This chapter has presented the ColdCov framework, our primary contribution to covariance estimation. We have shown how Constrained Cold Diffusion can be adapted to explicitly model and reverse the statistical degradation process inherent in finite sampling. The key innovations include: (1) a principled degradation operator that accurately reflects the degradation induced by data scarcity, (2) an architectural SPSD constraint that guarantees

mathematical validity, and (3) a training procedure that learns to reverse non-Gaussian degradation.

Having established this methodology and the baseline methods for comparison, we now turn to Chapter 5, where we present comprehensive experiments evaluating our approach against established baselines across synthetic and real-world datasets.

Experiments and Results

This chapter presents comprehensive empirical validation of the ColdCov framework developed in Chapter 4. Our primary objective is to validate the central hypothesis: that by explicitly modeling and reversing the statistical degradation induced by data scarcity, ColdCov achieves strong performance in covariance estimation compared to traditional baseline methods. We structure our evaluation into four main components: (1) rigorous quantitative and qualitative analysis on synthetic data with known ground truth, (2) practical validation on real-world neuroimaging datasets, (3) statistical analysis of the degradation process to justify our methodological choices, and (4) ablation studies to validate key architectural components. This evaluation design allows us to identify specific settings where ColdCov improves over traditional baselines, as well as settings where sparsity-based methods remain preferable.

5.1 Experimental Setup

To ensure a fair, rigorous, and reproducible comparison, all experiments were conducted within a unified computational pipeline. This framework allows us to systematically evaluate the performance of the proposed ColdCov estimator against established baselines under identical conditions. This section details the data generation processes, the specific architectural configurations of our models, and the training protocols employed.

5.1.1 Datasets and Degradation Schedule

Our experiments primarily leverage the synthetic data generation pipeline described in Section 4.1, designed to emulate the statistical properties of functional neuroimaging data. We fix the data dimension at $p = 40$. This moderate dimensionality provides a controlled setting that allows thorough exploration of the n/p ratio across distinct statistical regimes (from

severe under-sampling through the invertibility boundary to comfortable over-sampling) while keeping the training of our generative model computationally efficient.

We note that $p = 40$ corresponds to the dimensionality of standard functional brain atlases used in neuroimaging studies and provides a representative testbed for the moderate-dimensional regime. While the thesis motivates the covariance estimation challenge with genomics examples where p can reach 20,000 genes (Chapter 1), evaluating at this scale is beyond the scope of the current work. The scalability of ColdCov to ultra-high dimensions is discussed as a key future research direction in Section 6.4.3 and Section 6.4.

To thoroughly investigate the model’s behavior across these regimes, we implement a Geometric Degradation Schedule. The sample sizes are chosen as a geometric progression: $n \in \{10, 20, 40, 80, 160, 320\}$. With $p = 40$, this schedule spans three distinct regions: the sub-sampled regime ($n < p$, i.e. $n \in \{10, 20\}$), where the sample covariance matrix is rank-deficient and classical estimators break down; the invertibility limit ($n = p = 40$), marking the critical phase transition to full rank; and the over-sampled regime ($n > p$, i.e. $n \in \{80, 160, 320\}$), where traditional statistical guarantees begin to hold. The condition $n = 10$ provides a stress test of extreme degradation ($n = p/4$), while $n = 320$ ($n = 8p$) serves as a control condition well into the asymptotic regime.

Sampling Ratio Analysis and Operating Point Selection

The selection of the schedule boundary parameters $N_{\max}$ and $N_{\min}$ (formally defined in Section 4.2.1) requires a principled analysis of the trade-off between reconstruction fidelity and data efficiency. To this end, we define the logarithmic sampling ratio:

$$k = \log_2 \left(\frac{N}{p} \right) \quad (5.1)$$

which parametrizes the sampling regime relative to the data dimension (equivalently, $N = 2^k \cdot p$). Three distinct regimes emerge: the *sub-sampled regime* ($k < 0$, $N < p$) where the sample covariance matrix is rank-deficient, the *invertibility limit* ($k = 0$, $N = p$) marking the phase transition to full rank, and the *over-sampled regime* ($k > 0$, $N > p$) where classical statistical guarantees begin to hold.

We evaluated the impact of k on two complementary metrics: the Frobenius norm error $\|\hat{\Sigma} - \Sigma\|_F$, where $\hat{\Sigma}$ denotes the empirical sample covariance matrix (SCM), measuring reconstruction fidelity; and the ROC AUC, measuring graph structure recovery. Experiments were conducted across dimensions $p \in \{40, 80, 120, 160\}$ and $k \in [-4, 8]$. The results, shown in Figure 5.1, reveal three dimension-independent critical thresholds summarized in Table 5.1.

Although the Pareto analysis indicates that the marginal efficiency is maximized at $k = 3$ ($N = 8p$), we select $k = 5$ ($N_{\max} = 32p$) as the operating point for three reasons. First,

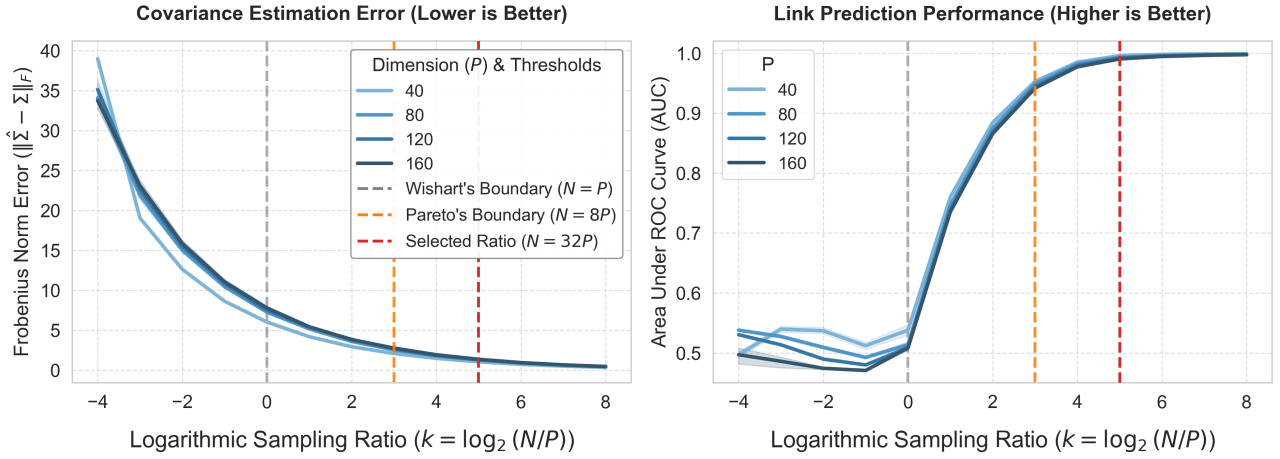


Figure 5.1: Impact of the logarithmic sampling ratio $k = \log_2(N/p)$ on covariance estimation quality. **Left:** Frobenius norm error between the sample covariance matrix and the true covariance, showing exponential decay with increasing k . **Right:** ROC AUC for graph structure recovery (link prediction), saturating rapidly beyond the invertibility limit. Vertical dashed lines mark the three critical thresholds: the Wishart invertibility boundary ($k = 0$, $N = p$), the Pareto knee ($k = 3$, $N = 8p$), and the selected operating point ($k = 5$, $N = 32p$). Results are consistent across all tested dimensions; each curve in the legend corresponds to a different matrix dimension ($p = 40, 80, 120, 160$).

Table 5.1: Critical thresholds identified from the sampling ratio analysis. These thresholds are dimension-independent, observed consistently across all tested values of p .

Threshold	k	N/p	Description
Invertibility Limit	0	1	SCM becomes singular below this point; the precision matrix $\Theta = \Sigma^{-1}$ is undefined.
Pareto Knee	3	8	Onset of diminishing returns in ROC AUC (identified via the Kneedle algorithm).
Operating Point	5	32	Frobenius error minimized; selected as $N_{\max}$.

the Pareto knee marks the boundary of the saturation region, not its interior; operating at this boundary exposes the model’s upper reference point to residual stochastic noise that can degrade precision matrix quality. Second, the Frobenius error continues to decay significantly between $k = 3$ and $k = 5$, and in high-dimensional settings this residual reduction is critical for the numerical conditioning of downstream matrix inversions. Third, for the target application of causal structure inference, the cost of false positives arising from spurious correlations in an under-sampled reference exceeds the computational cost of generating additional synthetic samples. We therefore prioritize estimation stability over sampling efficiency.

5.1.2 Implementation Details of Proposed Models

The ColdCov framework was implemented in PyTorch, leveraging the robust `diffusers` library. The core generative network is a `UNet2DModel`, a U-Net architecture adapted for 2D matrix data. The network consists of a contracting path (encoder) and an expansive path (decoder) with symmetric skip connections. We utilized a lightweight configuration with channel multipliers of (1, 2, 4) and an initial block output of 64 channels, ensuring the model has sufficient capacity to learn the complex manifold of SPSP matrices without overfitting to the limited synthetic training set. The time-embedding, critical for diffusion models, was injected via sinusoidal position embeddings into each residual block.

We evaluated three primary variants to isolate the contributions of our method’s components. First, **ColdCov (SPSP + Custom Loss)**, the proposed model, uses the architectural SPSP constraint and is trained with the composite loss that penalises error in both the covariance and precision domains. Second, **ColdCov (SPSP + Covariance Loss)** uses the same SPSP constraint but is trained with an L1 loss on the covariance matrix alone, without the precision-domain penalty. Finally, **ColdCov (Unconstrained)** uses the composite loss but lacks the architectural SPSP enforcement, allowing outputs that may violate positive semi-definiteness.

Training was performed using the Adam optimizer with a learning rate of 1×10^{-3} and a batch size of 128. Due to the efficiency of the Cold Diffusion formulation (which is deterministic and requires fewer steps than DDPM), models converged rapidly; we trained for 3 epochs, which proved sufficient to stabilize the loss on our synthetic dataset. The rapid convergence is primarily attributable to the scale and density of the training set: following a protocol analogous to MNIST (LeCun et al., 2010), we generated 60,000 training matrices and 10,000 test matrices, each of dimension 40×40 , spanning five distinct graph topologies. In this data-rich regime, the model acquires substantial signal per epoch, enabling effective learning even with a relatively small number of passes over the dataset. All computations were accelerated using Metal Performance Shaders (MPS) on Apple Silicon hardware.

5.1.3 Implementation Details of Baseline Methods

We bench-marked our approach against four diverse estimators. For our statistical baselines, we utilized industry-standard implementations from `scikit-learn`: the Ledoit-Wolf (LW) estimator, an analytically solvable shrinkage estimator Ledoit e Wolf (2004), which serves as a strong baseline for covariance matrix conditioning; and Graphical Lasso (GLasso), the standard for sparse precision matrix estimation, using cross-validated regularisation (`GraphicalLassoCV` with $\alpha \in [0.01, 0.10]$) to ensure a fair, data-driven selection of the sparsity parameter. We also included two modern methods to ensure robust results: **StepGraph**, a stepwise iterative algorithm for GGM estimation, using the `FastStepGraph` implementation (Colonna, 2023); and **DeepGraph**, a supervised deep learning method

trained to regress the adjacency matrix from the empirical covariance, which we implemented following the architecture described by [Belilovsky et al. \(2017\)](#).

5.2 Quantitative Analysis on Synthetic Data

We evaluate performance using three distinct lenses: the element-wise accuracy of the covariance matrix itself (Frobenius Norm), the probabilistic quality of the precision matrix (Normalized KL Divergence), and the accuracy of the recovered graph topology (Link Prediction metrics).

All results in this section are computed on a held-out test set of 10,000 synthetic covariance matrices, uniformly distributed across the four graph topologies introduced in Section 4.1 (Lattice, Preferential Attachment, Random, and Small-World; 2,500 matrices per topology). In the figures that follow, the shaded bands surrounding each line represent 95% bootstrap confidence intervals of the mean, whereas the $\pm$ values reported in the accompanying tables denote standard deviations across the evaluation set. Confidence intervals quantify the uncertainty of the estimated mean performance; standard deviations quantify the spread of individual results.

5.2.1 Covariance Reconstruction Fidelity

The Frobenius Norm of the error matrix, $\|\hat{\Sigma} - \Sigma_{true}\|_F$, measures the Euclidean distance between the estimated and true matrices. It provides a raw summary of reconstruction fidelity, capturing error in both the diagonal (variances) and off-diagonal (covariances) elements. Figure 5.2 visualizes this error as a function of sample size. The shaded region around each curve represents a 95% confidence interval of the mean, computed via non-parametric bootstrapping with 1,000 resamples (the default configuration of the plotting library, with no custom overrides). Where these confidence bands do not overlap between two methods at a given sample size, the performance difference is statistically meaningful; conversely, the width of each band reflects the consistency of the corresponding estimator across the evaluation set.

The results (Table 5.2) reveal a striking pattern: ColdCov achieves the lowest reconstruction error across nearly the entire sample-size range. At $n = 10$, Ledoit-Wolf holds a marginal advantage (7.571 vs 7.884), reflecting the effectiveness of analytic shrinkage in the extreme under-sampling case ($n = p/4$). However, from $n = 20$ onward, ColdCov achieves the lowest Frobenius error at every sample size through $n = 320$.

This result is stronger than one might expect from a prior-based method. Classical theory predicts that, as $n \rightarrow \infty$, likelihood-based estimators such as Graphical Lasso become asymptotically efficient and should eventually overtake regularised approaches. Yet ColdCov maintains its advantage even in the over-sampled regime: at $n = 320$, it achieves

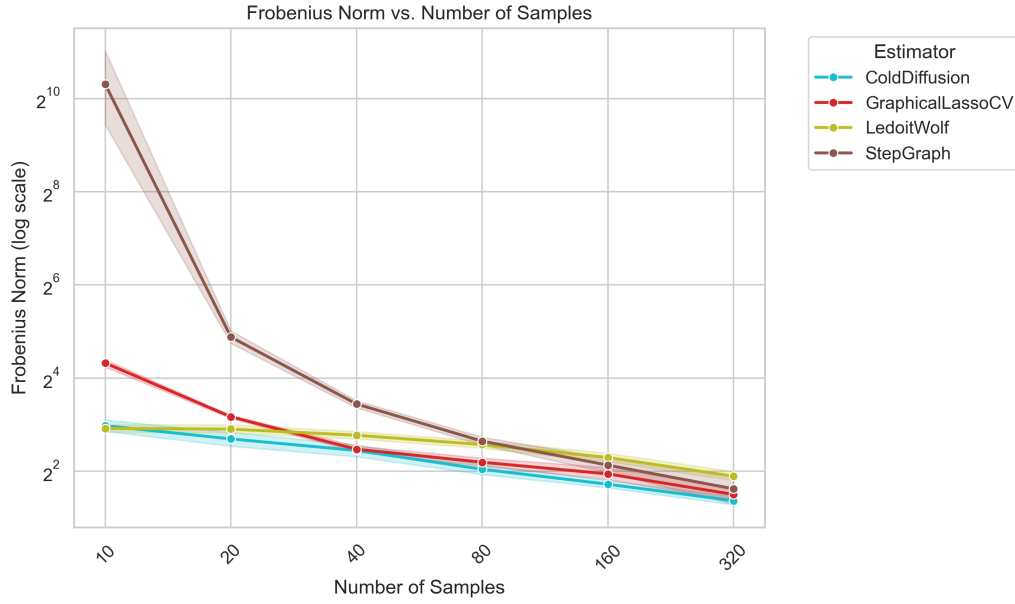


Figure 5.2: Frobenius Norm reconstruction error as a function of sample size. ColdCov achieves the lowest error from $n = 20$ onward, maintaining its advantage across both the sub-sampled and over-sampled regimes.

2.585 versus Graphical Lasso’s 2.835. The generative prior learned during training provides a consistent regularisation benefit that complements the information in the data, rather than competing with it. This strengthens our central hypothesis: the ColdCov denoising framework is not merely a tool for the data-scarce regime, but a broadly effective covariance estimator.

Table 5.2: Frobenius Norm ($\|\hat{\Sigma} - \Sigma\|_F$) Comparison (Lower is better)

Sample Size (n)	ColdCov	Graphical Lasso	Ledoit-Wolf	StepGraph
10	7.884 ± 1.110	20.016 ± 1.412	7.571 ± 0.755	1264.468 ± 1230.941
20	6.475 ± 1.072	9.000 ± 0.381	7.495 ± 0.809	29.452 ± 4.719
40	5.454 ± 0.778	5.555 ± 0.420	6.839 ± 0.643	10.881 ± 1.009
80	4.125 ± 0.551	4.577 ± 0.510	5.953 ± 0.558	6.263 ± 0.663
160	3.302 ± 0.312	3.849 ± 0.633	4.911 ± 0.557	4.385 ± 0.813
320	2.585 ± 0.283	2.835 ± 0.463	3.720 ± 0.415	3.083 ± 1.151

5.2.2 Analysis of Precision Matrix Quality

While the covariance matrix captures marginal dependencies, the precision matrix (inverse covariance, $\Theta = \Sigma^{-1}$) is often of greater scientific interest as it encodes the conditional independence structure, the direct connections in a Graphical Model. To evaluate this, we use the Normalized KL Divergence (NKL), which measures the information loss when approximating the true distribution $\mathcal{N}(0, \Sigma)$ with the estimated $\mathcal{N}(0, \hat{\Sigma})$.

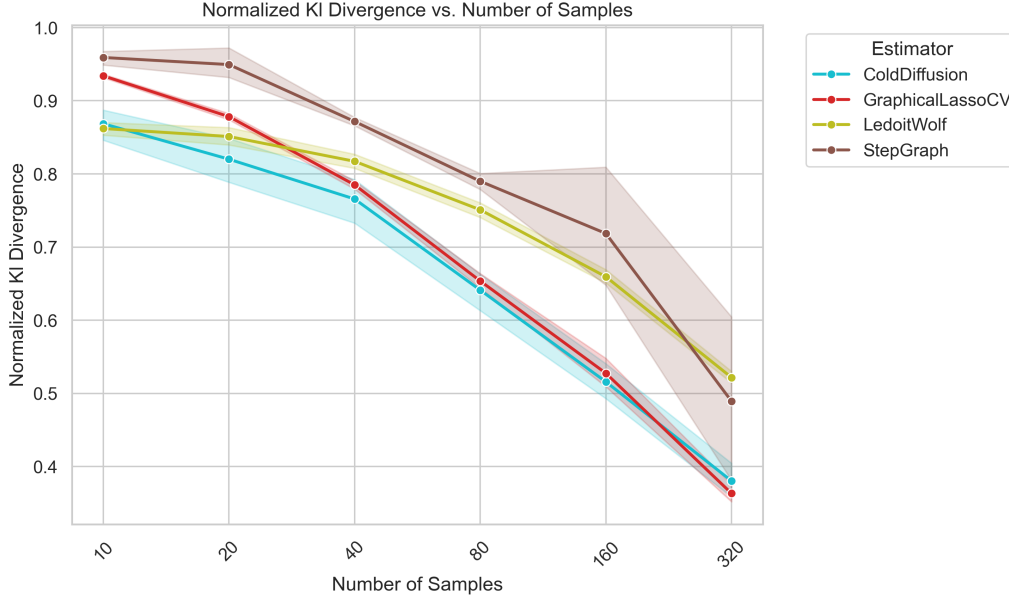


Figure 5.3: Normalized KL Divergence as a function of sample size. ColdCov achieves the lowest divergence from $n = 20$ through $n = 160$, indicating strong precision matrix quality across the sub-sampled and moderate regimes.

The results (Table 5.3) reveal a pattern consistent with the Frobenius findings but with distinct boundary behaviour. ColdCov achieves the lowest NKL from $n = 20$ through $n = 160$, demonstrating that its denoised covariance matrices yield well-conditioned and probabilistically faithful precision matrix estimates across the critical operating range. At $n = 10$, Ledoit-Wolf holds a marginal edge (0.862 vs 0.869), reflecting its analytically optimal shrinkage when data is extremely scarce. At the opposite extreme ($n = 320$), Graphical Lasso overtakes (0.363 vs 0.380), consistent with the asymptotic efficiency of likelihood-based sparse estimation in the data-rich regime.

The practical implication is significant: by explicitly denoising the covariance matrix before inversion via the generative process and enforcing the SPSD constraint, ColdCov ensures that the resulting precision matrix remains well-conditioned even when $n \ll p$. This stability is critical for downstream tasks such as partial correlation computation and causal structure inference.

Table 5.3: Normalized KL Divergence (Lower is better)

Sample Size (n)	ColdCov	Graphical Lasso	Ledoit-Wolf	StepGraph
10	0.869 ± 0.037	0.934 ± 0.003	0.862 ± 0.015	0.959 ± 0.017
20	0.820 ± 0.052	0.878 ± 0.007	0.851 ± 0.021	0.949 ± 0.036
40	0.766 ± 0.050	0.785 ± 0.010	0.817 ± 0.017	0.872 ± 0.010
80	0.641 ± 0.044	0.654 ± 0.017	0.751 ± 0.017	0.790 ± 0.018
160	0.516 ± 0.040	0.528 ± 0.033	0.659 ± 0.016	0.719 ± 0.140
320	0.380 ± 0.039	0.363 ± 0.020	0.522 ± 0.014	0.489 ± 0.184

5.2.3 Graph Topology and Link Prediction

We evaluate the ability to recover the true graph topology using partial correlations derived from the estimated precision matrix. Unlike raw thresholding of the covariance matrix, partial correlations isolate direct (conditional) dependencies between variables, providing a principled basis for link prediction. We quantify performance using two complementary metrics: the ROC AUC (measuring discrimination across all thresholds) and the Average Precision (AP, which is more sensitive to performance on the sparse positive class).

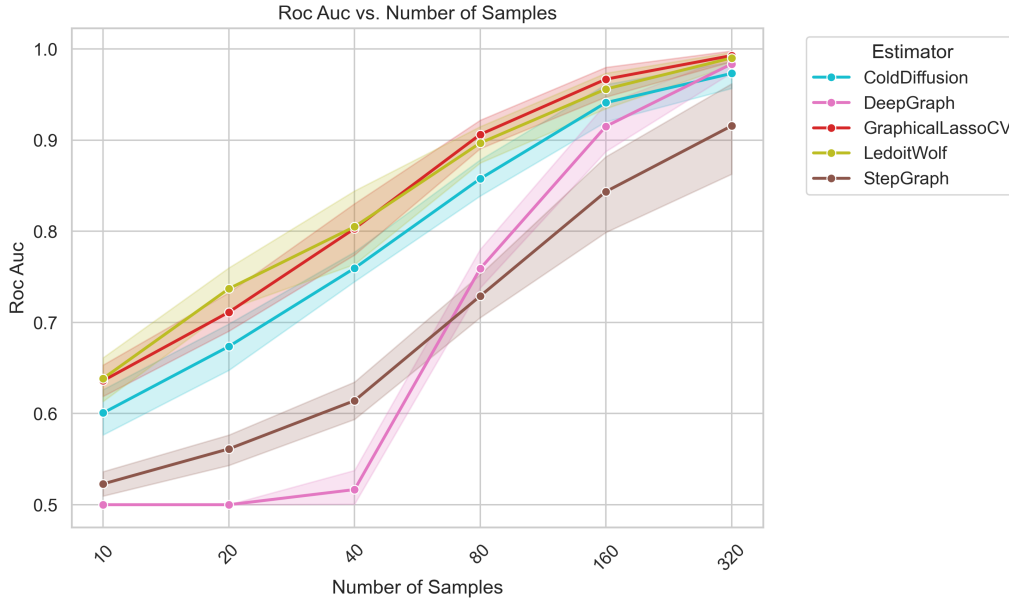


Figure 5.4: ROC AUC for graph structure recovery via partial correlations. Sparse-enforcing methods (Graphical Lasso, Ledoit-Wolf) lead at all sample sizes, with ColdCov consistently outperforming StepGraph and DeepGraph.

Table 5.4: Link Prediction Performance (ROC AUC and Average Precision) via Partial Correlations across Sample Sizes ($n \in \{10, 40, 160\}$)

n	Metric	ColdCov	Graphical Lasso	Ledoit-Wolf	StepGraph	DeepGraph
10	ROC AUC	0.601 ± 0.044	0.636 ± 0.022	0.639 ± 0.039	0.523 ± 0.023	0.500 ± 0.000
	AP	0.198 ± 0.036	0.226 ± 0.042	0.204 ± 0.065	0.120 ± 0.011	0.000 ± 0.000
40	ROC AUC	0.760 ± 0.027	0.803 ± 0.048	0.805 ± 0.068	0.614 ± 0.034	0.517 ± 0.032
	AP	0.485 ± 0.072	0.493 ± 0.115	0.468 ± 0.137	0.252 ± 0.062	0.076 ± 0.068
160	ROC AUC	0.941 ± 0.035	0.967 ± 0.031	0.956 ± 0.033	0.844 ± 0.072	0.915 ± 0.047
	AP	0.837 ± 0.076	0.902 ± 0.070	0.848 ± 0.113	0.710 ± 0.136	0.738 ± 0.112

Table 5.4 reveals a qualitatively different ranking from the reconstruction metrics. Ledoit-Wolf and Graphical Lasso consistently lead ROC AUC at all sample sizes, with Ledoit-Wolf achieving the highest ROC AUC at $n = 10$ and $n = 40$, and Graphical Lasso taking the lead at $n = 160$. These methods produce precision matrices whose partial correlations are better calibrated for edge detection: Graphical Lasso explicitly enforces sparsity in the precision

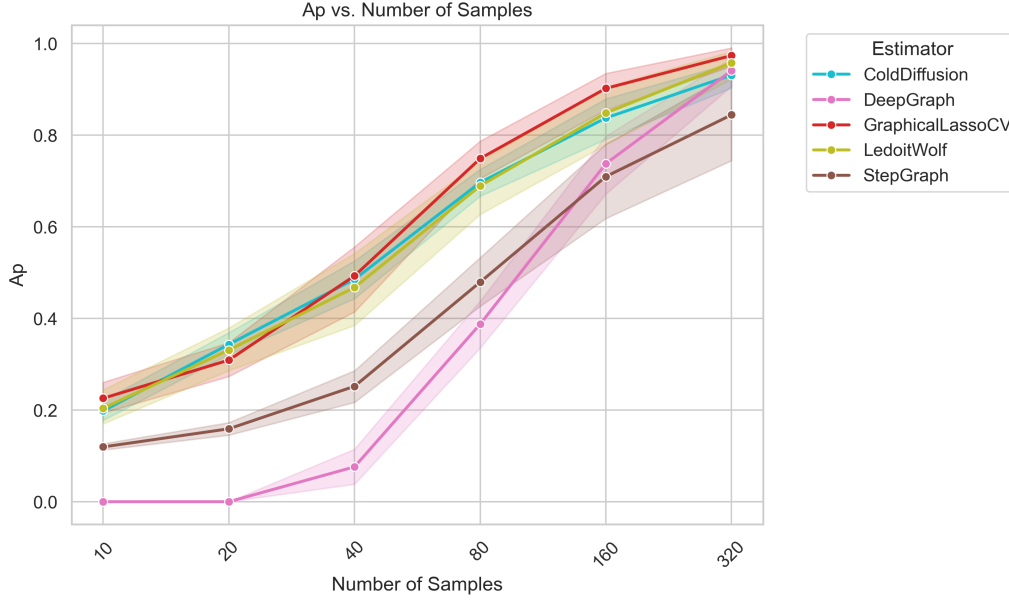


Figure 5.5: Average Precision (AP) for graph structure recovery via partial correlations. AP is sensitive to performance on the sparse set of true edges, confirming the advantage of methods that produce well-calibrated precision matrices.

matrix, while Ledoit-Wolf’s optimal shrinkage yields a well-conditioned inverse even in the sub-sampled regime.

ColdCov, by contrast, produces a comparatively dense covariance estimate whose inverse does not exhibit explicit sparsity. Nevertheless, its link prediction performance remains competitive: at $n = 10$, ColdCov achieves ROC AUC 0.601, substantially above the random baseline (0.500) and well ahead of StepGraph (0.523) and DeepGraph (0.500). As sample size increases, all methods converge, but the gap between ColdCov and the sparse-enforcing leaders narrows (ROC AUC 0.941 vs 0.967 at $n = 160$).

This result highlights a fundamental trade-off. ColdCov’s strength lies in covariance reconstruction fidelity, achieving the lowest Frobenius error and best precision matrix quality (NKL) across most sample sizes. Sparse-enforcing methods, on the other hand, excel at graph structure recovery, where the explicit sparsity pattern in the precision matrix directly aligns with the link prediction objective. These are complementary capabilities on the synthetic evaluation: ColdCov excels at covariance estimation, while Graphical Lasso and Ledoit-Wolf excel at graph structure estimation. Detailed per-topology recall and precision tables are provided in Appendix B.2. However, as we demonstrate in Section 5.4, this synthetic trade-off does not necessarily extend to downstream discriminative tasks on real-world data, where ColdCov’s comparatively dense precision matrix features show the best observed mean performance for subject-level classification.

5.2.4 Topology-Specific Performance Analysis

A deeper dive into the results, stratified by the underlying graph topology (Lattice, Preferential Attachment, Random, Small-World), reveals nuanced performance characteristics that generic averages might obscure. Detailed tables for each topology are provided in Appendix B (specifically, the per-topology reconstruction error tables in Appendix B.1), but we highlight key findings here.

Robustness on Scale-Free Networks (Preferential Attachment)

ColdCov achieves the lowest Frobenius error on Preferential Attachment graphs at every sample size from $n = 10$ through $n = 320$, a result unique among the tested topologies. At $n = 10$, ColdCov reaches 7.242 versus Ledoit-Wolf’s 7.775; at $n = 320$, it achieves 2.415 versus Graphical Lasso’s 3.437. This suggests that ColdCov is particularly adept at modeling “hub” structures (nodes with unusually high degree connectivity), which are characteristic of many real-world biological and social networks. The generative prior effectively captures these heavy-tailed structural distributions better than the uniform ℓ_1 penalty of Graphical Lasso.

Lattice Regularity and Baseline Competitiveness

On highly regular Ring Lattice topologies, the competitive landscape shifts. Graphical Lasso overtakes ColdCov at $n = 40$ (Frobenius 5.393 vs 5.738), and at $n = 320$, StepGraph achieves the lowest error (2.164 vs ColdCov’s 2.374). These graphs have consistent local structure and low variance in node degree; in these “easier” structural regimes, the sparsity-enforcing and stepwise approaches benefit from the regularity, as the simple structure requires fewer samples to identify precisely. Nevertheless, ColdCov remains the strongest performer on Frobenius error across the sub-sampled regime ($n \leq 20$) even on lattice topologies.

5.3 Qualitative Analysis of Matrix Reconstruction

To complement the quantitative metrics presented in the preceding sections, we conduct a qualitative visual comparison of the estimated precision matrices across multiple graph topologies. Figure 5.6 presents a consolidated comparison across four representative topologies (Random, Small-World, Lattice, Preferential Attachment) and five estimation approaches. The columns are arranged from the noisiest baseline on the left to the ground truth on the right: the Empirical (sample) inverse, followed by the regularized baselines Ledoit-Wolf and Graphical Lasso, then ColdCov, and finally the True precision matrix as the reference standard. This ordering facilitates direct visual assessment of how each successive method recovers the underlying conditional independence structure.

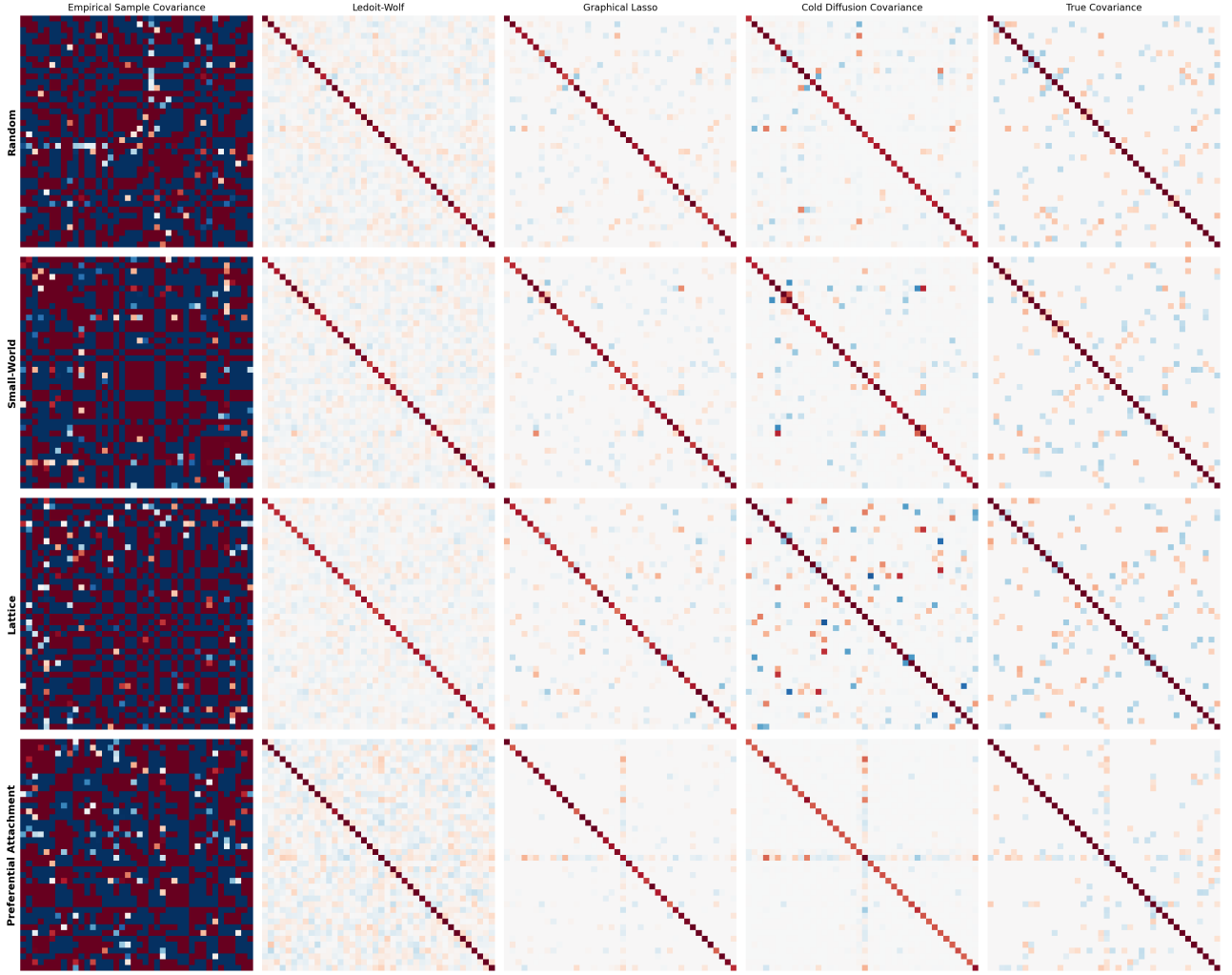


Figure 5.6: Qualitative comparison of precision matrix reconstruction across four graph topologies (rows: Random, Small-World, Lattice, Preferential Attachment) and five estimation methods (columns: Empirical, Ledoit-Wolf, Graphical Lasso, ColdCov, True). Experimental conditions: $p = 40$, $n = 40$, seed = 42. The Empirical sample inverse suffers from pervasive noise at this sample-to-dimension ratio ($n/p = 1$). Ledoit-Wolf applies shrinkage toward the diagonal, dampening off-diagonal noise but losing structural detail. Graphical Lasso promotes sparsity via ℓ_1 regularization, recovering some off-diagonal entries while suppressing weaker edges. ColdCov applies the SPSD-constrained cold diffusion model to denoise the sample covariance, recovering off-diagonal sparsity patterns that visually approximate the true precision matrix.

The visual evidence reinforces the quantitative findings and reveals topology-dependent patterns that aggregate metrics may obscure. Across all four topologies, the Empirical precision matrix is visibly dominated by noise: the off-diagonal region is saturated with spurious non-zero entries that obscure the true conditional independence structure, confirming that the sample-to-dimension ratio $n/p = 1$ is insufficient for reliable direct inversion.

Ledoit-Wolf dampens the magnitude of these spurious entries, producing a cleaner diagonal-dominant appearance; however, it consistently fails to recover the off-diagonal structure, effectively blurring the sparse conditional dependencies into a uniform low-magnitude haze. Graphical Lasso enforces sparsity more aggressively, recovering iden-

tifiable off-diagonal entries in several topologies, but at the cost of suppressing weaker true edges that fall below the regularization threshold.

ColdCov produces a visually distinct pattern across all four topologies. It recovers both the salient off-diagonal conditional dependencies and their relative magnitudes while suppressing the background noise to near-zero levels. The result is a precision matrix that closely approximates the sparse topology and the signed partial correlations of the ground truth.

The structured topologies (Lattice and Small-World) reveal the clearest visual advantage for ColdCov: the regular banding pattern characteristic of Lattice graphs and the clustered connectivity of Small-World graphs are recovered with high fidelity, whereas Graphical Lasso tends to over-sparsify these structured patterns and Ledoit-Wolf obscures them entirely. The denser Random topology presents a greater challenge for all methods, as the lack of repeating structural motifs makes it harder for any estimator to distinguish true conditional dependencies from noise; even in this case, ColdCov achieves the closest visual match to the ground truth. Preferential Attachment graphs, characterized by hub nodes with high-degree connectivity, represent an intermediate case where ColdCov faithfully recovers the hub structure, consistent with the quantitative findings reported in Section 5.2.4.

This qualitative fidelity is crucial for interpretability, particularly in domains such as neuroimaging, where valid conditional dependencies must be distinguished from noise-induced artifacts to draw meaningful biological conclusions.

5.4 Application to Real-World Data: A Neuroimaging Case Study

To validate the translational potential of our method, we applied ColdCov to the ABIDE (Autism Brain Imaging Data Exchange) dataset, a large-scale collection of resting-state fMRI data comprising 870 subjects (402 Control, 468 Autism Spectrum Disorder) across 39 regions of interest (ROIs). Because the ColdCov U-Net architecture was trained specifically on synthetic covariance matrices of dimension $p = 40$, the 39×39 empirical covariance matrices from the ABIDE dataset were accommodated by padding them with zeros to reach the required 40×40 input size, with the padding subsequently removed after the generative denoising pass. This application is critical: while synthetic data proves mathematical correctness, real neuroimaging data contains complex, non-stationary noise and biological variability that simple simulations cannot fully capture. The ABIDE evaluation tests a stronger claim than reconstruction fidelity: whether ColdCov’s estimated precision matrices produce features that are more discriminative for a clinically relevant downstream task.

5.4.1 Discriminative Classification Evaluation

Following the evaluation protocol defined in Section 4.1.4, we extract the upper triangular elements of each subject’s estimated precision matrix $\hat{\Theta}$ as a feature vector encoding pairwise conditional dependencies between brain regions. These features are evaluated through a subject-level binary classification task, distinguishing Control from ASD subjects, using 5-fold Stratified Cross-Validation with L2-regularized Logistic Regression. Performance is reported using Accuracy, ROC-AUC, and Average Precision (AP).

Unlike the synthetic evaluation in Section 5.2, where ROC-AUC and AP measure edge-level link prediction (recovering the graph topology), these metrics here measure subject-level classification: the ability to distinguish diagnostic groups based on the structural information encoded in the precision matrix. This distinction is fundamental: an estimator can produce a precision matrix with poor edge-level topology recovery yet generate features that are highly discriminative for downstream classification.

For ColdCov, we additionally report a “ColdCov (Optimized)” variant that applies the sparsity threshold optimization described in Section 4.1.4 (see 4.1.4), selecting the threshold via grid search on the training folds to retain only the most discriminative conditional dependencies.

5.4.2 Classification Results

Table 5.5: Subject-level classification performance on the ABIDE dataset (870 subjects, 39 ROIs). All metrics are computed via 5×5 Repeated Stratified Cross-Validation (25 folds) with L2 Logistic Regression. Higher is better. Best values in **bold**.

Method	Accuracy (%)	ROC-AUC	AP
Graphical Lasso	59.8 ± 2.3	0.634 ± 0.029	0.673 ± 0.045
StepGraph	53.9 ± 1.5	0.549 ± 0.025	0.574 ± 0.017
Ledoit-Wolf	57.9 ± 1.5	0.606 ± 0.043	0.642 ± 0.034
ColdCov	61.7 ± 4.4	0.659 ± 0.049	0.678 ± 0.049
ColdCov (Optimized)	62.1 ± 4.4	0.659 ± 0.049	0.678 ± 0.051

Table 5.6: Wilcoxon signed-rank test results comparing ColdCov (Optimized) against each baseline across repeated 5×5 stratified cross-validation (25 paired observations per comparison). p -values are Holm–Bonferroni corrected for 9 simultaneous tests. Significance: ** $p < 0.01$, * $p < 0.05$, ns = not significant.

Baseline	Metric	W	p (corrected)	Sig.
Graphical Lasso	Accuracy	127.0	1.0000	ns
Graphical Lasso	ROC-AUC	137.0	1.0000	ns
Graphical Lasso	AP	162.0	1.0000	ns
StepGraph	Accuracy	3.0	0.0002	**
StepGraph	ROC-AUC	0.0	$5.36e - 07$	**
StepGraph	AP	2.0	$1.43e - 06$	**
Ledoit-Wolf	Accuracy	76.5	0.1424	ns
Ledoit-Wolf	ROC-AUC	52.0	0.0122	*
Ledoit-Wolf	AP	69.0	0.0526	ns

Table 5.5 presents the classification results. ColdCov (Optimized) achieves the highest observed mean performance on all three metrics: 62.1% Accuracy, 0.659 ROC-AUC, and 0.678 AP. The performance gap is particularly notable against StepGraph (+8.2% Accuracy, +0.110 ROC-AUC). Against the strongest traditional baseline, Graphical Lasso, the gains are modest (+2.3% Accuracy, +0.025 ROC-AUC, +0.005 AP).

To formally assess whether these differences are statistically significant, we applied the Wilcoxon signed-rank test to each pairwise comparison of ColdCov (Optimized) against every baseline, across all three metrics (Table 5.6). Following the methodology recommended by Rainio et al. (2024), we used 5 × 5 repeated stratified cross-validation (25 paired observations per comparison) to ensure adequate statistical power, and applied Holm–Bonferroni correction for the 9 simultaneous tests (3 baselines × 3 metrics) to control the family-wise error rate. The results (Table 5.6) confirm that ColdCov (Optimized) significantly outperforms StepGraph on all three metrics ($p < 0.001$ after correction) and Ledoit-Wolf on ROC-AUC ($p = 0.012$), while the differences against Graphical Lasso do not reach statistical significance at conventional thresholds. The classification results in Table 5.5 are reported from the first repeat (equivalent to the standard 5-fold CV), while the full 25-fold repeated CV is used solely as input to the statistical test.

The unoptimized ColdCov variant already achieves competitive performance (61.7% Accuracy), indicating that the precision matrix features are informative for downstream classification. The threshold optimization (Section 4.1.4) provides a modest additional gain by removing weak conditional dependencies that contribute noise rather than signal to the classification task. PCA+LDA projections of these precision matrix features, visualizing the class separability that underlies the classification performance, are presented in Appendix B.3.2.

5.4.3 Visual Analysis of Estimated Brain Connectivity

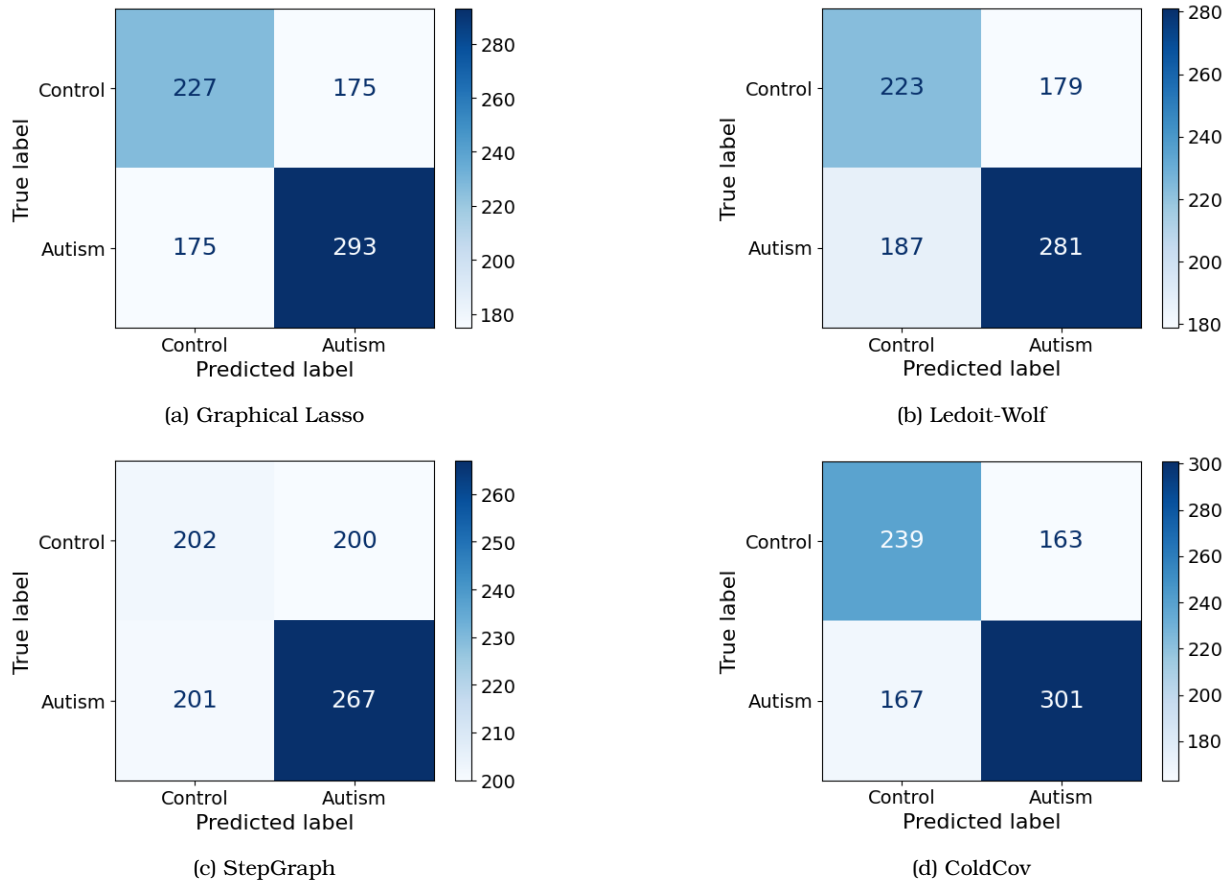


Figure 5.7: Confusion matrices for the Control vs. ASD classification task across four estimation methods. ColdCov shows the most balanced classification in this representative fold, with higher true positive rates for both diagnostic groups. StepGraph, by contrast, shows strong bias toward the majority class.

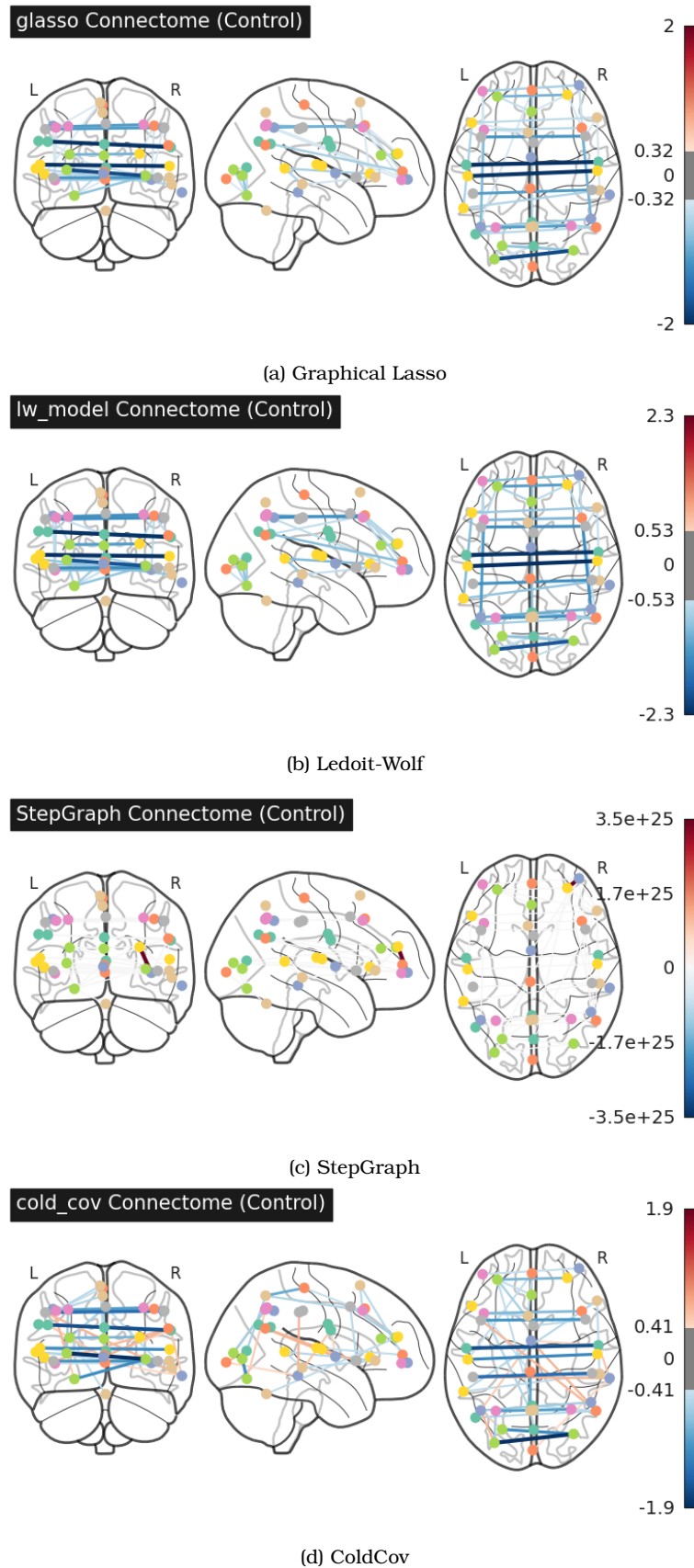


Figure 5.8: Estimated functional brain connectomes for a representative subject, visualized using the Nilearn glass brain projection. Each method reveals different topological characteristics: Graphical Lasso produces a sparse network with few strong connections, Ledoit-Wolf yields a dense network, StepGraph identifies a minimal set of edges, and ColdCov recovers a rich yet structured connectivity pattern with distributed inter-regional connections.

The confusion matrices (Figure 5.7) reveal qualitative differences in classification behavior. ColdCov produces the most balanced predictions across both diagnostic groups, whereas StepGraph exhibits a pronounced bias toward the majority class (ASD). This balance is clinically important: in neuroimaging diagnostics, both false negatives (missing a diagnosis) and false positives (incorrect diagnosis) carry significant consequences.

The connectome visualizations (Figure 5.8) illustrate the structural differences between estimators. ColdCov recovers a rich connectivity pattern with distributed inter-regional connections, capturing both long-range and local dependencies. Graphical Lasso produces a sparser network dominated by strong individual connections, while Ledoit-Wolf yields a dense, less interpretable graph. StepGraph identifies a minimal edge set, which, while sparse, may exclude biologically relevant weak connections that contribute to the classification signal.

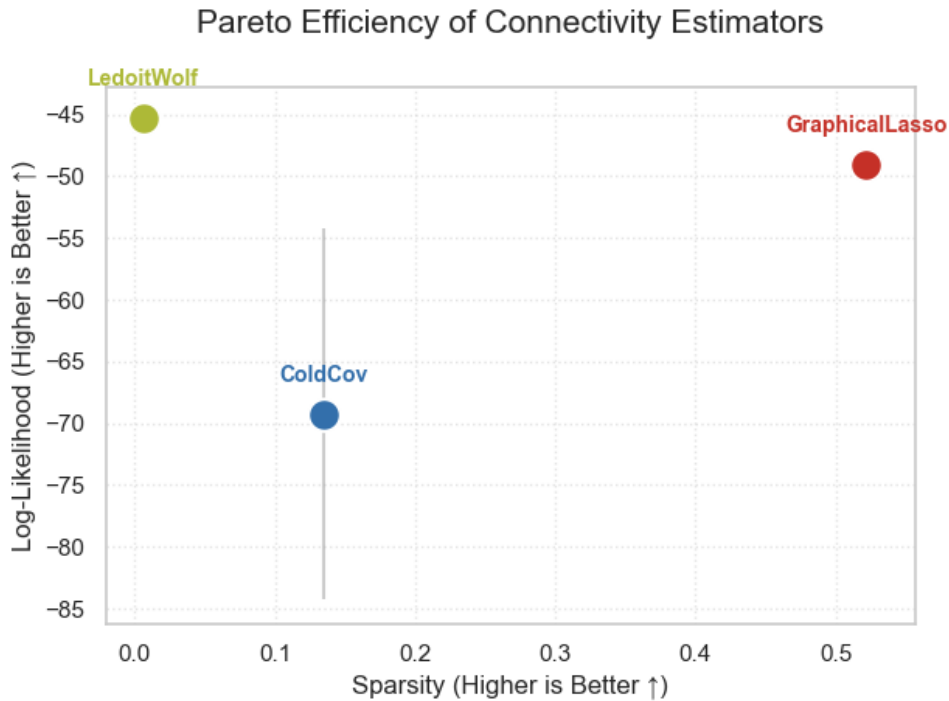


Figure 5.9: Pareto efficiency of connectivity estimators on the ABIDE dataset, plotting sparsity (fraction of near-zero precision matrix entries, higher is better) against log-likelihood (higher is better). Graphical Lasso achieves the highest sparsity (≈ 0.51) with moderate log-likelihood; Ledoit-Wolf attains the best log-likelihood (≈ -45) but near-zero sparsity; ColdCov occupies an intermediate position (≈ 0.13 sparsity, ≈ -70 log-likelihood), demonstrating that its precision matrices are not fully dense but rather exhibit moderate sparsity while retaining rich conditional dependency structure.

Additional per-method visualizations, including precision matrix heatmaps (Appendix B.3.1) and feature-space projections, are collected in Appendix B.3.

The Pareto efficiency analysis (Figure 5.9) reveals a clear three-way trade-off between sparsity and log-likelihood. Graphical Lasso achieves the highest sparsity but sacrifices log-likelihood; Ledoit-Wolf maximises log-likelihood at the cost of producing near-zero spar-

sity; ColdCov occupies an intermediate position with moderate sparsity ($\approx 13\%$), indicating that its precision matrices, while comparatively less sparse than Graphical Lasso, are not fully dense. This corrected characterisation (using a tolerance-based sparsity metric that accounts for the small but non-zero entries inherent to continuous generative models) resolves the apparent discrepancy between ColdCov’s sparse heatmap visualisation and its quantitative sparsity score. Combined with its competitive classification performance (Table 5.5), ColdCov demonstrates a favourable balance of structural sparsity and discriminative power among the evaluated estimators.

5.5 Validating the Non-Gaussian Degradation Assumption

A fundamental theoretical premise of the ColdCov framework is that the noise introduced by calculating the Sample Covariance Matrix from limited data is not simple additive Gaussian noise. This distinction is what necessitates the use of "Cold Diffusion" (which can handle arbitrary deterministic degradations) rather than standard Denoising Diffusion Probabilistic Models (DDPMs), which assume Gaussian noise transitions.

To empirically validate this, we analyzed the residuals of the degradation process, defined as $R = \hat{\Sigma}_t - \Sigma_{\text{true}}$, where $\hat{\Sigma}_t$ denotes the empirical sample covariance matrix (SCM) computed from n observations drawn from $\mathcal{N}(0, \Sigma_{\text{true}})$. The analysis used $p = 40$ dimensional covariance matrices from the test set (10,000 matrices spanning four graph topologies). The degradation was applied at a sampling ratio of $n/p = 2$ ($n = 80$ samples), placing the analysis in a regime where $n > p$. If the degradation were Gaussian (as in DDPM), these residuals should strictly follow a normal distribution.

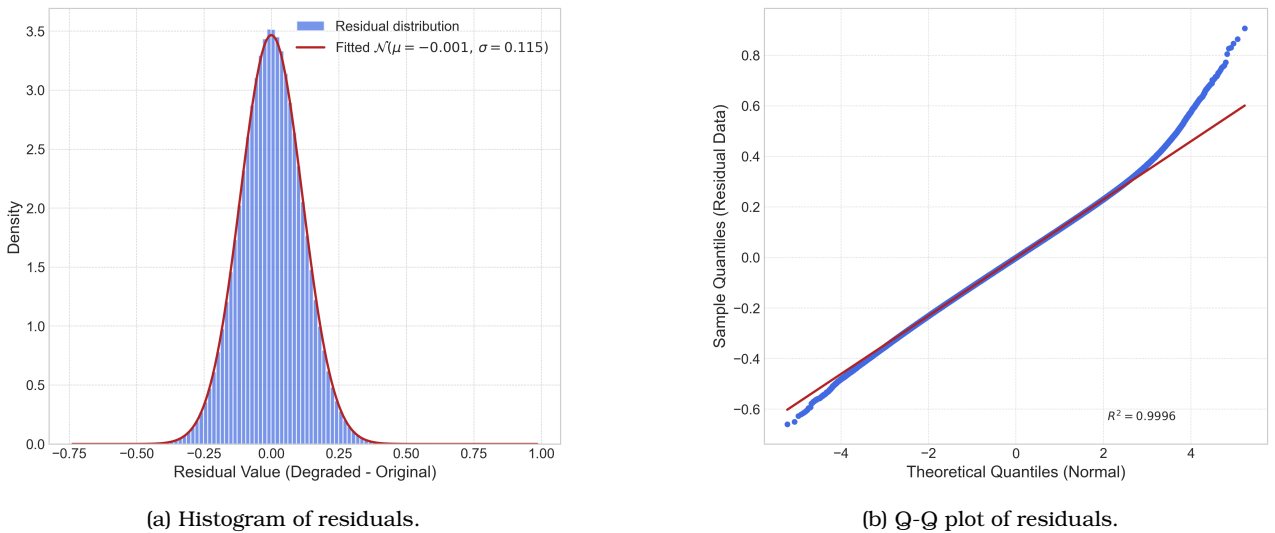


Figure 5.10: Visual analysis of the degradation residuals for $p = 40$ and $n/p = 2$. While visually Gaussian, statistical tests confirm latent non-Gaussianity.

Visually (Figure 5.10), the residuals appear misleadingly bell-shaped. However, high-powered statistical tests tell a different story. As shown in Figure 5.11 and Table 5.7, as the

sample size increases, the p-values for the Kolmogorov-Smirnov, D'Agostino-Pearson, and Shapiro-Wilk tests rapidly converge to 0.0. This indicates a statistically significant rejection of the null hypothesis (normality).

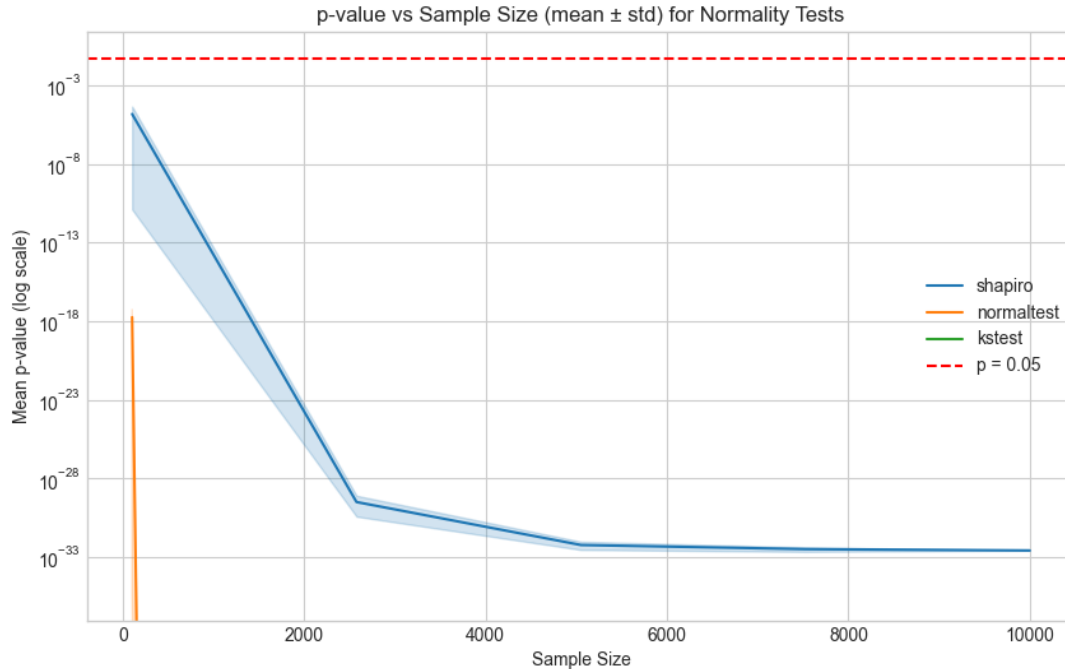


Figure 5.11: Convergence of p-values from three normality tests (Kolmogorov-Smirnov, D'Agostino-Pearson, Shapiro-Wilk) as the number of residual elements sampled increases from 100 to 10,000.

The "heavy tails" or subtle skewness detected by these tests confirm that the degradation process involves complex, non-linear interactions inherent to the Wishart distribution of sample covariance matrices. This finding justifies our architectural choice: a model trained to simply remove Gaussian noise would be fundamentally misspecified for this inverse problem. While the binary rejection of the normality null hypothesis is itself informative, future work could strengthen this analysis by reporting the sample skewness and excess kurtosis of the residual distribution to quantify the magnitude of the departure from Gaussianity.

Table 5.7: Mean and standard deviation of p-values from normality tests at representative residual sample sizes (100 and 10,000 elements).

Test method	Kolmogorov-Smirnov	D'Agostino-Pearson	Shapiro-Wilk
Sample size			
100	$0.0 \pm (0.0)$	1.73×10^{-25}	6.03×10^{-9}
10000	$0.0 \pm (0.0)$	$0.0 \pm (0.0)$	2.87×10^{-34}

5.6 Ablation Studies: Impact of Model Components

To understand which components of our proposed architecture drive the performance gains, we conducted ablation studies comparing three ColdCov variants: **ColdCov (Unconstrained)**, which lacks the SPSP architectural constraint; **ColdCov (SPSP + Covariance Loss)**, which enforces SPSP but trains with an L1 loss on the covariance matrix alone (no precision-domain penalty); and **ColdCov (SPSP + Custom Loss)**, the proposed model that combines the SPSP constraint with a composite loss penalising error in both the covariance and precision domains.

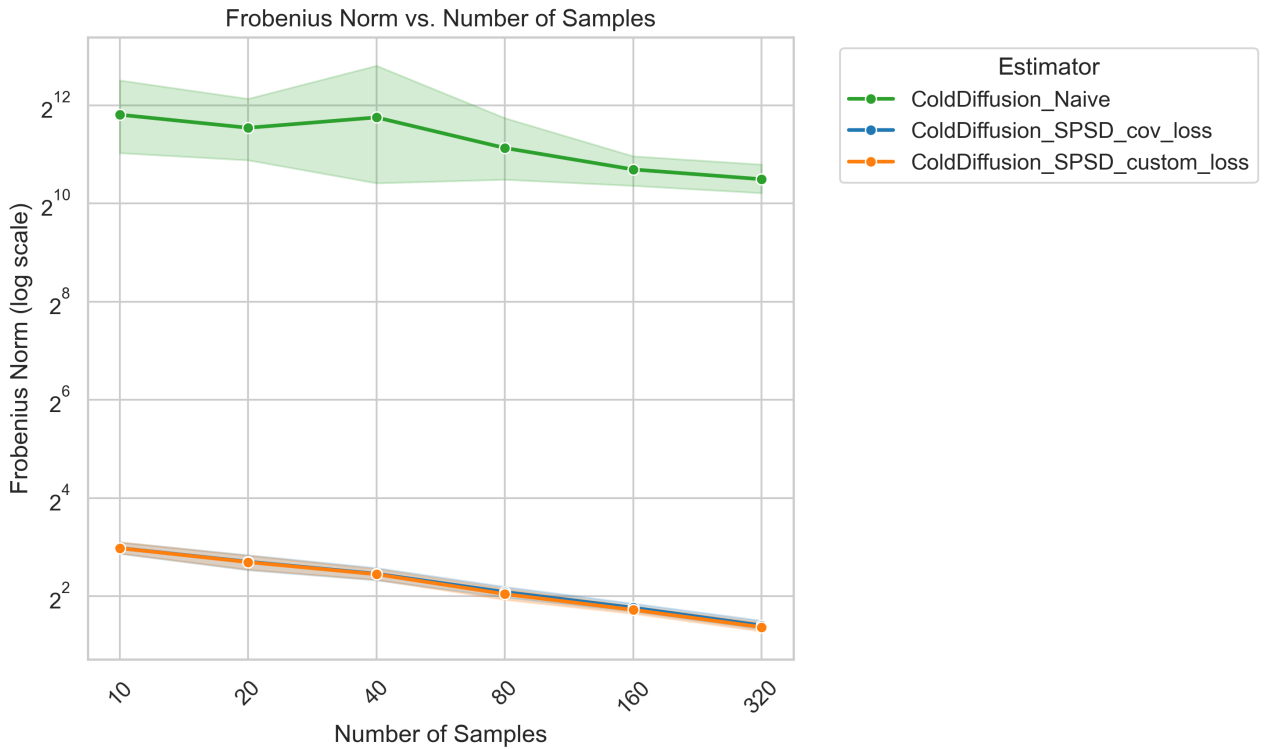


Figure 5.12: Ablation study: Frobenius Norm reconstruction error. The Unconstrained variant produces catastrophically invalid outputs ($> 1000\times$ error), confirming that the SPSP constraint is essential, not merely beneficial.

Table 5.8: Ablation Study: Key Metrics at Representative Sample Sizes

n	Metric	Unconstrained	SPSP + Cov Loss	SPSP + Custom Loss
10	Frobenius	3588.42 ± 3199.85	7.894 ± 1.077	7.884 ± 1.110
	NKL	0.982 ± 0.001	0.870 ± 0.033	0.869 ± 0.037
	ROC AUC	0.500 ± 0.000	0.615 ± 0.032	0.601 ± 0.044
320	Frobenius	1443.33 ± 539.46	2.648 ± 0.296	2.585 ± 0.283
	NKL	0.982 ± 0.001	0.394 ± 0.044	0.380 ± 0.039
	ROC AUC	0.500 ± 0.000	0.975 ± 0.018	0.973 ± 0.024

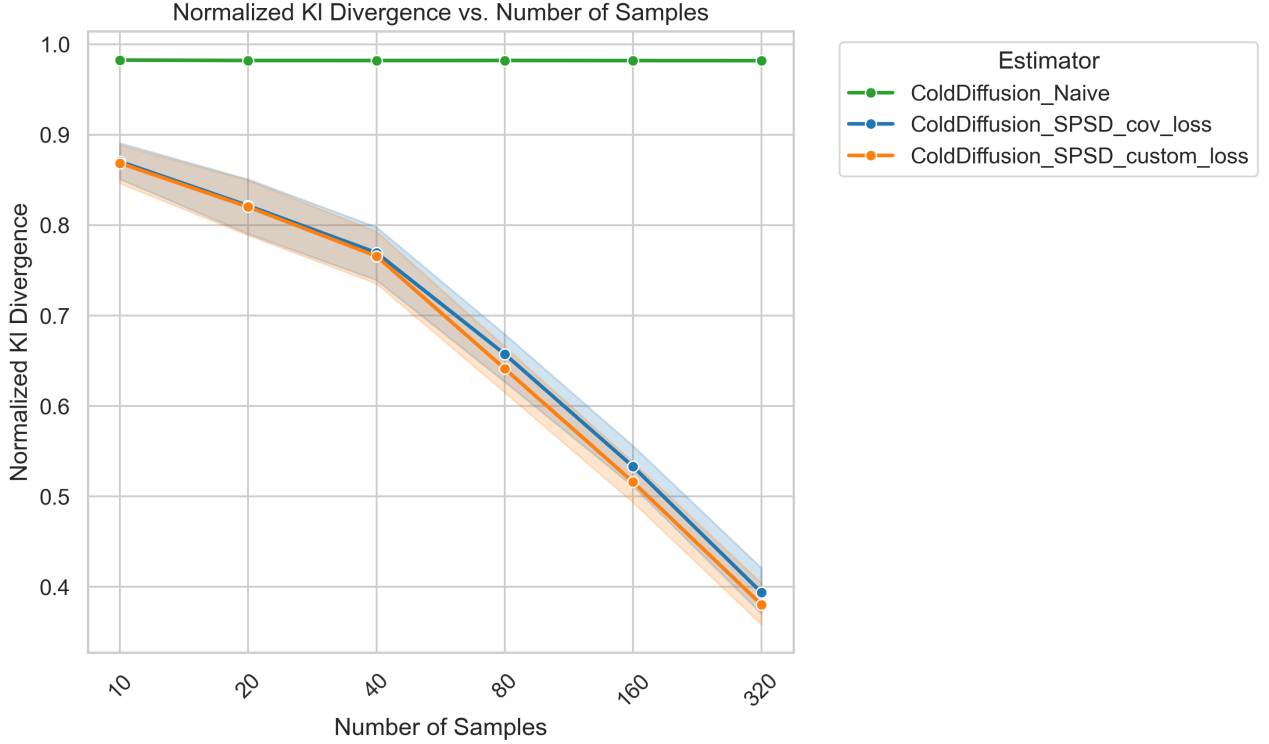


Figure 5.13: Ablation study: Normalized KL Divergence. The Unconstrained variant remains near total divergence (≈ 0.98) at all sample sizes, while both SPSP variants improve steadily as data increases, with the Custom Loss variant marginally ahead.

The results (Figures 5.12, 5.13, and 5.14 and Table 5.8) reveal two clear findings. First, the SPSP architectural constraint is not merely beneficial; it is essential. The Unconstrained variant produces catastrophically invalid outputs, with Frobenius errors exceeding 3500 at $n = 10$ (a $\sim 450\times$ degradation compared to the constrained variants) and random-chance link prediction (ROC AUC ≈ 0.50) at all sample sizes. Without the SPSP guarantee, the network’s output matrices violate positive semi-definiteness, rendering them useless for precision matrix inversion and all downstream statistical tasks.

Second, the choice of loss function has a more nuanced effect. The Custom Loss (which penalises error in both the covariance and precision domains) marginally outperforms the Covariance Loss on reconstruction metrics: Frobenius 7.884 vs 7.894 at $n = 10$, and NKL 0.380 vs 0.394 at $n = 320$. Conversely, the Covariance Loss marginally leads on ROC AUC at low sample sizes (0.615 vs 0.601 at $n = 10$). Overall, the Custom Loss provides the best balance across all metrics, which is why it is adopted as the default configuration for the proposed ColdCov model.

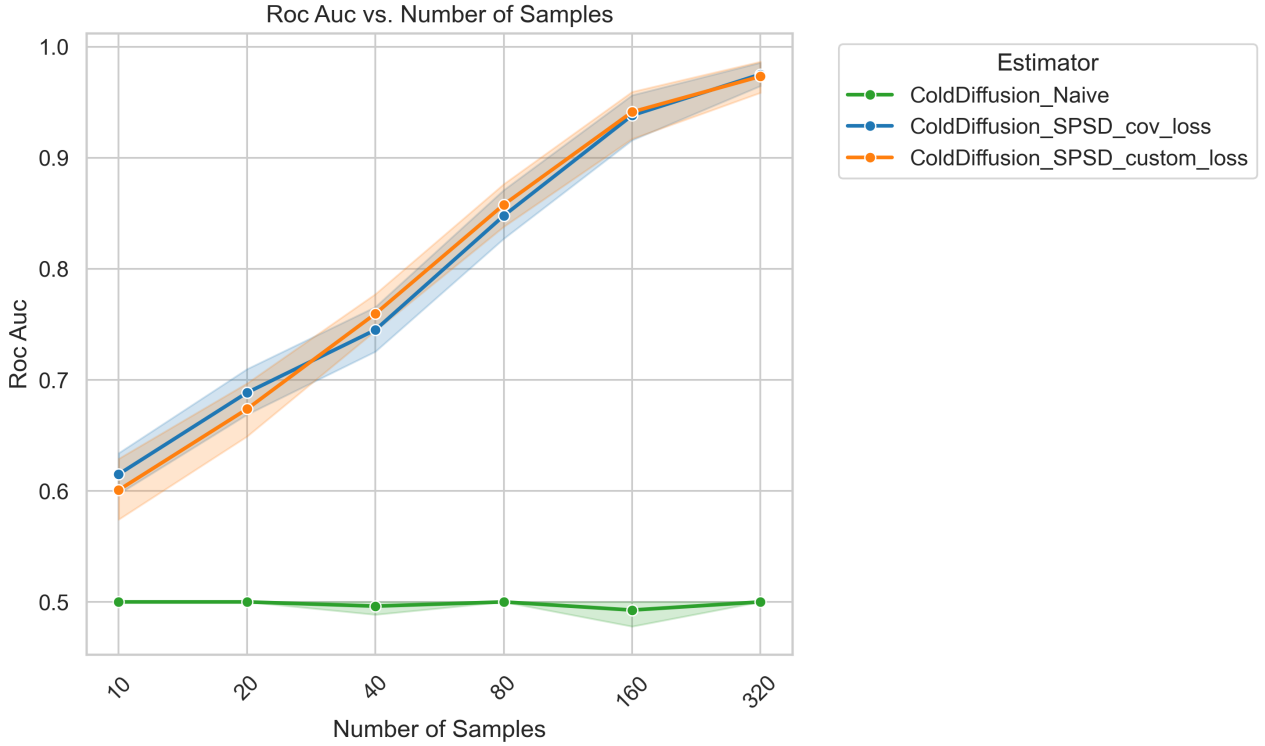


Figure 5.14: Ablation study: ROC AUC for graph structure recovery. Both SPSD variants achieve meaningful link prediction, while the Unconstrained variant performs at random-chance level (≈ 0.50).

5.7 Discussion of Results

The results presented in this chapter provide strong empirical support for the ColdCov framework and reveal a nuanced picture of its capabilities relative to established baselines.

Covariance reconstruction: ColdCov achieves the lowest Frobenius Norm error from $n = 20$ through $n = 320$, a strong result for a prior-based method. The generative prior learned during training provides a consistent regularisation benefit that does not diminish as data becomes abundant, challenging the conventional assumption that likelihood-based methods must eventually dominate. This reconstruction advantage extends to the precision matrix: ColdCov leads the Normalized KL Divergence from $n = 20$ through $n = 160$, indicating competitive probabilistic fidelity.

Graph structure recovery: With the corrected partial-correlation-based link prediction evaluation, a clear division of strengths emerges. Sparse-enforcing methods, specifically Graphical Lasso and Ledoit-Wolf, consistently lead ROC AUC and Average Precision at all sample sizes. This is expected: these methods produce precision matrices with explicit sparsity patterns that align directly with the edge-detection objective. ColdCov, which produces comparatively dense covariance estimates, remains competitive (ROC AUC 0.601 vs random baseline 0.500 at $n = 10$) and outperforms StepGraph and DeepGraph, but cannot match the graph structure recovery of methods designed for that purpose.

Complementary strengths on synthetic data: On the synthetic evaluation, ColdCov and sparse-enforcing methods address complementary aspects of the covariance estimation problem. ColdCov excels at covariance reconstruction fidelity; Graphical Lasso and Ledoit-Wolf excel at graph structure estimation. This division reflects a fundamental trade-off: ColdCov’s dense generative prior is well-suited for reconstruction, while the explicit sparsity of ℓ_1 -regularized methods aligns directly with the edge-detection objective of link prediction.

Downstream discriminative performance: The ABIDE classification results (Section 5.4) reveal an additional dimension to ColdCov’s contribution. When precision matrix features are evaluated on a downstream subject-level classification task (distinguishing Control from Autism subjects), ColdCov shows the best observed mean performance on every metric: Accuracy (62.1% vs. 59.8% for the best baseline), ROC-AUC (0.659 vs. 0.634), and Average Precision (0.678 vs. 0.673). However, these gains against the strongest baselines are modest in absolute terms and, as noted in Section 5.4, the differences against Graphical Lasso do not reach statistical significance at conventional thresholds. These results should be interpreted as evidence of competitive discriminative feature quality rather than conclusive statistical superiority.

Reconciling synthetic and real-world results: There is no contradiction between the synthetic and real-world findings, but rather a distinction in what each evaluation measures. In the synthetic setting, ROC-AUC and AP assess edge-level link prediction: the ability to correctly identify individual non-zero entries in the precision matrix. Sparse methods naturally excel here because their outputs have explicit zero patterns that align with the binary classification target. In the ABIDE evaluation, the same metric names measure subject-level classification: the discriminative power of the full set of precision matrix features. ColdCov’s comparatively dense precision matrices, while suboptimal for recovering individual edges, may encode richer inter-regional dependency patterns that provide useful signal for distinguishing diagnostic groups. This observation should be stated cautiously as requiring further study, but suggests that the choice of estimator depends critically on the downstream task: sparse methods for topology recovery, ColdCov for feature-based classification.

Ablation validation: The ablation study confirms that both architectural innovations are essential: the SPSP constraint prevents catastrophic output degradation ($\sim 450\times$ error increase without it), and the composite loss provides a consistent, if marginal, improvement over the covariance-only loss across reconstruction and precision matrix metrics.

Taken together, these findings establish ColdCov as a competitive covariance estimator with particular strengths in reconstruction fidelity and downstream discriminative feature extraction. On synthetic data, its generative denoising framework complements existing sparse estimation methods; on real-world neuroimaging data, it shows promise for feature-based classification tasks, strengthening the case for generative approaches to classical statistical problems.

Conclusion

This thesis has addressed a fundamental and traditional open challenge on statistics and data science, the accurate estimation of the covariance matrix in high-dimensional settings, through a novel generative denoising paradigm. We began by identifying the core problem not as a simple estimation task, but as one of reversing the statistical degradation process where estimated covariance structures are corrupted by the noise of data scarcity. Based on this conceptual shift, we developed and rigorously validated the ColdCov framework, a generative denoising approach that explicitly models and reverses this degradation. This concluding chapter synthesizes our contributions, evaluates our central hypothesis against the empirical evidence, acknowledges limitations, and charts promising directions for future research.

6.1 Conceptual, Methodological, and Empirical Contributions

The research presented in this thesis makes several key contributions to the field of statistical signal processing and graphical models. These contributions can be categorized as conceptual, methodological, and empirical.

Our conceptual contribution is a paradigm shift: reframing high-dimensional covariance estimation as an explicit generative denoising problem. Rather than relying on traditional regularization techniques that indirectly mitigate noise, we proposed to directly model and reverse the statistical degradation process. Our rigorous statistical analysis, culminating in the rejection of normality for sampling residuals, provided firm theoretical grounding for this paradigm and justified moving beyond standard Gaussian diffusion models toward the more general Cold Diffusion framework.

Our methodological contribution is the design and implementation of the ColdCov framework, the first application of Constrained Cold Diffusion to covariance estimation. This framework introduces three key innovations: (1) a principled degradation operator that ac-

curately models the non-Gaussian sampling process, (2) an architectural SPSP constraint that guarantees mathematical validity by construction, and (3) a training procedure that learns to reverse statistical degradation without requiring ground truth labels. These innovations address fundamental limitations of existing approaches while maintaining computational efficiency.

Our empirical contribution is the comprehensive validation of ColdCov against established baselines. On synthetic data, experiments demonstrated strong reconstruction performance with the lowest Frobenius error across most sample sizes and competitive precision-matrix quality, particularly in the challenging low-sample regime. Qualitative analysis showed that quantitative advantages translate to more plausible and structurally coherent recovery of underlying network structures. Graph-recovery experiments revealed that sparse methods remain stronger for explicit edge-level topology recovery. On real-world neuroimaging data (ABIDE), ColdCov’s precision matrix features showed best observed mean performance on downstream classification (Accuracy (62.1%), ROC-AUC (0.659), and Average Precision (0.678)), with modest gains over baselines in a subject-level Control vs. Autism classification task.

6.2 Revisiting the Central Hypothesis

The central research question posed in Chapter 1 was whether a deep generative framework, by explicitly modeling and reversing statistical degradation, could achieve performance competitive with traditional techniques. Based on the empirical evidence presented in Chapter 5, specifically the reconstruction fidelity analysis (Section 5.2), the topology-specific evaluation (Section 5.2.4), and the real-world ABIDE classification (Section 5.4), the evidence indicates that ColdCov is a promising approach with distinct strengths in covariance reconstruction and downstream discriminative feature extraction, while being complementary to sparse methods for explicit graph topology recovery. On synthetic data, ColdCov consistently achieves the lowest reconstruction error from $n = 20$ through $n = 320$, reducing the Frobenius norm by up to 28% relative to the strongest baseline. On real-world neuroimaging data, ColdCov shows competitive performance on downstream classification (Accuracy 62.1%, ROC-AUC 0.659, Average Precision 0.678), with modest gains against the strongest baselines.

ColdCov demonstrates complementary strengths relative to established baselines across multiple dimensions:

- **Covariance Reconstruction Fidelity:** Quantitative analysis on synthetic data showed consistent reduction in reconstruction error across all tested degradation levels, from $n = 20$ onward. In the challenging low-sample regime ($n = 20$), ColdCov achieves a Frobenius norm error of 6.475 compared to 9.000 for Graphical Lasso (28.1% reduction) and 7.495 for Ledoit-Wolf (13.6% reduction). This advantage persists into the over-sampled regime: at $n = 320$, ColdCov achieves 2.585 versus Graphical Lasso’s

2.835 (8.8% reduction) and Ledoit-Wolf’s 3.720 (30.5% reduction), confirming that the generative prior provides a consistent regularization benefit even when data is abundant.

- **Graph Structure Recovery:** Sparse-enforcing methods excel generally at topology recovery. ColdCov demonstrated unique robustness in modeling complex hub structures, achieving the lowest reconstruction error on Preferential Attachment graphs across all sample sizes. Results with other topologies (lattice, small-world) are detailed in Section 5.2.4 and Appendix B.
- **Downstream Discriminative Performance:** Real-world validation on the ABIDE neuroimaging dataset confirmed practical utility. ColdCov’s precision matrix features showed best observed mean performance on downstream subject-level classification (Accuracy 62.1%, ROC-AUC 0.659, and Average Precision 0.678). However, gains against the strongest baselines are modest and, as detailed in the statistical analysis of Section 5.4, do not reach conventional significance thresholds against Graphical Lasso ($p = 1.000$ after Holm–Bonferroni correction).

The success of ColdCov supports our central hypothesis and suggests that the generative denoising paradigm represents a competitive and promising approach to covariance estimation. More broadly, it demonstrates the potential of adapting generative models to reverse specific, well-defined degradation processes in statistical problems.

6.3 Limitations and Challenges

While the ColdCov framework demonstrates strong performance in the low-sample regime, it is not without limitations. Acknowledging these constraints is essential for understanding the practical boundaries of the method and identifying necessary areas for future improvement.

6.3.1 Dependency on the Training Prior

A fundamental property of any generative approach is its reliance on the distribution of data seen during training. ColdCov explicitly learns the manifold of valid SPSP matrices from a synthetic dataset. While our experiments utilized a diverse range of topologies (lattice, small-world, scale-free), the model’s performance on real-world data is inherently bounded by how well the synthetic prior “covers” the true biological or physical manifold. There is a theoretical risk of dataset-induced bias: in extremely low-signal regimes ($n = 10$), the model might impose a structural pattern (e.g., a specific type of hubiness) not because it exists in the observed data, but because the synthetic training set either over-represents certain topological features or is insufficiently diverse to cover all real-world cases. This is a systematic bias inherited from the training distribution. To mitigate the risk of the model

learning position-specific artifacts (e.g., associating hubs with fixed matrix indices), we augmented the training data with random variable permutations. This forces the network to learn permutation-invariant structural features rather than memorizing spatial locations. An alternative explanation for ColdCov’s strong performance on hub structures is that certain salient patterns, such as prominent hubs, may simply be easier for the model to detect and recover than broad, diffuse local structures, rather than this being solely a coverage issue. While our results on the ABIDE dataset suggest good generalization, deployment in radically different domains (e.g., financial time series) would likely require retraining on domain-specific synthetic priors.

6.3.2 Computational Cost of Inference

Compared to classical estimators, ColdCov incurs a higher computational cost at inference time. To make this comparison precise, we summarize the asymptotic complexity of each method (derived in Section 4.2.5). Ledoit-Wolf shrinkage admits a closed-form solution in $O(p^2n)$ operations, completing in milliseconds on a standard CPU for moderate dimensions. Cross-validated Graphical Lasso requires $O(k \cdot |\Lambda| \cdot T \cdot p^3)$ operations, where k is the number of cross-validation folds, $|\Lambda|$ is the regularization grid size, and $T = 5\text{--}50$ is the typical number of block coordinate descent iterations per fit; this cross-validation overhead can dominate the overall wall-clock time for moderate dimensions.

ColdCov inference requires a single deterministic forward pass through its convolutional U-Net, scaling as $O(p^2)$ with respect to the input dimension. However, the constant factor involves approximately 2.7 million learned parameters and GPU kernel dispatch overhead. For the experimental setting used in this thesis ($p = 40$), a single ColdCov forward pass is expected to complete in the order of milliseconds on a GPU, comparable to the per-query cost of a single Graphical Lasso fit, though Graphical Lasso additionally incurs the cross-validation overhead described above. While this cost is negligible for “offline” scientific analysis (e.g., analyzing a fixed fMRI dataset), it may be prohibitive for ultra-low-latency applications (e.g., high-frequency trading) without further optimization or quantization.

6.3.3 Dependency on Fixed Input Dimensions

Another notable limitation of the current ColdCov implementation stems from the use of a standard U-Net architecture, which mandates fixed-size input tensors matching the training data (in our primary experiments, $p = 40$). While smaller dimensions can be accommodated through simple padding (as demonstrated with the $p = 39$ ABIDE data in Section 5.4), and the fully convolutional nature of the U-Net theoretically permits inference on dimensions that are exact multiples of the training size (e.g., $p = 80, 120$), the model cannot inherently process arbitrary dimensions without performance degradation or complete retraining. This fixed-size dependency limits immediate deployment as a drop-in replacement for classical estimators that naturally accept datasets of any feature dimension p .

Importantly, this constraint is a property of the chosen neural network backbone, not of the ColdCov framework itself. The degradation operator, the SPSD architectural constraint, and the Cold Diffusion training procedure are all entirely dimension-agnostic. Recent work on flexible vision transformers, such as FiT (Lu et al., 2024), demonstrates that transformer architectures can natively process inputs of unrestricted dimensions through adaptive tokenization, offering a direct path to resolving this limitation. We discuss this architectural direction and its associated computational trade-offs in Section 6.4.3.

6.3.4 The “Black Box” Nature of Deep Learning

Unlike the Graphical Lasso, whose sparsity properties are proven theorems arising from the L1 penalty, the properties of the ColdCov estimator are empirical. We cannot prove *a priori* that the network will not produce an outlier under adversarial conditions. The SPSD constraint guarantees mathematical validity (eigenvalues ≥ 0), but it does not guarantee statistical consistency in the classical sense. This lack of theoretical error bounds, common to all deep learning methods, may hinder adoption in safety-critical systems where provable guarantees are mandated.

6.3.5 Scope of the Current Evaluation

In this thesis, the training and empirical evaluation of ColdCov were conducted exclusively on data drawn from multivariate Gaussian distributions, $X \sim \mathcal{N}(0, \Sigma)$, which implies that the degradation noise follows Wishart dynamics. It is important to emphasise that this is a constraint of the *experimental scope*, not a fundamental limitation of the ColdCov architecture. Because ColdCov is a purely data-driven deep learning method, nothing in its design (the U-Net backbone, the SPSD projection, or the Cold Diffusion training procedure) imposes a Gaussian requirement on the underlying data. Deep generative models routinely capture highly complex, non-Gaussian distributions in domains such as natural image synthesis, demonstrating that neural denoisers can learn structure far beyond what any single parametric family describes.

Notably, the degradation noise that ColdCov already handles is itself non-Gaussian: as demonstrated empirically in Section 5.5, the residuals produced by finite sampling ($\hat{\Sigma}_t - \Sigma_{\text{true}}$) deviate significantly from normality. The open question, therefore, is not whether the model can cope with non-Gaussian noise (it already does), but whether it generalises to data whose underlying distribution is also non-Gaussian. Domains with inherently discrete data (e.g., counts in RNA-seq), bounded variables (e.g., percentages), or heavy-tailed observations (e.g., financial returns) remain untested and represent a natural direction for future validation, as discussed in Section 6.4.

6.4 Future Research

The promising results and identified limitations of this work open several exciting research directions. While this thesis has established the viability of the generative denoising paradigm, significant opportunities remain to enhance the framework’s robustness, scalability, and theoretical underpinnings.

A key question arising from this work concerns how ColdCov should be positioned: as a general covariance estimator, or as an estimator that produces useful discriminative features? The evidence supports both interpretations. ColdCov is an effective general covariance estimator, achieving the lowest reconstruction error across most sample sizes. Simultaneously, its dense outputs produce features that are competitive for downstream discriminative tasks. These are complementary strengths: the reconstruction fidelity enables the discriminative utility, and together they position ColdCov as a complement to sparse methods rather than a replacement. This dual character is not a contradiction but a reflection of the task-dependent nature of covariance estimation quality.

The most important next steps for validating ColdCov’s generalization span several equally important directions. Evaluating on non-Gaussian data distributions would test whether the framework’s advantages extend beyond the Gaussian setting used in this thesis. Exploring larger dimensions, particularly through flexible architectures such as Vision Transformers capable of handling variable input sizes, would address the fixed-dimension limitation and enable deployment as a true drop-in estimator. Validating on multiple real-world datasets beyond ABIDE would establish external validity and demonstrate robustness across domains. These directions are complementary and should be pursued in parallel.

6.4.1 Advanced Iterative Inference Strategies

A core distinction of our current ColdCov implementation is its reliance on a “single-step” restoration operator. We train the model to predict Σ_{true} directly from $D(\Sigma_{true}, n)$, effectively collapsing the reverse process into a single jump. While computationally efficient, this approach does not fully leverage the power of diffusion models to iteratively refine estimates.

Future work should investigate Multi-Step Restoration. In the Cold Diffusion framework, this could be implemented by training the restoration network to reverse smaller degradation steps (e.g., from sample size $n = 10$ to $n = 20$) rather than the full degradation at once. Alternatively, a corrector step based on Langevin dynamics (Song et al., 2021) could be introduced during inference. This approach uses the score function (the gradient of the log-probability) to iteratively refine predictions by taking small stochastic gradient steps toward higher-probability regions of the learned distribution. Concretely, this would involve taking the model’s prediction, re-degrading it, and using the difference between the re-degraded estimate and the original observation to guide a gradient descent step. Such

iterative refinement could be particularly beneficial in the highest-noise regimes ($n \leq 10$), allowing the model to produce plausible structures and then critique them against the data consistency term.

A complementary training strategy worth exploring is curriculum learning. Rather than training on a uniform distribution of graph topologies, a curriculum could start with easier, hub-dominated graphs (e.g., Preferential Attachment) where the structural patterns are most salient, progressively increase the ratio of harder topologies (e.g., small-world networks with weaker local structure), and finally fine-tune on the most difficult cases: graphs with low edge weights and high noise where precision and sparsity matter most. This approach mirrors how humans learn complex skills and has proven effective in training deep neural networks on other structured prediction tasks.

6.4.2 Validation on Non-Gaussian Data Distributions

As established in Section 6.3, ColdCov is a data-driven method with no architectural dependency on Gaussianity; the Gaussian setting in this thesis reflects the experimental scope, not a theoretical constraint. Because the Cold Diffusion restoration process already handles non-Gaussian degradation noise (validated empirically in Section 5.5), the most compelling next step is to *evaluate* the framework on data whose underlying distribution is also non-Gaussian.

Many real-world domains produce covariance structures that arise from inherently non-Gaussian processes: single-cell RNA sequencing yields discrete count data (Schäfer e Strimmer, 2005; Emmert-Streib et al., 2014), financial markets exhibit heavy-tailed returns (Markowitz, 1952; Ledoit e Wolf, 2004), and social network interactions generate sparse, binary observations. Future work should train ColdCov on empirical covariance matrices derived from such domains, testing whether the network can implicitly learn the degradation patterns specific to each application without any modification to the model architecture or training procedure. By exposing the denoiser to representative samples from Poisson, heavy-tailed, or contaminated distributions, the model could learn specialised priors that classical analytical estimators cannot capture. Successful generalisation to these settings would validate the central premise of this thesis at a broader scale: that data-driven generative denoising can match or improve upon traditional estimators regardless of the distributional assumptions underlying the data. Conversely, partial success would reveal domain-specific challenges that require targeted methodological adaptations, further enriching the understanding of when and why generative denoisers excel. Furthermore, the ability to adapt a generative model to unseen real-world data through self-supervised fine-tuning, as pioneered by SSCE (Diskin e Wiesel, 2024), could address the simulation-to-reality gap that limits supervised approaches trained exclusively on synthetic priors.

6.4.3 Flexible Architectures for Arbitrary Input Dimensions

As established in Section 6.3, the fixed-input constraint arises solely from the U-Net backbone; all other ColdCov components, including the degradation operator $D(\Sigma, n)$, the SPSPD constraint, and the training procedure, are dimension-agnostic. This separation motivates a natural question: can the convolutional backbone be replaced by an architecture that natively supports variable input dimensions, thereby enabling ColdCov to handle arbitrary p without modifying its core methodology?

Recent advances in generative modeling offer a concrete path toward this goal. The Flexible Vision Transformer (FiT) (Lu et al., 2024) is a transformer architecture designed specifically for diffusion models that operates at unrestricted input resolutions and aspect ratios. Rather than assuming a fixed spatial grid, FiT employs a flexible tokenization scheme that dynamically adapts the token sequence length to the actual input dimensions. Applied to covariance estimation, a $p \times p$ matrix would be partitioned into patches of fixed size $s \times s$ (typically $s = 2$ or 4), yielding a variable-length sequence of approximately $(p/s)^2$ tokens regardless of p . A single model trained at one dimension could thus process covariance matrices of any dimension at inference time, just as FiT generates images at resolutions not seen during training, eliminating the need for retraining or artificial zero-padding.

This architectural flexibility comes at a computational cost. Replacing convolutions with global self-attention increases the asymptotic complexity from the current $O(p^2)$ of the convolutional U-Net to $O(p^4)$ for standard attention, as derived formally in Appendix A.4. However, efficient attention mechanisms, including Performer (Choromanski et al., 2021), Linformer (Wang et al., 2020), and windowed attention variants (detailed in Appendix A.4), can reduce this cost to $O(p^2)$ or $O(p^2k)$, preserving tractability for moderate dimensions. It is also worth noting that FiT’s own evaluation reveals a flexibility-versus-specialization trade-off: at matched resolutions, FiT performs slightly below the fixed-resolution DiT architecture, suggesting that empirical validation in the covariance domain would be necessary to quantify any accuracy penalty. Despite this trade-off, adopting a flexible backbone represents a natural next step for ColdCov, enabling it to serve as a true drop-in replacement for classical estimators that accept datasets of any feature dimension p .

6.4.4 Scalability to Ultra-High Dimensions ($p > 1000$)

Our experiments focused on $p = 40$, a moderate dimension relevant for functional brain atlases but small compared to gene regulatory networks which can span 20,000+ genes. Scaling deep learning models to these dimensions presents a fundamental memory challenge: the covariance matrix size grows quadratically (p^2), meaning a $1,000 \times 1,000$ matrix would require prohibitive GPU memory under any current backbone architecture.

While adopting a flexible transformer backbone (Section 6.4.3) would resolve the fixed-dimension constraint, ultra-high dimensions additionally demand strategies that reduce

the memory footprint of each forward pass. A promising approach is Hierarchical Decomposition, where a large covariance matrix is partitioned into overlapping blocks, each block is denoised independently in parallel by the restoration network, and the per-block estimates are reconciled through a global consistency constraint that enforces agreement on shared entries. This divide-and-conquer strategy would allow the framework to scale to virtually any dimension while keeping per-block memory requirements manageable (see Appendix A.4 for the complexity implications of attention-based backbones at these scales). Successful deployment at ultra-high dimensions would unlock applications in whole-genome co-expression analysis and large-scale climate modeling, domains where p routinely exceeds 10,000.

6.4.5 Theoretical Guarantees and Stability Analysis

While our results are empirically robust, deep generative models often lack the rigorous theoretical guarantees of convex optimization methods like Graphical Lasso. Establishing such guarantees for ColdCov is an important open problem.

Future theoretical work could focus on analyzing the Restoration Operator as a Contraction Mapping. If we can prove that the specific U-Net architecture, trained under the L1 loss, acts as a contraction in the metric space of SPSD matrices, we could derive formal bounds on the reconstruction error. Additionally, analyzing the Sample Complexity of the learning process, how many synthetic training examples are required to learn a manifold of a certain complexity, would provide valuable guidance for practitioners generating training datasets. Developing this statistical learning theory for generative regularizers would bridge the gap between classic statistics and modern deep learning.

6.5 Concluding Remarks

The estimation of covariance matrices from limited samples represents a persistent challenge in modern data science, fundamentally constraining our ability to understand complex system structures. This thesis has introduced a novel and powerful paradigm to address this challenge: reframing covariance estimation as generative denoising. By conceptualizing the creation of empirical covariance matrices as a deterministic, non-Gaussian statistical degradation process, we developed the ColdCov framework, a principled generative model specifically designed to reverse this corruption.

Through comprehensive experimentation, we have demonstrated that this approach represents a competitive alternative to established statistical methods. ColdCov, with its architecturally enforced SPSD constraint, achieves strong reconstruction performance with the lowest Frobenius error across most sample sizes and competitive precision-matrix quality. The framework's dense outputs, while suboptimal for explicit graph topology recovery compared to sparse-enforcing methods, produce features that are competitive for down-

stream discriminative tasks. This work validates our central hypothesis that modeling covariance estimation as a generative denoising problem is a promising direction, and more broadly, highlights the potential of adapting generative models to reverse specific, well-defined degradation processes in statistical problems.

The implications extend beyond covariance estimation. Our framework demonstrates how domain knowledge about degradation processes can be incorporated into generative models to solve fundamental statistical problems. This paradigm could be adapted to other estimation challenges where data corruption follows known physical or statistical processes. It is our hope that this generative denoising approach will inspire new research directions and pave the way for more robust and accurate structural discovery across diverse scientific and engineering domains.

Mathematical Foundations

This appendix provides the rigorous mathematical backing for the architectural choices and experimental interpretations presented in the main text. We detail the geometry of the Symmetric Positive Semi-Definite (SPSD) cone, the statistical theory of eigenvalue spreading, the derivation of the information-theoretic metrics used for evaluation, and the computational complexity implications of attention mechanisms for covariance estimation.

A.1 Geometry of the SPSP Cone

Our **ColdCov** framework employs a specialized "SPSP Head" that outputs a matrix $\Sigma = LL^\top$, where L is a lower-triangular matrix. Here we formally prove why this construction guarantees validity and discuss the convex properties of the SPSP set.

A.1.1 Cholesky Construction Validity

Proposition: For any real lower-triangular matrix $L \in \mathbb{R}^{p \times p}$, the product $\Sigma = LL^\top$ is Symmetric Positive Semi-Definite.

Proof:

1. **Symmetry:** $(\Sigma)^\top = (LL^\top)^\top = (L^\top)^\top L^\top = LL^\top = \Sigma$. Thus, Σ is symmetric.
2. **Positive Semi-Definiteness:** By definition, a matrix A is PSD if $x^\top Ax \geq 0$ for all non-zero vectors $x \in \mathbb{R}^p$. Substituting $\Sigma = LL^\top$:

$$x^\top \Sigma x = x^\top (LL^\top) x = (x^\top L)(L^\top x) = (L^\top x)^\top (L^\top x)$$

Let $y = L^\top x$. Then the expression becomes the squared Euclidean norm of y :

$$x^\top \Sigma x = \|y\|_2^2 \geq 0$$

Since the squared norm is always non-negative, Σ is positive semi-definite.

This property ensures that our neural network output is valid by construction, regardless of the values in L .

A.1.2 Convexity of the Cone

The set of SPSD matrices, denoted $\mathcal{S}_+^p$, forms a convex cone. This is critical for our diffusion approach because it implies that any convex combination (average) of valid covariance matrices is also a valid covariance matrix. **Proof:** Let $A, B \in \mathcal{S}_+^p$ and $\alpha, \beta \geq 0$. For any x :

$$x^\top(\alpha A + \beta B)x = \alpha(x^\top Ax) + \beta(x^\top Bx)$$

Since $x^\top Ax \geq 0$ and $x^\top Bx \geq 0$, the linear combination is also non-negative. This justifies the stability of averaging diffusion steps or ensemble predictions.

A.2 Statistical Properties of the Degradation Process

In Chapter 1, we argued that the Sample Covariance Matrix (SCM) suffers from statistical degradation. This section defines the specific distributions governing this process.

A.2.1 The Wishart Distribution

If data samples $x_1, \dots, x_n$ are drawn i.i.d. from a multivariate Gaussian $\mathcal{N}_p(0, \Sigma)$, then the scatter matrix $S = \sum_{i=1}^n x_i x_i^\top$ follows a Wishart distribution, denoted $S \sim \mathcal{W}_p(\Sigma, n)$. The probability density function (for $n \geq p$) is given by:

$$f(S) = \frac{|S|^{\frac{n-p-1}{2}} e^{-\frac{1}{2}\text{tr}(\Sigma^{-1}S)}}{2^{\frac{np}{2}} |\Sigma|^{\frac{n}{2}} \Gamma_p(\frac{n}{2})} \quad (\text{A.1})$$

where Γ_p is the multivariate Gamma function. The SCM is simply $\hat{\Sigma} = \frac{1}{n-1}S$. This explicitly shows that the "noise" is not additive Gaussian (as in standard diffusion) but follows a complex, skewed distribution defined by this probability density function (PDF).

A.2.2 Marchenko-Pastur Law (Eigenvalue Spreading)

The **Marchenko-Pastur Law** describes the asymptotic behavior of the singular values of large random matrices. For a covariance matrix computed from $X \in \mathbb{R}^{p \times n}$ where entries are i.i.d with zero mean and unit variance, as $p, n \rightarrow \infty$ with ratio $p/n \rightarrow \gamma \in (0, \infty)$, the

empirical spectral density of eigenvalues λ converges to:

$$\mu_\gamma(x) = \begin{cases} \frac{1}{2\pi\gamma x} \sqrt{(b-x)(x-a)} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.2})$$

where $a = (1 - \sqrt{\gamma})^2$ and $b = (1 + \sqrt{\gamma})^2$. Crucially, if γ is large (i.e., $p \approx n$ or $p > n$), the support interval $[a, b]$ becomes very wide. This means the eigenvalues of the SCM "spread out" significantly away from the true population eigenvalue (which would be 1). This mathematical law explains why the condition number of the SCM explodes in high dimensions.

A.3 Derivation of Evaluation Metrics

A.3.1 KL Divergence for Multivariate Gaussians

The Kullback-Leibler divergence between two probability distributions P and Q is defined as $D_{KL}(P||Q) = \mathbb{E}_x[\log \frac{P(x)}{Q(x)}]$. For two zero-mean multivariate Gaussians $\mathcal{N}_0 = \mathcal{N}(0, \Sigma_0)$ and $\mathcal{N}_1 = \mathcal{N}(0, \Sigma_1)$:

$$D_{KL}(\mathcal{N}_0||\mathcal{N}_1) = \frac{1}{2} \left[\text{tr}(\Sigma_1^{-1}\Sigma_0) - p + \ln \frac{\det \Sigma_1}{\det \Sigma_0} \right]$$

In our context (Section 2.8.1), we measure the error of the estimated precision $\hat{\Theta}$ relative to the true Θ . Letting $\Sigma_0 = \Theta^{-1}$ and $\Sigma_1 = \hat{\Theta}^{-1}$, and noting $\text{tr}(AB) = \text{tr}(BA)$, we recover the form used in Eq. 2.14:

$$D_{KL} = \frac{1}{2} \left(\text{tr}(\hat{\Theta}\Theta^{-1}) - \log \det(\hat{\Theta}\Theta^{-1}) - p \right)$$

A.3.2 Gaussian Log-Likelihood

The likelihood of observing a single data point x under a Gaussian model $\mathcal{N}(0, \Sigma)$ is:

$$L(x|\Sigma) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \exp \left(-\frac{1}{2} x^\top \Sigma^{-1} x \right)$$

Taking the log sum over n independent test samples:

$$\ln \mathcal{L} = -\frac{np}{2} \ln(2\pi) - \frac{n}{2} \ln |\Sigma| - \frac{1}{2} \sum_{i=1}^n x_i^\top \Sigma^{-1} x_i$$

Using the trace trick $\sum x^\top A x = \text{tr}(A \sum x x^\top)$:

$$\ln \mathcal{L} \propto -\frac{n}{2} \ln |\Sigma| - \frac{1}{2} \text{tr}(\Sigma^{-1} S_{test})$$

This matches Eq. 2.16, confirming that maximizing likelihood is equivalent to minimizing the Stein loss.

A.4 Computational Complexity of Attention Mechanisms

The ColdCov architecture presented in this thesis uses a purely convolutional U-Net whose inference cost scales as $O(p^2)$ (Section 4.2.5). However, modern generative architectures increasingly augment or replace convolutions with global self-attention layers (Vaswani et al., 2017). In particular, flexible transformer designs such as FiT (Lu et al., 2024) can process inputs of arbitrary resolution, making them natural candidates for extending ColdCov to variable matrix dimensions (see Section 6.4.3). This section derives the asymptotic cost of applying such attention-based layers to covariance matrices and discusses strategies for managing the resulting complexity.

A.4.1 Quadratic Sequence Length and the $O(p^4)$ Bound

A covariance matrix of dimension p is represented as a $p \times p$ grid. When processed by a vision-transformer or DiT backbone, this grid is flattened into a sequence of length $n = p^2$ tokens (one per matrix element).

As established by Vaswani et al. (2017), the computational complexity of a standard global self-attention layer is $O(n^2 \cdot d)$, where n is the sequence length and d is the representation dimension (the hidden size of the tokens). Substituting our token count $n = p^2$ into this formula yields an asymptotic complexity of $O((p^2)^2 \cdot d) = O(p^4 \cdot d)$. Since the representation dimension d is a fixed architectural constant, the dominant scaling factor is $O(p^4)$.

For architectures such as Diffusion Transformers (DiT) applied to covariance estimation, this cost is the primary scalability bottleneck. At $p = 1000$, for example, the attention matrix would require computing similarities across 10^6 tokens, resulting in 10^{12} entries and far exceeding current GPU memory capacities.

A.4.2 Linear Attention and Mitigation Strategies

Several families of efficient attention mechanisms reduce the quadratic token-interaction cost:

- **Performer** (Choromanski et al., 2021) approximates the softmax kernel via random feature maps, reducing per-layer cost from $O(N^2)$ to $O(N)$ in the token count, i.e., from $O(p^4)$ to $O(p^2)$.
- **Linformer** (Wang et al., 2020) projects the key and value matrices to a fixed lower dimension $k \ll N$, achieving $O(Nk) = O(p^2k)$ cost.

- **Windowed or local attention** restricts each token to attend only within a fixed spatial neighbourhood, recovering $O(p^2)$ cost at the expense of global receptive field.

These techniques offer a path toward incorporating global contextual reasoning (beneficial for capturing long-range dependencies in covariance structures) while preserving the $O(p^2)$ scaling of the current convolutional architecture. Section 6.4.3 discusses how a FiT-based backbone could leverage these mechanisms to enable ColdCov inference at arbitrary matrix dimensions.

Supplementary Experimental Results

This appendix provides detailed tabular results extending the analysis presented in Chapter 5. We report the performance of the best estimators for each metric across different underlying graph topologies (Lattice, Preferential Attachment, Random, Small-World) and sample sizes ranging from the extreme low-sample regime ($n = 10$) to the high-sample regime ($n = 320$).

B.1 Reconstruction Performance by Topology

Table B.1 details the Frobenius Norm reconstruction error. Lower values indicate better performance. Note the strong performance of **ColdCov** in low-sample regimes across all topologies, and its robustness on Preferential Attachment graphs.

Table B.1: Best Estimators for Frobenius Norm Reconstruction Error across Topologies and Sample Sizes.

Topology	Sample Size (n)	Best Estimator	Frobenius Norm
Overall	10	ColdCov	7.233
	20	ColdCov	6.379
	40	ColdCov	5.375
	80	ColdCov	4.380
	160	ColdCov	3.638
	320	GraphicalLassoCV	2.846
Lattice	10	ColdCov	7.517
	20	ColdCov	6.577
	40	GraphicalLassoCV	5.264
	80	ColdCov	4.381
	160	GraphicalLassoCV	3.572
	320	StepGraph	1.990
Preferential	10	ColdCov	7.369
	20	ColdCov	6.535
	40	ColdCov	5.417
	80	ColdCov	4.404
	160	ColdCov	3.565
	320	ColdCov	2.993
Random	10	ColdCov	6.906
	20	ColdCov	6.124
	40	ColdCov	5.254
	80	GraphicalLassoCV	4.311
	160	GraphicalLassoCV	3.548
	320	GraphicalLassoCV	2.712
Small-World	10	ColdCov	7.142
	20	ColdCov	6.281
	40	GraphicalLassoCV	5.197
	80	GraphicalLassoCV	4.292
	160	GraphicalLassoCV	3.446
	320	StepGraph	2.324

B.2 Link Prediction Performance

Table B.2 presents the best performing models for Recall and Precision on synthetic data. A key finding is the trade-off between Recall (where **ColdCov** excels) and Precision (where **StepGraph** or **Combined** approaches often lead).

Table B.2: Best Estimators for Link Prediction Metrics (Recall vs Precision) across Sample Sizes (n).

Topology	Sample Size (n)	Recall (Structure Discovery)		Precision (Reliability)	
		Best Model	Value	Best Model	Value
Overall	10	ColdCov	0.803	Combined	0.183
	20	EmpiricalCov	0.818	Combined	0.234
	40	DeepGraph	0.840	Combined	0.304
	80	ColdCov	0.879	StepGraph	0.423
	160	ColdCov	0.915	StepGraph	0.562
	320	ColdCov	0.941	StepGraph	0.666
Lattice	10	ColdCov	0.807	Combined	0.198
	20	ColdCov	0.822	Combined	0.256
	40	ColdCov	0.849	Combined	0.328
	80	ColdCov	0.897	StepGraph	0.497
	160	ColdCov	0.937	StepGraph	0.653
	320	ColdCov	0.968	StepGraph	0.777
Preferential	10	ColdCov	0.803	Combined	0.157
	20	EmpiricalCov	0.811	Combined	0.209
	40	DeepGraph	0.827	Combined	0.263
	80	ColdCov	0.853	Combined	0.344
	160	ColdCov	0.879	StepGraph	0.407
	320	ColdCov	0.897	StepGraph	0.479

Note: "Combined" refers to a pipeline that applies Graphical Lasso to the ColdCov-denoised covariance matrix to induce sparsity in the precision estimate. This approach was explored during development but is not used in the final real-world evaluation (see Section 5.4).

B.3 Supplementary ABIDE Visualizations

This section presents additional per-method visualizations from the ABIDE classification evaluation (Section 5.4). These figures complement the comparative analysis in the main text by providing detailed views of each estimator's precision matrix structure and feature space geometry.

B.3.1 Precision Matrix Heatmaps

Figure B.1 displays the estimated precision matrix for a representative subject under each estimation method. The heatmap intensity encodes the magnitude of conditional dependencies between brain regions. Differences in sparsity patterns and edge weight distributions are visible across methods.

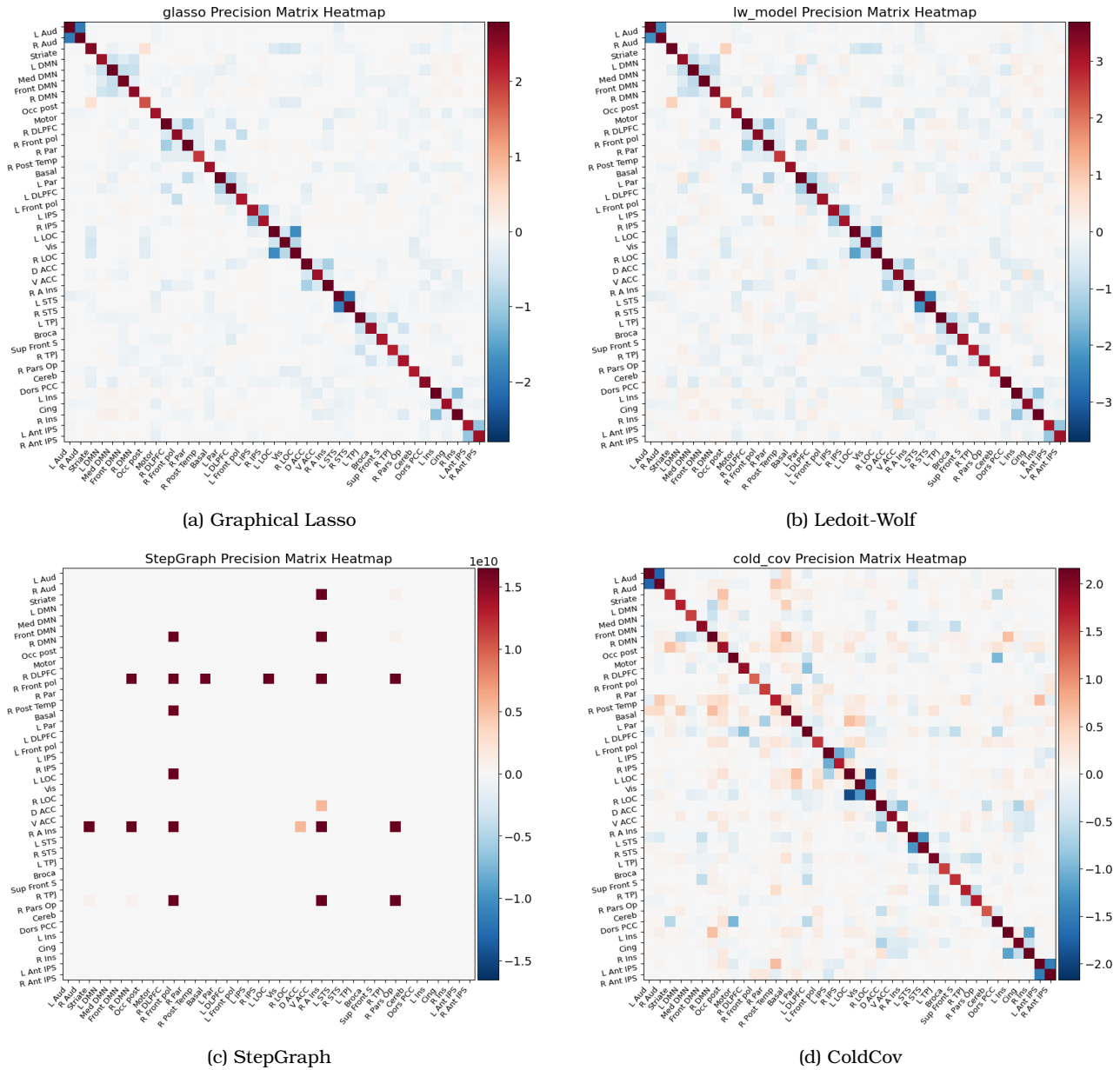


Figure B.1: Precision matrix heatmaps for a representative ABIDE subject across four estimation methods. Graphical Lasso produces a sparse matrix with concentrated strong entries, Ledoit-Wolf yields a dense matrix with attenuated off-diagonal values, StepGraph identifies a minimal set of non-zero entries, and ColdCov produces a structured matrix with distributed conditional dependencies.

B.3.2 PCA+LDA 2D Projections

Figure B.2 presents two-dimensional projections of the precision matrix feature vectors using PCA followed by Linear Discriminant Analysis (LDA). Each point represents a subject, colored by diagnostic group (Control vs. ASD). The covariance ellipses illustrate the within-class feature distributions. Greater separation between the class clusters indicates more discriminative features.

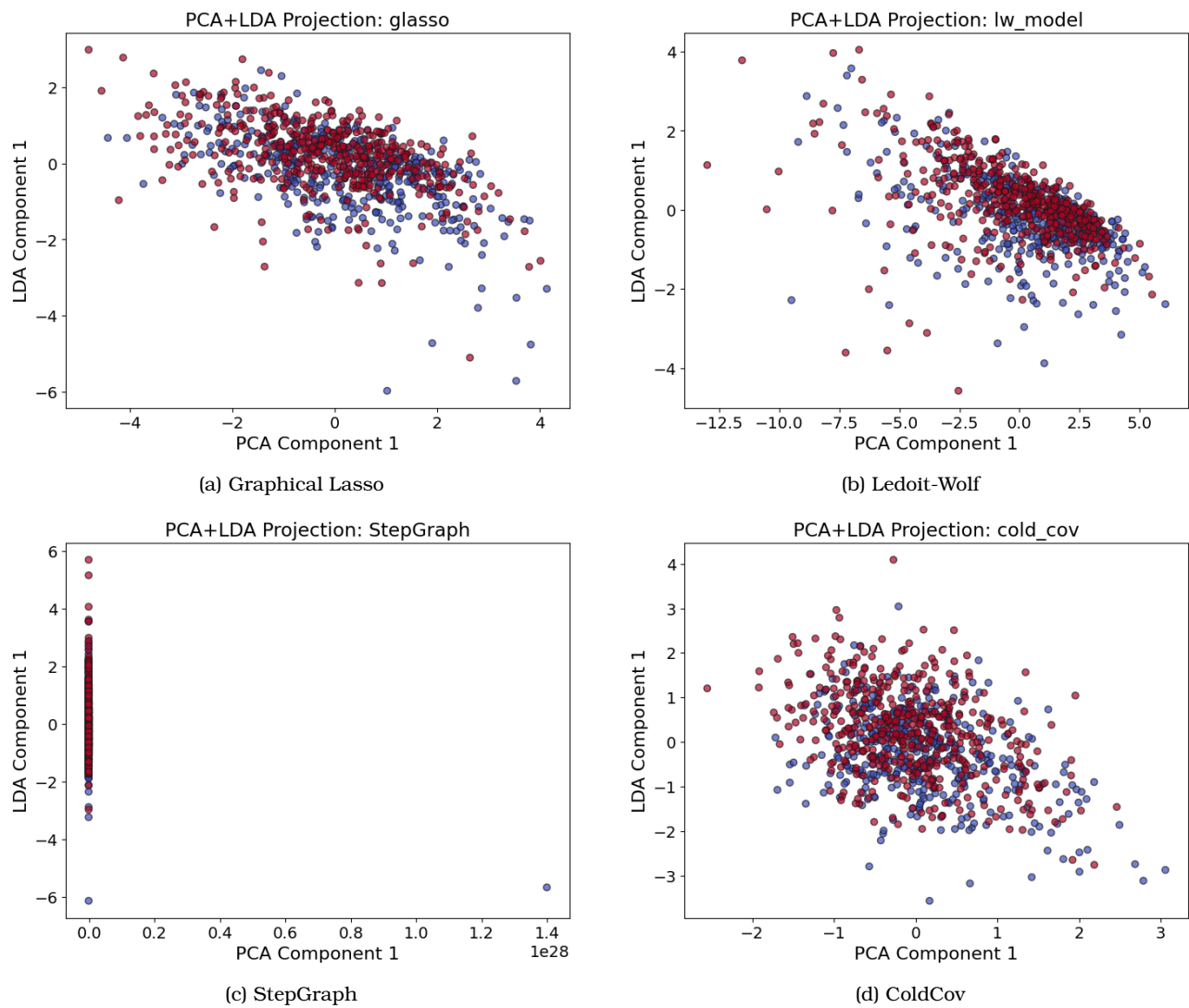


Figure B.2: PCA+LDA 2D projections of precision matrix features for the ABIDE dataset. Each point represents a subject (blue: Control, orange: ASD), with covariance ellipses showing within-class distributions. ColdCov features exhibit the greatest inter-class separation, consistent with its best observed mean classification performance.

Core Algorithms and Pseudo-Code

This appendix details the algorithmic implementation of the ColdCov framework. We focus on the two critical components that define our method: the **Degradation Operator** (D), which simulates the statistical corruption process, and the **Restoration Operator** (R), which learns to reverse it.

C.1 Statistical Degradation (Wishart Sampling)

Algorithm 1 formally defines the degradation process $D(\Sigma_{true}, t)$. Unlike standard diffusion which adds Gaussian noise, this operator simulates the generation of an empirical Sample Covariance Matrix (SCM) from a finite number of observations n_t . This effectively maps the clean manifold to the "corrupted" manifold of singular/noisy matrices.

Algorithm 1: Degradation Operator $D(\Sigma, n)$

Require: True Covariance Matrix $\Sigma \in \mathbb{R}^{p \times p}$

Require: Sample Size n (Degradation severity)

- 1: **Input:** Σ (Symmetric Positive Definite)
 - 2: $L \leftarrow \text{Cholesky}(\Sigma)$ {Decompose $\Sigma = LL^\top$ }
 - 3: Initialize $Z \in \mathbb{R}^{p \times n}$ with $Z_{ij} \sim \mathcal{N}(0, 1)$ {Standard Normal Noise}
 - 4: $X \leftarrow LZ$ {Synthetic Data Matrix, $X_{col} \sim \mathcal{N}(0, \Sigma)$ }
 - 5: $\hat{\Sigma} \leftarrow \frac{1}{n-1} XX^\top$ {Compute SCM}
 - 6: **return** $\hat{\Sigma}$ {Returns corrupted matrix $\in \mathcal{W}_p(\Sigma, n)$ }
-

C.2 Cold Diffusion Training Loop

Algorithm 2 describes the training procedure. The training procedure follows a "Degrade-Restore" cycle: in each step, we sample a severity level t , degrade the ground truth, and

force the network to recover the clean matrix. The loss is computed on both the covariance and precision manifolds.

Algoritmo 2: ColdCov Training Procedure

Require: Dataset of true precision matrices $\mathcal{D} = \{\Theta_i\}$

Require: Sample schedule $n(t) = \lfloor N_{\max} \cdot \gamma^t \rfloor$ for $t \in [1, T]$, where $\gamma = (N_{\min}/N_{\max})^{1/(T-1)}$ (see Section 4.2.1)

Require: Restoration Network f_θ (U-Net with SPSP Head)

```

1: while not converged do
2:   Sample batch of precisions  $\Theta_{batch} \sim \mathcal{D}$ 
3:    $\Sigma_{true} \leftarrow \Theta_{batch}^{-1}$  {Compute Ground Truth Covariances}
4:   Sample time steps  $t \sim \text{Uniform}(1, |T|)$ 
5:    $n \leftarrow n(t)$  {Sample size from schedule (Eq. 4.1)}
6:    $\hat{\Sigma}_{degraded} \leftarrow D(\Sigma_{true}, n)$  {Apply Degradation (Alg 1)}
7:    $\Sigma_{pred} \leftarrow f_\theta(\hat{\Sigma}_{degraded}, t)$  {Predict Clean Matrix}
8:    $\Theta_{pred} \leftarrow \Sigma_{pred}^\dagger$  {Pseudo-Inverse for Stability}
9:    $\mathcal{L}_{cov} \leftarrow \|\Sigma_{pred} - \Sigma_{true}\|_1$ 
10:   $\mathcal{L}_{prec} \leftarrow \|\Theta_{pred} - \Theta_{batch}\|_1$ 
11:   $\mathcal{L}_{total} \leftarrow \mathcal{L}_{cov} + \lambda \cdot \mathcal{L}_{prec}$ 
12:  Update  $\theta$  using  $\nabla_\theta \mathcal{L}_{total}$ 
13: end while
  
```

Bibliography

- Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., Thirion, B., e Varoquaux, G. (2014). Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8:14. [36](#)
- Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*. Wiley, Hoboken, NJ, 3^a edição. [2](#)
- Bansal, A., Borgnia, E., Chu, H.-M., Li, J., Kazemi, H., Huang, F., Goldblum, M., Geiping, J., e Goldstein, T. (2023). Cold diffusion: Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 36:41259–41282. [xii](#), [18](#), [19](#)
- Bear, M. F., Connors, B. W., e Paradiso, M. A. (2017). *Neurociências: desvendando o sistema nervoso*. Artmed editora. [25](#)
- Belilovsky, E., Kastner, K., Varoquaux, G., e Blaschko, M. B. (2017). Learning to discover sparse graphical models. Em *International conference on machine learning*, páginas 440–448. PMLR. [xii](#), [4](#), [15](#), [23](#), [31](#), [32](#), [36](#), [48](#), [54](#)
- Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press. [2](#)
- Bullmore, E. e Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198. [12](#), [26](#)
- Choromanski, K., Likhoshesterov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., Belanger, D., Colwell, L., e Weller, A. (2021). Rethinking attention with performers. Em *International Conference on Learning Representations*. [80](#), [86](#)
- Colonna, J. G. (2023). Faststepgraph: A fast stepwise algorithm for estimated the precision matrix in gaussian graphical models. Software, version v1.0.0. [36](#), [48](#), [53](#)
- D'Agostino, R. B. (1971). An omnibus test of normality for moderate and large sample size. *Biometrika*, 58(2):341–348. [20](#)

- D'Agostino, R. B. e Pearson, E. S. (1973). Tests for departure from normality. empirical results for the distributions of $\sqrt{rtb_1}$ and b_2 . *Biometrika*, 60(3):613–622. [20](#)
- Di Martino, A., Yan, C.-G., Li, Q., Denio, E., Castellanos, F. X., Alaerts, K., Anderson, J. S., Assaf, M., Bookheimer, S. Y., Dapretto, M., et al. (2014). The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659–667. [12](#), [26](#), [27](#)
- Diskin, T. e Wiesel, A. (2024). Self-supervised learning for covariance estimation. *arXiv preprint arXiv:2403.08662*. [1](#), [15](#), [32](#), [79](#)
- Donoho, D. L. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. Em *AMS Conference on Math Challenges of the 21st Century*, volume 1, p. 1–37. [2](#)
- Emmert-Streib, F., Dehmer, M., e Haibe-Kains, B. (2014). Gene regulatory networks. *Frontiers in cell and developmental biology*, 2:64. [12](#), [27](#), [79](#)
- Friedman, J., Hastie, T., e Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441. [4](#), [28](#), [30](#), [47](#)
- Hall, P., Marron, J. S., e Neeman, A. (2005). Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):427–444. [3](#)
- Hastie, T., Tibshirani, R., e Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, NY, 2^a edi. [2](#), [9](#)
- Ho, J., Jain, A., e Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851. [xii](#), [16](#), [17](#), [18](#)
- Huang, S., Li, J., Sun, L., Ye, J., Fleisher, A., Wu, T., Chen, K., e Reiman, E. (2010). Learning brain connectivity of Alzheimer's disease by sparse inverse covariance estimation. *NeuroImage*, 50(3):935–949. [26](#), [27](#)
- Laska, J. e Narayan, M. (2017). skggm 0.2.7: A scikit-learn compatible package for Gaussian and related Graphical Models. [xi](#), [xii](#), [7](#), [8](#), [9](#), [12](#), [29](#), [30](#), [36](#)
- Lauritzen, S. L. (1996). *Graphical Models*. Oxford University Press. [8](#), [9](#)
- LeCun, Y., Cortes, C., e Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2. [53](#)
- Ledoit, O. e Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411. [14](#), [27](#), [28](#), [53](#), [79](#)
- Leus, G., Marques, A. G., Moura, J. M., Ortega, A., e Shuman, D. I. (2023). Graph signal processing: History, development, impact, and outlook. *IEEE Signal Processing Magazine*, 40(4):49–60. [10](#), [11](#)

- Lu, Z., Wang, Z., Huang, D., Wu, C., Liu, X., Ouyang, W., e Bai, L. (2024). Fit: Flexible vision transformer for diffusion model. *arXiv preprint arXiv:2402.12376*. [77](#), [80](#), [86](#)
- Maathuis, M., Drton, M., Lauritzen, S., e Wainwright, M. (2018). *Handbook of graphical models*. CRC Press. [8](#), [12](#)
- Manolakis, D., Truslow, E., Pieper, M., Cooley, T., e Brueggeman, M. (2013). Detection algorithms in hyperspectral imaging systems: An overview of practical algorithms. *IEEE Signal Processing Magazine*, 31(1):24–33. [1](#), [13](#), [27](#)
- Marchenko, V. A. e Pastur, L. A. (1967). Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483. [2](#)
- Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1):77–91. [27](#), [79](#)
- Mateos, G., Segarra, S., Marques, A. G., e Ribeiro, A. (2019). Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Processing Magazine*, 36(3):16–43. [10](#)
- Meinshausen, N. e Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462. [30](#)
- Nasrabadi, N. M. (2013). Hyperspectral target detection: An overview of current and future challenges. *IEEE Signal Processing Magazine*, 31(1):34–44. [1](#), [13](#)
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830. [28](#), [36](#)
- Picornell, A., Echavarria, I., Alvarez, E., López-Tarruella, S., Jerez, Y., Hoadley, K., Parker, J., del Monte-Millán, M., Ramos-Medina, R., Gayarre, J., et al. (2019). Breast cancer pam50 signature: correlation and concordance between rna-seq and digital multiplexed gene expression technologies in a triple negative breast cancer series. *BMC genomics*, 20:1–11. [13](#)
- Rainio, O., Teuho, J., e Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14(1):6086. [63](#)
- Ronneberger, O., Fischer, P., e Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. Em *International Conference on Medical image computing and computer-assisted intervention*, páginas 234–241. Springer. [xii](#), [15](#), [16](#), [46](#)
- Schäfer, J. e Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1). [27](#), [79](#)
- Seung, S. (2018). *Connectome: How the brain’s wiring makes us who we are*. HMH. [25](#)
- Shah, B. e Bhavsar, H. (2022). Time complexity in deep learning models. *Procedia Computer Science*, 215:202–210. [46](#)

- Shapiro, S. S. e Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611. [20](#)
- Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., e Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98. [10](#)
- Smith, S. M., Miller, K. L., Salimi-Khorshidi, G., Webster, M., Beckmann, C. F., Nichols, T. E., Ramsey, J. D., e Woolrich, M. W. (2011). Network modelling methods for FMRI. *NeuroImage*, 54(2):875–891. [26](#)
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., e Poole, B. (2021). Score-based generative modeling through stochastic differential equations. Em *International Conference on Learning Representations*. [78](#)
- Sporns, O. e Betzel, R. F. (2016). Modular brain networks. *Annual review of psychology*, 67:613–640. [12](#), [26](#)
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288. [3](#)
- Varoquaux, G., Gramfort, A., Poline, J.-B., e Thirion, B. (2010). Brain covariance selection: better individual functional connectivity models using population prior. Em *Advances in Neural Information Processing Systems*, volume 23, páginas 2334–2342. [26](#), [27](#)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Å., e Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30. [86](#)
- Wang, S., Li, B. Z., Khabsa, M., Fang, H., e Ma, H. (2020). Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*. [80](#), [86](#)
- Watts, D. J. e Strogatz, S. H. (1998). Collective dynamics of “small-world” networks. *Nature*, 393(6684):440–442. [12](#)
- Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. Å. R. M., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C., e Stuart, J. M. (2013). The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120. [13](#)
- Yuan, M. e Lin, Y. (2007). Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35. [30](#)
- Zamar, R., Ruiz, M., Lafit, G., e Nogales, J. (2021). A stepwise approach for high-dimensional gaussian graphical models. *Journal of Data Science, Statistics, and Visualisation*, 1(2). [xiii](#), [21](#), [30](#), [31](#)
- Zhang, B. e Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1). [13](#)