AT-SKM-Net: An Accelerated Trainable Sampling Kaczmarz-Motzkin Framework for Linear Hard-Constraint Feasibility on Dynamic Graphs

Xiaochen Zhang **Haoyu Zhu** **Yao Zhang** **Qingchun Hou***
Zhejiang University
xiaochen.23@intl.zju.edu.cn zhuhaoyu03@zju.edu.cn
eezhangyao@zju.edu.cn houqingchun@zju.edu.cn

Abstract

Graph-structured optimization with linear constraints is fundamental to critical infrastructure but faces scalability limits due to massive strict hard constraints and high dimensionality. While recent projection-based methods such as Trainable Sampling Kaczmarz-Motzkin Net (T-SKM-Net) guarantee feasibility, they face high computational costs in dynamic environments by processing the entire constraint set and requiring expensive matrix factorizations. To bridge this gap, we propose the Accelerated Trainable-SKM (AT-SKM) Net framework. To concentrate computation on the active constraints and eliminate redundant calculations, we introduce a hybrid sampling strategy guided by a topology-aware heterogeneous GNN model. To efficiently handle topological shifts in graph-based constraints, we employ a Cholesky Update mechanism that theoretically reduces the equality projection complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$ under low-rank perturbations. Experiments on random geometric graphs, N-1 Security-Constrained DC-OPF, and minimum-cost gas transport problem demonstrate that AT-SKM reduces iteration counts by up to 85% and achieves 2.95×-7.29× SKM layer speedups, while maintaining zero constraint violations. Code is available at `https://github.com/IDO-Lab/AT-SKM-Net`.

1 Introduction

Constrained optimization is essential for decision-making in a wide range of critical engineering systems [1]. In particular, graph-based formulations are particularly critical in infrastructure domains, such as power system dispatch and transportation network flow control [2]. These applications typically face a dual challenge: on one hand, decisions must strictly satisfy hard constraints imposed by physical laws or operational requirements, where even minor violations are unacceptable; on the other hand, real-world environments are highly dynamic, for instance, sudden line failures in power grids cause instantaneous changes to the underlying graph topology [3]. This demands that solvers adapt to topological changes and produce feasible solutions within milliseconds, yet traditional optimization algorithms struggle to meet real-time control requirements when confronted with large-scale networks and frequent topological perturbations.

In recent years, neural network has been widely employed to approximate solutions for various optimization problems, due to its powerful function approximation capabilities [4, 5]. However, standard neural networks cannot inherently guarantee the feasibility of their outputs. Existing approaches to address this limitation fall into two categories. Soft constraint methods suppress violations through penalty terms but lack theoretical safety guarantees, posing significant risks in safety-critical applications [6–8]. Hard constraint methods, such as differentiable optimization

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

layers [9–11] and projection-based approaches [12, 13], can enforce feasibility but often require solving complex embedded optimization subproblems during inference, facing severe computational scalability bottlenecks that hinder their application to large-scale graph data with thousands of nodes [14].

Recently, Zhu et al. [15] proposed a Trainable Sampling Kaczmarz-Motzkin Net (T-SKM-Net) framework which successfully enables end-to-end training while guaranteeing linear hard constraint satisfaction by incorporating the SKM algorithm. However, T-SKM-Net relies on SVD to handle equality constraint projection, and topological changes necessitate expensive recomputation. Moreover, in practical optimization problems, typically only a few constraints are active. Consequently, blind uniform sampling wastes substantial computation budget on redundant constraints, then limiting convergence speed.

To address these challenges, we propose AT-SKM-Net, a trainable hard-constraint correction framework for dynamic graph-structured optimization. Instead of simply treating feasibility correction as a post-processing step, AT-SKM-Net aligns the correction procedure with two structural properties of graph-based constrained problems, low-rank topology-induced perturbations and sparse active constraints. Our main contributions are as follows:

- We formulate a graph structure-aware trainable framework that preserves the hard-feasibility guarantee of projection-based SKM while solving the real-time scalability bottlenecks caused by large-scale constraints and topological perturbations.
- We accelerate the equality projection and inequality iteration bottleneck stages of SKM-based correction. For equality projection, we introduce Cholesky updates, reducing the adaptation complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$ under low-rank topological perturbations. For inequality iteration, we introduce active-set-guided hybrid sampling to focus computation on likely binding constraints.
- We develop a heterogeneous graph neural network (HGNN) tailored to dynamic topologies to provide both warm-start candidates and active-set scores for AT-SKM-Net. Experiments demonstrate that AT-SKM-Net achieves substantial speedups while maintaining zero equality and inequality constraint violations.

2 Related Work

Existing work has made significant progress in improving the feasibility of solutions produced by neural solvers and traditional optimization methods. The literature relevant to our work can be broadly categorized into three distinct streams: neural methods for hard constraint satisfaction, acceleration strategies for the SKM algorithm, and applications of active-set prediction.

2.1 Neural Methods for Hard Constraint Satisfaction

Differentiable Optimization Layer-Based Methods. These approaches integrate optimization problems or iterative operations directly into the neural inference pipeline. While exact solvers like OptNet [9] and CvxPyLayers [10] enable differentiability for quadratic programming (QP) and general convex problems, their high computational complexity limits scalability. SCQPTH [16] accelerates solving via first-order ADMM, it remains restricted to convex QP formulations. Similarly, iterative projection methods such as LinSATNet [17] and the entropy-regularized GLinSAT [4] achieve constraint satisfaction but incur high computational costs due to the necessity of extensive iterations.

Feasible Set Parameterization Methods. These methods map the latent space onto the feasible region. CP [18] parameterizes solutions for linear inequalities but struggles with high dimensionality. While Rayen [19] and scaling-based approaches [20] extend support to quadratic and other convex constraints, they rely on a pre-identified strictly feasible interior point, rendering them ill-suited for constraints that vary dynamically with inputs. Safety decision rules [14] ensure feasibility via convex combinations of a solver and a safety network, but they require convex constraint sets and inherently compromise optimality.

Correction and Projection Methods. These methods correct infeasible predictions through post-processing or iterative updates. DC3 [21] employs equality completion and inequality correction but

strictly guarantees only equality satisfaction. ProjectNet [22] utilizes alternating projections but is limited to simple non-negative constraints. Rashwan et al. [23] enforce hard linear constraints via the Component-Averaged Dykstra algorithm. T-SKM-Net [15] achieves rigorous satisfaction of both equality and inequality constraints via null-space projection and SKM iterations. However, the use of SVD factorization and uniform constraint sampling may not fully utilize the structure of sparse active constraint sets and constraint matrix, limiting inference efficiency.

2.2 Acceleration Methods for SKM

Among efforts to accelerate Kaczmarz-type iterative methods, the Greedy Kaczmarz algorithm [24] has been proven to achieve the theoretically optimal convergence rate. However, each iteration requires scanning all constraint rows to find the maximum residual, resulting in prohibitively high per-step computational cost that hinders application to large-scale problems. While Randomized Block SKM [25] achieves faster convergence rates through the parallel iteration of multiple constraints, it is plagued by the difficulty of tuning parameters like sampling scale and block size. Probably Accelerated SKM (PASKM) [26] incorporates Nesterov momentum into the SKM framework, leveraging historical gradient information to accelerate convergence toward the feasible region. Nevertheless, this method still employs uniform sampling and does not fundamentally address the inefficiency of constraint sampling.

2.3 Application of Active-Set Prediction

Xiang et al. [27] demonstrated that active set prediction can effectively reduce both the computational time and memory usage required for solving Lasso problems. Similarly, Misra et al. [28] leveraged active set prediction to mitigate the complexity of constrained optimization; however, their approach necessitates solving reduced KKT systems for all identified candidate active sets, which can lead to substantial computational overhead if the prediction accuracy is insufficient. Ng et al. [29] propose ensemble-based active set prediction, predicting active constraints at the optimal solution based on input parameters. However, this method assumes the constraint system itself remains unchanged and cannot handle active set shifts caused by topological variations. Schmidtobreick et al. [30] proposed using GNN to predict active sets for warm-starting conventional solvers.

3 Preliminaries

Given a graph $\mathcal{G}$, consider the objective function F and the feasibility of the state vector $\mathbf{x} \in \mathbb{R}^n$ subject to linear constraints:

$$\begin{aligned} \min_{\mathbf{x}} \quad & F(\mathcal{G}, \mathbf{x}) \\ \text{s.t.} \quad & A(\mathcal{G})\,\mathbf{x} \le \mathbf{b},\; C(\mathcal{G})\,\mathbf{x} = \mathbf{d} \end{aligned} \tag{1}$$

where the constraint matrices $A(\mathcal{G}) \in \mathbb{R}^{n_{\text{ineq}} \times n}$ and $C(\mathcal{G}) \in \mathbb{R}^{n_{\text{eq}} \times n}$, with $\mathbf{b} \in \mathbb{R}^{n_{\text{ineq}}}$ and $\mathbf{d} \in \mathbb{R}^{n_{\text{eq}}}$ representing n_{ineq} inequality constraints and n_{eq} equality constraints, respectively.

The SKM with null space projection method first obtains an equality-feasible point by projecting an initial guess $\mathbf{x}_0$ onto the equality-constraint hyperplane via: [15]

$$\mathbf{x}_{\text{eq}} = \mathbf{x}_0 - C^{\dagger}(C\mathbf{x}_0 - \mathbf{d}). \tag{2}$$

Let $N \in \mathbb{R}^{n \times r}$ be a basis for $\text{null}(C)$. Any equality-feasible iterate admits $\mathbf{x} = \mathbf{x}_{\text{eq}} + N\boldsymbol{\omega}$ with $\boldsymbol{\omega} \in \mathbb{R}^r$.

Let $\tilde{A} = AN$ with rows $\tilde{\mathbf{a}}_i^{\top}$, and $\tilde{\mathbf{b}} = \mathbf{b} - A\mathbf{x}_{\text{eq}}$. The feasible set in null-space coordinates is $\Omega = \{\omega \in \mathbb{R}^r \mid \tilde{A}\omega \le \tilde{b}\}$. Using a sampled constraint set $\mathcal{S}_k \subseteq \{1, \ldots, n_{\text{ineq}}\}$ and the hinge operator $[t]_+ \triangleq \max\{t, 0\}$, the SKM update on the null-space coefficients is:

$$\begin{aligned} i_k &\in \arg\max_{i \in \mathcal{S}_k} \left[\tilde{\mathbf{a}}_i^{\top}\boldsymbol{\omega}_k - \tilde{b}_i\right]_+, \\ \boldsymbol{\omega}_{k+1} &= \boldsymbol{\omega}_k - \alpha\, \frac{\left[\tilde{\mathbf{a}}_{i_k}^{\top}\boldsymbol{\omega}_k - \tilde{b}_{i_k}\right]_+}{\|\tilde{\mathbf{a}}_{i_k}\|_2^2}\, \tilde{\mathbf{a}}_{i_k}, \\ \mathbf{x}_{k+1} &= \mathbf{x}_{\text{eq}} + N\boldsymbol{\omega}_{k+1}, \quad \boldsymbol{\omega}_0 = \mathbf{0}, \end{aligned} \tag{3}$$

where $\alpha \in (0, 2)$ is the step-size. Equation (3) enforces $C\mathbf{x}_k = \mathbf{d}$ while progressively reducing violations of the transformed inequalities.

4 Accelerated Trainable-SKM Framework

We propose the Accelerated Trainable Sampling Kaczmarz-Motzkin Net (AT-SKM-Net) framework to resolve the conflict between strict feasibility guarantees and real-time scalability in dynamic environments. As shown in Figure 1, given a graph instance with updated topology and constraint features, the HGNN first produces a warm-start candidate and active-set scores. The warm-start solution is then projected onto the equality-constraint manifold, and SKM iterations are performed in the null space to remove the remaining inequality violations. The complete inference procedure is summarized in Appendix A.1. The framework aims to accelerate these two stages by using complementary mechanisms for equality projection and inequality iteration to achieve end-to-end speedup:

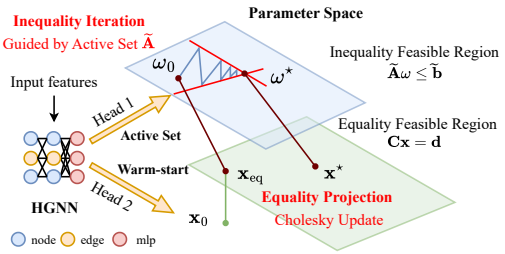


Figure 1: Flowchart of the AT-SKM-Net.

Inequality SKM Iteration: Sampling Acceleration. To overcome the inefficiency of uniform sampling, we introduce a hybrid strategy guided by Active-Set Prediction in Sec. 4.1. By leveraging topological priors to identify binding constraints, we effectively shrink the search space from the massive global set to the local active manifold. We provide the theoretical analysis in Sec. 4.2.

Equality Projection: Computing Acceleration. To address the bottleneck of SVD refactorization under low-rank constraint matrix perturbations, we propose a Cholesky Update mechanism in Sec. 4.3. This utilizes the low-rank nature of constraint matrix to theoretically reduce the equality projection complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$.

These modules are powered by our proposed Heterogeneous GNN in Sec. 4.4, which encodes the dynamic graph structure to warm-start the solver and guide the optimization process.

4.1 Accelerated Sampling for SKM Iteration: Active-Set Prediction

We address the structural inefficiency of standard SKM methods arising from the mismatch between constraints sparsity and sampling uniformity. The geometry of the feasible region is typically characterized by a sparse set of active constraints, while the vast majority of constraints are redundant at the solution. However, standard SKM remains oblivious to this structure and samples uniformly from the total constraint set m. This kind of blind search leads to inefficient sampling, where the likelihood of selecting an informative constraint is significantly diminished by the large number of inactive constraints.

To bridge this gap, we propose a neural-network-based hybrid sampling strategy that concentrates sampling probability on the active constraints.

4.1.1 Hybrid Sampling Strategy via Predicted Active-sets

Consider the linear inequality constraints over the polyhedron:

$$P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \le \mathbf{b}\}, \tag{4}$$

where $A \in \mathbb{R}^{m \times n}$ and $\mathbf{a}_i^\top$ denotes the i-th row of A. Let $\mathbf{x}^*$ be the targeted feasible solution.

At $\mathbf{x}^*$, the active constraint set is defined as

$$\mathcal{A}^*(\mathbf{x}^*) = \{i \in \{1, \dots, m\} \mid \mathbf{a}_i^\top \mathbf{x}^* = b_i\}. \tag{5}$$

In a neighborhood of $\mathbf{x}^*$, the feasible region is locally characterized by inequalities $A_{\mathcal{A}^*}\mathbf{x} \le \mathbf{b}_{\mathcal{A}^*}$ since other inequality constraints are still slack.

To address the structural sampling inefficiency, we propose a hybrid sampling strategy guided by neural network predictions. This strategy aims to concentrate computational resources on constraints with high active probability, while retaining a component of uniform sampling to guarantee theoretical global convergence.

Let $\phi(\mathcal{G})$ be the neural network model. For an input optimization instance $\mathcal{G}$, the model outputs unnormalized logits $\mathbf{z} \in \mathbb{R}^m$. We transform the predicted active scores into a sampling distribution $\mathbf{p}$ via global normalization:

$$p_i = \frac{\sigma(z_i)}{\sum_{j=1}^{m} \sigma(z_j)}, \quad \forall i \in \{1, \dots, m\}. \tag{6}$$

In each SKM iteration k, with a total batch size β and a mixing ratio $\rho \in (0, 1]$, the sampled index set τ_k is constructed from two components:

Prediction-Guided Exploitation (τ_{learn}). We select $\beta_1 = \lfloor \rho \cdot \beta \rfloor$ indices by weighted sampling based on distribution $\mathbf{p}$. This step mimics sampling from the predicted active set $\hat{\mathcal{A}}$, aiming to accelerate convergence by significantly reducing the effective sampling space dimension from m to approximately $|\hat{\mathcal{A}}|$.

Uniform Exploration (τ_{unif}). We uniformly sample $\beta_2 = \beta - \beta_1$ indices from the full constraint index set $\{1, \dots, m\}$. This step acts as a safeguard, preventing the omission of true active constraints due to prediction errors and ensuring the global robustness of the algorithm.

The final sampled constraint subset for iteration k is $\tau_k = \tau_{\text{learn}} \cup \tau_{\text{unif}}$. This hybrid approach effectively leverages prior geometric information while overcoming the potential blind spots of purely prediction-based methods.

4.2 Theoretical Analysis

To quantify the acceleration achieved by active-set prediction, we introduce the following assumption regarding the local geometry.

Assumption 1 (Local Active-Set Stability)**.** Let $\bar{\omega}$ satisfy $\mathbf{x}^* = \mathbf{x}_{\text{eq}} + N\bar{\omega}$. There exists a ball $\mathcal{U} = B(\bar{\omega}, \delta)$ with $\delta > 0$ such that for any iterate $\omega \in \mathcal{U}$, all constraints inactive at $\bar{\omega}$ remain slack at $\Pi_\Omega(\omega)$. That is, $\mathcal{A}(\Pi_\Omega(\omega)) \subseteq \mathcal{A}^*$, where $\mathcal{A}(\omega)$ denotes the constraints active at ω.

We provide a unified theoretical analysis of the acceleration mechanism. We first establish the global convergence guarantee of the algorithm, and then quantify its acceleration relative to standard SKM in the local linear regime.

Theorem 1 (Global Convergence and Robustness)**.** *For the null-space iteration* (3) *with mixing ratio $\rho < 1$, the proposed algorithm maintains global linear convergence in expectation of the squared distance to the feasible set Ω, with*

$$\mathbb{E}_k[d_{k+1}^2] \leq \left(1 - \frac{\alpha(2-\alpha)}{m\mathcal{H}^2(\tilde{A})}\right) d_k^2. \tag{7}$$

Here $d_k = \operatorname{dist}(\omega_k, \Omega)$, $\alpha \in (0, 2)$ is the step size, $\mathcal{H}(\tilde{A})$ is the Hoffman constant, and $\mathbb{E}_k$ averages over the current batch.

Proof Sketch. The proof relies on the positive sampling probabilities. Since $\rho < 1$, every constraint retains a strictly positive probability of being sampled. The maximum violation in the hybrid batch is at least that of a uniform draw. Combining the resulting expected decrease in squared distance [31, 32] with the global Hoffman bound [33, 34] yields (7), ensuring robustness against prediction errors. Complete proof in Appendix A.2. □

Theorem 2 (Efficiency Gain via Active-Set Prediction)**.** *Consider the iterate in the local linear regime from Assumption 1 with row-normalized constraints. With $p_* = \min_{i \in \mathcal{A}^*} p_i$ and $H_* = \mathcal{H}(\tilde{A}_{\mathcal{A}^*})$, the complete hybrid batch satisfies*

$$\mathbb{E}_k[d_{k+1}^2] \leq \left[1 - \frac{\alpha(2-\alpha)}{H_*^2} \max\left\{\frac{1}{m}, p_*\right\}\right] d_k^2. \tag{8}$$

Proof Sketch. The acceleration stems from the distance contraction behavior in the local linear regime by Assumption 1. In neighborhood $\mathcal{U}$, projection optimality allows the active subsystem residual to bound the distance to Ω. The uniform and guided branches yield sampling factors $1/m$ and p_*, respectively. Applying the larger of these two bounds to the complete batch proves (8). Complete proof in Appendix A.2. □

This theorem shows the acceleration through sampling efficiency. By concentrating sampling probability on active constraints, the guided branch improves the local contraction bound when $p_* > 1/m$.

4.3 Accelerated Computing for Equality Projection: Cholesky Update

In T-SKM-Net, the SKM layer first projects the neural prediction $\mathbf{x}_0$ onto the equality manifold $\mathbf{C}\mathbf{x} = \mathbf{d}$ and then performs inequality iterations in the null space of C. This equality projection is essential for preserving feasibility, but it becomes a major bottleneck in dynamic graph problems. When the graph topology changes, the equality matrix C also changes, and the SVD-based projection used in T-SKM-Net has to be recomputed under an expensive $\mathcal{O}(N^3)$ computational cost for each perturbed instance, where N denotes the problem size [35].

However, mild topological changes, such as line outages or reconnections, typically induce localized modifications in the adjacency matrix or the graph Laplacian. Motivated by this observation, we propose an incremental Cholesky update framework that exploits such structured changes to accelerate the equality-constraint projection phase.

Given a row full-rank matrix $\mathbf{C} \in \mathbb{R}^{m\times n}$, a rank-1 perturbation defined by $\tilde{\mathbf{C}} = \mathbf{C} + \mathbf{u}\mathbf{v}^\top$ induces a rank-2 modification in its corresponding Gram matrix $\mathbf{G} = \mathbf{C}\mathbf{C}^\top$. To handle this efficiently, we bypass computationally expensive SVD re-computations in favor of a generic incremental Cholesky update strategy. By sequentially applying one rank-1 update and one rank-1 downdate to the original $\mathbf{G} = \mathbf{L}\mathbf{L}^\top$, we can directly evolve the triangular factor $\tilde{\mathbf{L}}$ to adapt to the new topology without reconstructing the matrix from scratch.

Based on the updated factor $\tilde{\mathbf{L}}$, we transform the explicit matrix inversion into two efficient triangular matrix solves. For the constraint residual $\mathbf{r} = \tilde{\mathbf{C}}\mathbf{x}_0 - \mathbf{d}$, the projected solution can be expressed as:

$$\tilde{\mathbf{x}}_{\text{eq}} = \mathbf{x}_0 - \tilde{\mathbf{C}}^\top(\tilde{\mathbf{L}}^{-\top}(\tilde{\mathbf{L}}^{-1}\mathbf{r})) \tag{9}$$

The null-space representation used by the subsequent SKM iterations is updated consistently with the perturbed equality constraints. Instead of reconstructing the whole null-space basis from scratch, we reuse the previous basis and correct only the low-dimensional subspace affected by the perturbation [36]. Intuitively, a low-rank change in $\mathbf{C}$ only alters a small number of directions in the null space, while the remaining directions can be retained after an orthogonal alignment. The explicit construction of this basis update is provided in Appendix B.1.

Computational Complexity. While the SVD approach necessitates a full re-factorization with cubic complexity $\mathcal{O}(N^3)$, our proposed framework significantly reduces the computational burden by exploiting the low-rank perturbations of constraint matrix. By replacing global decomposition with localized Cholesky rank-1 updates and efficient basis rotations, we effectively decouple the high-dimensional problem into sequential vector operations and triangular solves. This strategy lowers the overall complexity to $\mathcal{O}(N^2)$, offering a theoretical speedup of order N compared to standard methods. Description with more details is in Appendix B.2.

4.4 Topology-Aware Heterogeneous Graph Neural Network for Active-Set Prediction

The hybrid sampling strategy proposed in Section 4.1 relies on precise estimation of active probabilities for individual constraints, which poses two key challenges: capturing the heterogeneity of constraints, and generalizing across varying topological scenarios. To address such challenges, we propose a heterogeneous graph neural network (HGNN) architecture that aligns with the structure of constraints and topology of graph-constrained optimization problems.

4.4.1 Heterogeneous Graph Representation

In graph-based optimization problems, constraints are naturally partitioned into node-level constraints and edge-level constraints. Homogeneous GNNs treat edges as message-passing channels, which tends to smooth edge-specific information into node representations [37]. This smoothing effect hinders the extraction of distinct active probabilities for edge-based couplings.

To address this, we adopt a heterogeneous bipartite graph architecture where both original nodes and edges are modeled as distinct computational entities. This design establishes a structural correspondence between neural network embeddings and the optimization constraint set. Let $\mathbf{h}_u^{(l)}$ and $\mathbf{h}_e^{(l)}$ denote the latent representations of node u and edge $e = (u, v)$ at layer l. The heterogeneous message passing updates are:

$$\mathbf{h}_e^{(l+1)} = \phi_{\text{edge}}\left(\mathbf{h}_e^{(l)}, \mathbf{h}_u^{(l)}, \mathbf{h}_v^{(l)}\right), \qquad \mathbf{h}_u^{(l+1)} = \phi_{\text{node}}\left(\mathbf{h}_u^{(l)}, \sum_{e \in \mathcal{N}(u)} \mathbf{h}_e^{(l+1)}\right), \tag{10}$$

where ϕ_{edge} and ϕ_{node} are nonlinear transition functions. This formulation preserves the representation $\mathbf{h}_e$ of each coupling constraint independently throughout the propagation depth, rather than absorbing it into nodal states.

4.4.2 Static-Dynamic Feature Separation

When the graph topology changes, the neural model needs to generalize across these perturbations. We observe that problem inputs can be decomposed into two categories: static geometric parameters $\mathbf{z}_{\text{stat}}$ that define the constraint coefficient matrix A, and dynamic instance conditions $\mathbf{z}_{\text{dyn}}$ that define the boundary vector $\mathbf{b}$.

To prevent static structural information from being diluted by dynamic variations during deep message passing, we employ a structure-aware normalization mechanism. The mechanism first concatenates the dynamic hidden features $\mathbf{h}^{(l)}$ with static physical features $\mathbf{z}_{\text{stat}}$, then generates the normalization parameters through graph convolution:

$$\mathbf{h}^{(l+1)} = \gamma(\mathbf{h}^{(l)}, \mathbf{z}_{\text{stat}}) \odot \frac{\mathbf{h}^{(l)} - \mu}{\sigma} + \beta(\mathbf{h}^{(l)}, \mathbf{z}_{\text{stat}}), \tag{11}$$

where the affine parameters γ and β are generated from the concatenation $[\mathbf{h}^{(l)} \| \mathbf{z}_{\text{stat}}]$. This formulation is related to adaptive normalization method proposed by [38], but instead of injecting random noise, we condition on the global static topology of the graph. This helps ensure that predictions remain consistent with constraints of $\mathcal{G}$ when its topology changes.

4.4.3 Output Heads

The HGNN serves as a generator for the accelerated solver. Its final embeddings are fed into two output heads. A regression head predicts the primal variable candidate $\hat{\mathbf{x}}_0$ to provide a warm start for the SKM layer. A classification head generates active scores for both node and edge constraints. The sampling distribution $\mathbf{p}$ required by the hybrid strategy is obtained via global normalization:

$$p_k = \frac{\sigma(\text{MLP}(\mathbf{h}_k))}{\sum_{j \in \mathcal{V} \cup \mathcal{E}} \sigma(\text{MLP}(\mathbf{h}_j))}, \tag{12}$$

where k indexes over all node and edge constraints.

5 Experiments

We evaluate the proposed AT-SKM method on random geometric graphs and N-1 security-constrained DC optimal power flow problem, and provide an additional validation on the minimum-cost gas transport problem in Appendix C. All experiments are conducted on an Apple M4 chip.

5.1 Equality Projection on Random Geometric Graphs

We validate the asymptotic computational complexity of the proposed Cholesky update method for equality-constraint projection using random geometric graphs (RGGs) of varying sizes. In an RGG,

nodes are uniformly distributed within the unit square, with edges connecting pairs whose Euclidean distance falls below a threshold r; this local connectivity structure closely resembles real-world networks where physical proximity constrains connections.

Figure 2 compares the computation time of the Cholesky Update method, which incrementally updates the initial factorization, against SVD recomputation that performs a full decomposition for each topology change. On a log-log scale, the SVD computation time closely follows the $\mathcal{O}(N^3)$ reference curve as the number of nodes increases. Meanwhile, the empirical runtime of our proposed method aligns well with $\mathcal{O}(N^2)$. Notably, when the graph size reaches approximately 2000 nodes, SVD requires seconds to complete, which is prohibitive for safety-critical real-time applications such as power system dispatch. In contrast, the Cholesky update maintains computation times on the order of 10^{-2} seconds, demonstrating a substantial efficiency advantage. A detailed breakdown of these computational overheads is provided in Appendix B.3.

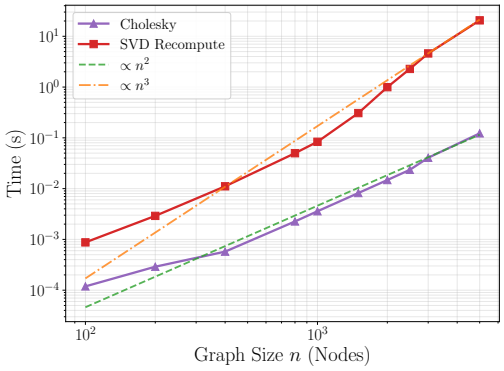


Figure 2: Computational complexity comparison between SVD recomputation and Cholesky update on RGGs. SVD exhibits $\mathcal{O}(N^3)$ scaling while Cholesky update achieves $\mathcal{O}(N^2)$.

5.2 N-1 Security-Constrained DC Optimal Power Flow

We evaluate the performance of our proposed method on the N-1 Security-Constrained DC Optimal Power Flow (N-1 SC DC-OPF) problem. This problem stands as a fundamental challenge in critical power system infrastructure and represents a distinct class of graph-based constrained optimization characterized by strict safety requirements and dynamic network topologies [39].

The primary objective is to minimize costs subject to physical constraints, including power balance equations, generator output limits, and transmission line thermal limits. The rigorous N-1 security criterion mandates that the system must maintain feasibility following the loss of a single transmission component [40]. The detailed mathematical formulation is provided in Appendix D.1.

The experiment is based on IEEE 57-bus, 118-bus, and 300-bus systems. While we consider all feasible N-1 contingencies, we include only a subset of these scenarios in the training set to assess the model's adaptability and generalization to unseen topologies. The testing set evaluates the solver's performance in terms of cost optimization and constraint satisfaction across both seen and unseen topologies. The ground truth solutions for all instances are generated using the PyPower library [41].

We benchmark our method against: (1) **PyPower**: A traditional interior-point solver serving as the ground truth; (2) **NN / HGNN**: End-to-end learning baselines without explicit constraint enforcement; (3) **DC3**: A differentiable optimization method guaranteeing equality constraints via projection; and (4) **T-SKM-Net**: The original Trainable SKM framework. To decouple the contributions of our core modules, we evaluate three variants: **AT-SKM-A** (Active-set prediction only), **AT-SKM-C** (Cholesky update only), and **AT-SKM-AC** (Full).

As shown in Table 1, our proposed method achieves significant speedups over the T-SKM-Net baseline across all system sizes, while strictly satisfying all equality and inequality constraints without compromising solution quality. Regarding solution quality, although unconstrained baselines NN and HGNN achieve small optimality gaps, they exhibit varying degrees of constraint violations, rendering them unsafe for real-world operations. Moreover, the NN shows poor generalization on unseen topologies. While DC3 satisfies equality constraints, it fails to eliminate inequality violations.

We further analyze the isolated impacts of active-set prediction and Cholesky updates:

Active-Set Prediction: AT-SKM-A reduces the average iteration count by 80%, 55%, and 85% on the 57, 118, and 300-bus systems, respectively. This confirms that our hybrid sampling strategy effectively concentrates computational resources on the sparse subset of active constraints.

Cholesky Update: AT-SKM-C significantly accelerates the projection phase. In the 300-bus system, although the results indicate that the Cholesky update does not affect the inequality iteration rate,

Table 1: Performance comparison across different methods. The Optimality Gap and Constraint Violations (tol=10^{-5}) are formatted as "Seen / Unseen". The "Iter. Num." column reports the mean (max) number of inequality iterations. The SKM-related metrics are reported on unseen scenarios to demonstrate robust acceleration.

Method	Tot. Time (ms)	Iter. Num.	SKM Time	SKM Speedup	Opt. Gap (%)	Max Eq. Vio.	Max Ineq. Vio.
IEEE 57-Bus System: $n = 64$, $n_{\text{eq}} = 58$, $n_{\text{ineq}} = 174$							
PyPower	14.785	-	-	-	0 / 0	0 / 0	0 / 0
NN	0.048	-	-	-	0.173 / 5.379	24.77 / 53.07	0.101 / 192.71
HGNN	4.735	-	-	-	0.071 / 0.061	17.52 / 5.163	0.061 / 0.041
DC3	6.278	-	-	-	0.013 / 0.013	0 / 0	0.067 / 0.051
T-SKM-Net	5.831	6.93 (148)	1.096	1×	0.008 / 0.008	0 / 0	0 / 0
AT-SKM-A	5.202	**1.42 (11)**	0.467	2.35×	0.008 / 0.008	0 / 0	0 / 0
AT-SKM-C	5.353	6.93 (148)	0.618	1.77×	0.008 / 0.008	0 / 0	0 / 0
AT-SKM-AC	5.009	**1.42 (11)**	**0.274**	**4.00**×	0.008 / 0.008	0 / 0	0 / 0
IEEE 118-Bus System: $n = 172$, $n_{\text{eq}} = 119$, $n_{\text{ineq}} = 480$							
PyPower	31.557	-	-	-	0 / 0	0 / 0	0 / 0
NN	0.057	-	-	-	0.142 / 1.481	72.59 / 93.21	0.031 / 664.11
HGNN	6.938	-	-	-	0.017 / 0.016	61.06 / 67.36	0.010 / 0.047
DC3	8.040	-	-	-	0.019 / 0.019	0 / 0	0.011 / 0.009
T-SKM-Net	10.251	34.60 (141)	3.313	1×	0.026 / 0.025	0 / 0	0 / 0
AT-SKM-A	9.349	**15.68 (57)**	2.411	1.37×	0.026 / 0.025	0 / 0	0 / 0
AT-SKM-C	8.942	34.60 (141)	2.004	1.65×	0.026 / 0.025	0 / 0	0 / 0
AT-SKM-AC	8.060	**15.68 (57)**	**1.122**	**2.95**×	0.026 / 0.025	0 / 0	0 / 0
IEEE 300-Bus System: $n = 369$, $n_{\text{eq}} = 301$, $n_{\text{ineq}} = 960$							
PyPower	64.398	-	-	-	0 / 0	0 / 0	0 / 0
NN	0.078	-	-	-	0.057 / 0.367	2.00e+3 / 9.44e+3	1.53e+3 / 8.52e+3
HGNN	9.390	-	-	-	0.068 / 0.068	64.61 / 62.36	537.66 / 0.086
DC3	16.138	-	-	-	0.102 / 0.102	0 / 0	0.471 / 0.465
T-SKM-Net	18.382	24.72 (285)	8.992	1×	0.108 / 0.108	0 / 0	0 / 0
AT-SKM-A	17.264	**3.68 (9)**	7.874	1.14×	0.108 / 0.108	0 / 0	0 / 0
AT-SKM-C	12.179	24.72 (285)	2.789	3.22×	0.110 / 0.110	0 / 0	0 / 0
AT-SKM-AC	10.623	**3.69 (9)**	**1.233**	**7.29**×	0.110 / 0.110	0 / 0	0 / 0

AT-SKM-C reduces the SKM runtime from 8.99ms to 2.79ms ($3.22\times$ speedup), whereas AT-SKM-A reduces iterations by 85% but only yields a $1.14\times$ speedup. This reveals that in large-scale systems, the computational bottleneck shifts from the iterative search to linear algebraic operations, highlighting the $\mathcal{O}(n^2)$ advantage of our Cholesky update.

Interestingly, we observe a synergistic effect over these two methods. The total speedup of AT-SKM-AC approximates or even exceeds the product of the individual speedups. For instance, on the 57-bus system, $S_A(2.35\times) \times S_C(1.77\times) \approx S_{AC}(4.00\times)$, and on the 300-bus system, the combined speedup ($7.29\times$) significantly surpasses the product of individual gains $S_A(1.14\times) \times S_C(3.22\times) \ll S_{AC}(7.29\times)$, suggesting a super-linear efficiency improvement.

Further experiments are provided in the appendix: hybrid sampling sensitivity (D.4), detailed runtime breakdown (D.5), high-load robustness (D.6), active-set prediction quality (D.7), active-set perturbation (D.8), comparison with additional SKM variants (D.9), and transfer learning with pseudo-label active-set training (D.10).

6 Conclusion

In this paper, we propose the AT-SKM framework to effectively address the trade-off between strict feasibility and real-time scalability in graph-structured optimization. By combining neural network active-set predictions with efficient Cholesky incremental updates, AT-SKM-Net overcomes the computational bottlenecks that limit the original projection-based T-SKM-Net in dynamic environments.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Key R&D Program of China (No. 2024YFB2409300) and in part by the General Program of the National Natural Science Foundation of China (No. 52677165).

References

[1] James Kotary, Ferdinando Fioretto, Pascal Van Hentenryck, and Bryan Wilder. End-to-end constrained optimization learning: A survey. In *International Joint Conference on Artificial Intelligence*, 2021.

[2] Wei Wei, WU Danman, WU Qiuwei, Miadreza Shafie-Khah, and João PS Catalão. Interdependence between transportation system and power distribution system: A comprehensive review on models and applications. *Journal of Modern Power Systems and Clean Energy*, 7(3): 433–448, 2019.

[3] Mojtaba Khanabadi, Hassan Ghasemi, and Meysam Doostizadeh. Optimal transmission switching considering voltage security and n-1 contingency analysis. *IEEE Transactions on Power Systems*, 28(1):542–550, 2013.

[4] Hongtai Zeng, Chao Yang, Yanzhen Zhou, Cheng Yang, and Qinglai Guo. Glinsat: The general linear satisfiability neural network layer by accelerated gradient descent. *Advances in Neural Information Processing Systems*, 37:122584–122615, 2024.

[5] Utkarsh Utkarsh, Danielle C. Maddix, Ruijun Ma, Michael W. Mahoney, and Yuyang Wang. End-to-end probabilistic framework for learning with hard constraints. *arXiv preprint arXiv:2506.07003*, 2025.

[6] Ferdinando Fioretto, Pascal Van Hentenryck, Terrence WK Mak, Cuong Tran, Federico Baldo, and Michele Lombardi. Lagrangian duality for constrained deep learning. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 118–135. Springer, 2020.

[7] Luiz FO Chamon, Santiago Paternain, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained learning with non-convex losses. *IEEE Transactions on Information Theory*, 69(3): 1739–1760, 2022.

[8] James Kotary and Ferdinando Fioretto. Learning constrained optimization with deep augmented lagrangian methods. *arXiv preprint arXiv:2403.03454*, 2024.

[9] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pages 136–145. PMLR, 2017.

[10] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.

[11] Youngjae Min and Navid Azizan. Hardnet: Hard-constrained neural networks with universal approximation guarantees. *arXiv preprint arXiv:2410.10807*, 2024.

[12] Enming Liang, Minghua Chen, and Steven H. Low. Low complexity homeomorphic projection to ensure neural-network solution feasibility for optimization over (non-)convex set. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[13] Xiang Pan, Tianyu Zhao, Minghua Chen, and Shengyu Zhang. Deepopf: A deep neural network approach for security-constrained dc optimal power flow. *IEEE Transactions on Power Systems*, 36(3):1725–1735, 2020.

[14] Gonzalo E Constante-Flores, Hao Chen, and Can Li. Enforcing hard linear constraints in deep learning models with decision rules. *arXiv preprint arXiv:2505.13858*, 2025.

[15] Haoyu Zhu, Yao Zhang, Jiashen Ren, and Qingchun Hou. T-skm-net: Trainable neural network framework for linear constraint satisfaction via sampling kaczmarz-motzkin method. *arXiv preprint arXiv:2512.10461*, 2025.

[16] Andrew Butler. Scqpth: an efficient differentiable splitting method for convex quadratic programming. *arXiv preprint arXiv:2308.08232*, 2023.

[17] Runzhong Wang, Yunhao Zhang, Ziao Guo, Tianyi Chen, Xiaokang Yang, and Junchi Yan. Linsatnet: the positive linear satisfiability neural networks. In *International Conference on Machine Learning*, pages 36605–36625. PMLR, 2023.

[18] Thomas Frerix, Matthias Nießner, and Daniel Cremers. Homogeneous linear inequality constraints for neural network activations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 748–749, 2020.

[19] Jesus Tordesillas, Jonathan P How, and Marco Hutter. Rayen: Imposition of hard convex constraints on neural networks. *arXiv preprint arXiv:2307.08336*, 2023.

[20] Andrei V Konstantinov and Lev V Utkin. A new computationally simple approach for implementing neural networks with output hard constraints. In *Doklady Mathematics*, volume 108, pages S233–S241. Springer, 2023.

[21] Priya Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*, 2021.

[22] Rares Cristian, Pavithra Harsha, Georgia Perakis, Brian L Quanz, and Ioannis Spantidakis. End-to-end learning for optimization via constraint-enforcing approximators. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 7253–7260, 2023.

[23] Ahmed Rashwan, Keith Briggs, Chris Budd, and Lisa Kreusser. Enforcing convex constraints in graph neural networks. *arXiv preprint arXiv:2510.11227*, 2025.

[24] Julie Nutini, Behrooz Sepehry, Issam Laradji, Mark Schmidt, Hoyt Koepke, and Alim Virani. Convergence rates for greedy kaczmarz algorithms, and faster randomized kaczmarz rules using the orthogonality graph. *arXiv preprint arXiv:1612.07838*, 2016.

[25] Yanjun Zhang and Hanyu Li. Randomized block subsampling kaczmarz-motzkin method. *Linear Algebra and its Applications*, 667:133–150, 2023.

[26] Md Sarowar Morshed, Md Saiful Islam, and Md Noor-E-Alam. Sampling kaczmarz-motzkin method for linear feasibility problems: generalization and acceleration. *Mathematical Programming*, 194(1):719–779, 2022.

[27] Zhen James Xiang, Yun Wang, and Peter J Ramadge. Screening tests for lasso problems. *IEEE transactions on pattern analysis and machine intelligence*, 39(5):1008–1027, 2016.

[28] Sidhant Misra, Line Roald, and Yeesian Ng. Learning for constrained optimization: Identifying optimal active constraint sets. *INFORMS Journal on Computing*, 34(1):463–480, 2022.

[29] Yeesian Ng, Sidhant Misra, Line A. Roald, and Scott Backhaus. Statistical learning for dc optimal power flow. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7, 2018.

[30] Ella J Schmidtobreick, Daniel Arnström, Paul Häusner, and Jens Sjölund. Warm-starting active-set solvers using graph neural networks. *arXiv preprint arXiv:2511.13174*, 2025.

[31] Jamie Haddock and Anna Ma. Greed works: An improved analysis of sampling kaczmarz–motzkin. *SIAM Journal on Mathematics of Data Science*, 3(1):342–368, 2021.

[32] Jesus A De Loera, Jamie Haddock, and Deanna Needell. A sampling kaczmarz–motzkin algorithm for linear feasibility. *SIAM Journal on Scientific Computing*, 39(5):S66–S87, 2017.

[33] Diethard Klatte and Gisbert Thiere. Error bounds for solutions of linear equations and inequalities. *Zeitschrift für Operations Research*, 41(2):191–214, 1995.

[34] Javier Peña, Juan C. Vera, and Luis F. Zuluaga. New characterizations of Hoffman constants for systems of linear constraints. *Mathematical Programming*, 187(1-2):79–109, 2021.

[35] Nisheeth K Vishnoi. Laplacian solvers and their algorithmic applications. *Foundations and Trends® in Theoretical Computer Science*, 8(1-2):1–145, 2010.

[36] Jorge Nocedal and Stephen J. Wright. Numerical optimization. *Springer Ser. Oper. Res. Financ. Eng./Springer*, 2006.

[37] Salah Ghamizi, Aoxiang Ma, Jun Cao, and Pedro Rodriguez Cortes. Opf-hgnn: Generalizable heterogeneous graph neural networks for ac optimal power flow. In *2024 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2024.

[38] Moshe Eliasof, Beatrice Bevilacqua, Carola-Bibiane Schönlieb, and Haggai Maron. Granola: Adaptive normalization for graph neural networks. *Advances in Neural Information Processing Systems*, 37:90514–90551, 2024.

[39] Xiaochen Zhang, Kaijie Xu, Shengchen Liao, Lin Qiu, Chengjin Ye, and Youtong Fang. Disturbed security-constrained and time-variant optimal power flow for dynamic power system based on chaotic-genetic-centroid puffin optimization. *Applied Energy*, 397:126287, 2025.

[40] Min Zhou, Chensheng Liu, Amir Abiri Jahromi, Deepa Kundur, Jing Wu, and Chengnian Long. Revealing vulnerability of n-1 secure power systems to coordinated cyber-physical attacks. *IEEE Transactions on Power Systems*, 38(2):1044–1057, 2023.

[41] Richard Lincoln. PYPOWER. `https://pypi.org/project/PYPOWER/5.1.19/`, 2025. Version 5.1.19; released July 10, 2025; accessed January 1, 2026.

[42] Alston S Householder. Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)*, 5(4):339–342, 1958.

[43] Martin Schmidt, Denis Aßmann, Robert Burlacu, Jesco Humpola, Imke Joormann, Nikolaos Kanelakis, Thorsten Koch, Djamal Oucherif, Marc E. Pfetsch, Lars Schewe, et al. Gaslib—a library of gas network instances. *Data*, 2(4):40, 2017.

[44] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

A AT-SKM-Net Algorithm Flow and Theoretical Proofs

A.1 AT-SKM-Net Algorithm Flow

Algorithm 1 Pseudo-code of the proposed AT-SKM-Net Framework.

Require: Constraint data (A, b, C, d), predictor ϕ, sampling batch size β, mixing ratio ρ, maximum iterations K
Ensure: Feasible solution x_K

$(x_0, s) \leftarrow \phi(A, b, C, d)$
$p_i \leftarrow \sigma(s_i)/\sum_{j=1}^{m} \sigma(s_j),\ i = 1, \ldots, m$
Equality projection stage.
$L \leftarrow \textsc{CholeskyUpdate}(C)$ such that $LL^\top = CC^\top$
Solve $LL^\top y = Cx_0 - d$ and set $x_{\text{eq}} = x_0 - C^\top y$
$N \leftarrow \textsc{NullSpaceUpdate}(C, L), \quad CN = 0$
$\widetilde{A} \leftarrow AN, \quad \widetilde{b} \leftarrow b - Ax_{\text{eq}}, \quad \omega_0 \leftarrow 0$
Inequality iteration stage.
for $k = 0, 1, \ldots, K-1$ **do**
  $\beta_{\text{act}} \leftarrow \lfloor \rho\beta \rfloor, \quad \beta_{\text{unif}} \leftarrow \beta - \beta_{\text{act}}$
  Sample τ_{act} from $\{1, \ldots, m\}$ according to p with size β_{act}
  Uniformly sample τ_{unif} from $\{1, \ldots, m\}$ with size β_{unif}
  $\tau_k \leftarrow \tau_{\text{act}} \cup \tau_{\text{unif}}$
  $i_k \leftarrow \arg\max_{i \in \tau_k} [\widetilde{a}_i^\top \omega_k - \widetilde{b}_i]_+$
  if $\widetilde{a}_{i_k}^\top \omega_k > \widetilde{b}_{i_k}$ **then**
    $\omega_{k+1} = \omega_k - \alpha \frac{\widetilde{a}_{i_k}^\top \omega_k - \widetilde{b}_{i_k}}{\|\widetilde{a}_{i_k}\|_2^2} \widetilde{a}_{i_k}, \qquad \alpha \in (0, 2).$
  else
    $\omega_{k+1} \leftarrow \omega_k$
  end if
end for
$x_K \leftarrow x_{\text{eq}} + N\omega_K$

return x_K

A.2 Accelerated Sampling by Active-Set Prediction

Theorem 1 (Global Convergence and Robustness)**.** *For the null-space iteration* (3) *with mixing ratio* $\rho < 1$*, the proposed algorithm maintains global linear convergence in expectation of the squared distance to the feasible set* Ω*, with*

$$\mathbb{E}_k[d_{k+1}^2] \le \left(1 - \frac{\alpha(2-\alpha)}{m\mathcal{H}^2(\tilde{A})}\right) d_k^2. \tag{7}$$

Here $d_k = \operatorname{dist}(\omega_k, \Omega)$*,* $\alpha \in (0, 2)$ *is the step size,* $\mathcal{H}(\tilde{A})$ *is the Hoffman constant, and* $\mathbb{E}_k$ *averages over the current batch.*

Proof. The AT-SKM algorithm iterates on the null-space coefficients ω. Let $\Omega = \{\omega \in \mathbb{R}^r \mid \tilde{A}\omega \le \tilde{b}\}$ be the nonempty feasible set in the null-space representation, where $\tilde{A} = AN$ and $\tilde{b} = b - Ax_{eq}$. Let $\omega_k^* = \Pi_\Omega(\omega_k)$ be the projection of the current iterate ω_k onto Ω. We define the distance as $d_k = \|\omega_k - \omega_k^*\|_2$.

Consider the k-th iteration with $\|\tilde{a}_i\|_2 = 1$ for all i. Let $g_{j,k} = \frac{[\tilde{a}_j^\top \omega_k - \tilde{b}_j]_+^2}{\|\tilde{a}_j\|^2} = [\tilde{a}_j^\top \omega_k - \tilde{b}_j]_+^2$ denote the squared normalized violation of the row-normalized j-th constraint. The algorithm selects the active constraint index i_k from a sampled batch $\mathcal{S}_k$ using the greedy criterion:

$$i_k = \arg\max_{j \in \mathcal{S}_k} g_{j,k}. \tag{13}$$

Write $r_{i,k} = [\tilde{a}_i^\top \omega_k - \tilde{b}_i]_+$, so $g_{i,k} = r_{i,k}^2$. Since ω_k^* is feasible, whenever $r_{i,k} > 0$ we have $\tilde{a}_i^\top(\omega_k - \omega_k^*) \ge r_{i,k}$. For $r_{i,k} = 0$, both sides of the corresponding product inequality vanish. Hence

$$r_{i,k}\tilde{a}_i^\top(\omega_k - \omega_k^*) \ge r_{i,k}^2. \tag{14}$$

The standard Kaczmarz update with step size $\alpha \in (0, 2)$ yields the following distance contraction recurrence [32]. Using $\omega_{k+1} = \omega_k - \alpha r_{i_k,k}\tilde{a}_{i_k}$ and $\|\tilde{a}_{i_k}\|_2 = 1$, we obtain

$$\begin{aligned} d_{k+1}^2 &\le \|\omega_{k+1} - \omega_k^*\|_2^2 \\ &= d_k^2 - 2\alpha r_{i_k,k}\tilde{a}_{i_k}^\top(\omega_k - \omega_k^*) + \alpha^2 r_{i_k,k}^2 \\ &\le d_k^2 - \alpha(2-\alpha) g_{i_k,k}. \end{aligned} \tag{15}$$

The hybrid sampling strategy ensures that the batch $\mathcal{S}_k$ contains a subset τ_{unif} of size $\beta_{\text{unif}} \ge 1$, which is sampled uniformly from all m constraints, which is guaranteed by $\rho < 1$. The maximum violation in the full batch is bounded below by the maximum violation in the uniform subset, which is further bounded below by the violation of its first draw U_k,

$$g_{i_k,k} = \max_{j \in \mathcal{S}_k} g_{j,k} \ge \max_{j \in \tau_{\text{unif}}} g_{j,k} \ge g_{U_k,k}. \tag{16}$$

Let $\mathcal{F}_k$ contain the history before sampling batch $\mathcal{S}_k = \tau_k$, including ω_k and the predicted distribution $\mathbf{p}$, and write $\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_k]$. We take the expectation with respect to the random sampling of the batch, conditional on $\mathcal{F}_k$. Since τ_{unif} is drawn uniformly from $\{1, \dots, m\}$, $\mathbb{P}(U_k = i \mid \mathcal{F}_k) = 1/m$. By the linearity of expectation,

$$\mathbb{E}_k[g_{i_k,k}] \ge \mathbb{E}_k[g_{U_k,k}] = \frac{1}{m}\sum_{i=1}^m g_{i,k}. \tag{17}$$

The summation term corresponds to the squared norm of the residual vector $\|[\tilde{A}\omega_k - \tilde{b}]_+\|^2$. By the global Hoffman error bound theorem [33, 34], there exists a least constant $\mathcal{H}(\tilde{A})$ valid for every right-hand side with a nonempty feasible set such that

$$\sum_{i=1}^m g_{i,k} = \|[\tilde{A}\omega_k - \tilde{b}]_+\|^2 \ge \frac{1}{\mathcal{H}^2(\tilde{A})} d_k^2. \tag{18}$$

Substituting this back into the expectation of the distance recurrence gives (7). For a fixed initial iterate, the tower property then yields

$$\mathbb{E}[d_k^2] \le \left(1 - \frac{\alpha(2-\alpha)}{m\mathcal{H}^2(\tilde{A})}\right)^k d_0^2. \tag{19}$$

Under $\alpha \in (0, 2)$ and $\mathcal{H}(\tilde{A}) < \infty$, the algorithm converges linearly in expectation. □

Theorem 2 (Efficiency Gain via Active-Set Prediction)**.** *Consider the iterate in the local linear regime from Assumption 1 with row-normalized constraints. With $p_* = \min_{i \in \mathcal{A}^*} p_i$ and $H_* = \mathcal{H}(\tilde{A}_{\mathcal{A}^*})$, the complete hybrid batch satisfies*

$$\mathbb{E}_k[d_{k+1}^2] \le \left[1 - \frac{\alpha(2-\alpha)}{H_*^2} \max\left\{\frac{1}{m}, p_*\right\}\right] d_k^2. \tag{8}$$

Proof. Consider a nonempty active set $\mathcal{A}^*$ and a hybrid batch with $\beta_1, \beta_2 \ge 1$. Let $\Omega_{\mathcal{A}^*} = \{\omega \mid \tilde{A}_{\mathcal{A}^*}\omega \le \tilde{b}_{\mathcal{A}^*}\}$ be the feasible set of the active subsystem. For $\omega \in \mathcal{U}$, write $y = \Pi_\Omega(\omega)$ and $I = \mathcal{A}(y)$. Projection optimality for a polyhedron gives nonnegative multipliers λ_i such that

$$\omega - y = \sum_{i \in I} \lambda_i \tilde{a}_i, \qquad \lambda_i \ge 0. \tag{20}$$

By Assumption 1, $I \subseteq \mathcal{A}^*$. For every $z \in \Omega_{\mathcal{A}^*}$, feasibility of z and activity at y imply

$$\langle \omega - y, z - y\rangle = \sum_{i \in I} \lambda_i(\tilde{a}_i^\top z - \tilde{b}_i) \le 0. \tag{21}$$

Together with $y \in \Omega_{\mathcal{A}^*}$, this is the projection optimality condition for $\Omega_{\mathcal{A}^*}$. Thus $y = \Pi_{\Omega_{\mathcal{A}^*}}(\omega)$, and the Hoffman bound for the active subsystem [34] yields

$$\begin{aligned} \operatorname{dist}(\omega, \Omega) &= \operatorname{dist}(\omega, \Omega_{\mathcal{A}^*}) \\ &\le H_* \|[\tilde{A}_{\mathcal{A}^*}\omega - \tilde{b}_{\mathcal{A}^*}]_+\|_2. \end{aligned} \tag{22}$$

In particular, $R_{*,k} = \sum_{i \in \mathcal{A}^*} g_{i,k} \geq d_k^2 / H_*^2$.

Let J_k denote the first guided draw, with $\mathbb{P}(J_k = i \mid \mathcal{F}_k) = p_i$. Since U_k and J_k both belong to $\mathcal{S}_k$, the maximum selected violation satisfies

$$\begin{aligned} \mathbb{E}_k[g_{i_k,k}] &\geq \mathbb{E}_k[g_{U_k,k}] = \frac{1}{m} \sum_{i=1}^{m} g_{i,k} \geq \frac{1}{m} R_{*,k}, \\ \mathbb{E}_k[g_{i_k,k}] &\geq \mathbb{E}_k[g_{J_k,k}] = \sum_{i=1}^{m} p_i g_{i,k} \geq p_* R_{*,k}. \end{aligned} \tag{23}$$

Combining these inequalities with (22) gives

$$\mathbb{E}_k[g_{i_k,k}] \geq \max\left\{\frac{1}{m}, p_*\right\} R_{*,k} \geq \frac{1}{H_*^2} \max\left\{\frac{1}{m}, p_*\right\} d_k^2. \tag{24}$$

Substitution into the conditional expectation of (15) proves (8).

Finally, applying the expansion in (15) to the feasible reference point $\bar{\omega}$ gives Fejér monotonicity,

$$\|\omega_{k+1} - \bar{\omega}\|_2^2 \leq \|\omega_k - \bar{\omega}\|_2^2 - \alpha(2-\alpha) g_{i_k,k} \leq \|\omega_k - \bar{\omega}\|_2^2. \tag{25}$$

Thus, once an iterate enters $\mathcal{U} = B(\bar{\omega}, \delta)$, all subsequent iterates remain in $\mathcal{U}$, and the local bound applies at every subsequent step. □

Assumption 2 (Effective Active-Set Covering)**.** We assume the predictor identifies a candidate set $\hat{\mathcal{A}}$ that satisfies two conditions:

- **Coverage**: It includes the true active set, i.e., $\mathcal{A}^* \subseteq \hat{\mathcal{A}}$.
- **Sparsity**: Its size is significantly smaller than the total number of constraints, i.e., $|\hat{\mathcal{A}}| \ll m$.

Lemma 1 (Uniform Sampling over a Predicted Active Set)**.** *Under Assumptions 1 and 2, let $\omega_k \in \mathcal{U}$ and $\mathcal{A}^* \neq \varnothing$. For row-normalized constraints and $\beta_1, \beta_2 \geq 1$, uniform guided sampling over $\hat{\mathcal{A}}$ satisfies*

$$\mathbb{E}_k[d_{k+1}^2] \leq \left(1 - \frac{\alpha(2-\alpha)}{|\hat{\mathcal{A}}| H_*^2}\right) d_k^2. \tag{26}$$

Proof. For uniform guided sampling over a predicted set satisfying Assumption 2, take $p_i = |\hat{\mathcal{A}}|^{-1} \mathbf{1}_{\{i \in \hat{\mathcal{A}}\}}$. Coverage implies $p_* = 1/|\hat{\mathcal{A}}| \geq 1/m$, and (8) reduces to (26). □

Lemma 2 (Hoffman Bound under LICQ)**.** *If the transformed active submatrix $B = \tilde{A}_{\mathcal{A}^*}$ has full row rank, then*

$$\mathcal{H}(B)^{-2} \geq \sigma_{\min}^2(B). \tag{27}$$

Proof. Under LICQ, write $B = \tilde{A}_{\mathcal{A}^*}$ and $c = \tilde{b}_{\mathcal{A}^*}$. For any ω, set $s = [B\omega - c]_+$ and $z = \omega - B^\dagger s$. Full row rank gives $BB^\dagger = I$, so $Bz = B\omega - s \leq c$ and $z \in \Omega_{\mathcal{A}^*}$. Consequently,

$$\operatorname{dist}(\omega, \Omega_{\mathcal{A}^*}) \leq \|\omega - z\|_2 = \|B^\dagger s\|_2 \leq \frac{\|[B\omega - c]_+\|_2}{\sigma_{\min}(B)}. \tag{28}$$

The construction holds for every right-hand side c, giving $\mathcal{H}(B) \leq 1/\sigma_{\min}(B)$ for the least Hoffman constant [34]. Substituting this bound into (8) and (26) yields the corresponding singular-value bounds. In particular, uniform sampling over $\hat{\mathcal{A}}$ gives

$$\mathbb{E}_k[d_{k+1}^2] \leq \left(1 - \frac{\alpha(2-\alpha)\sigma_{\min}^2(\tilde{A}_{\mathcal{A}^*})}{|\hat{\mathcal{A}}|}\right) d_k^2. \tag{29}$$

□

B Computational Complexity of Proposed Cholesky Update Method

B.1 Implementation Details of the Cholesky Update

The key of our proposed Cholesky update framework is to utilize the properties of low-rank perturbations, allowing us to reuse updated Cholesky factors. This avoids the high computational cost of Singular Value Decomposition while efficiently maintaining both the projection operator and the null-space basis.

Let $\mathbf{G} = \mathbf{C}\mathbf{C}^\top$ be the original Gram matrix. The perturbed Gram matrix $\tilde{\mathbf{G}}$ is derived as:

$$\begin{aligned}\tilde{\mathbf{G}} &= (\mathbf{C} + \mathbf{u}\mathbf{v}^\top)(\mathbf{C} + \mathbf{u}\mathbf{v}^\top)^\top \\ &= \mathbf{C}\mathbf{C}^\top + \mathbf{C}\mathbf{v}\mathbf{u}^\top + \mathbf{u}\mathbf{v}^\top\mathbf{C}^\top + \mathbf{u}(\mathbf{v}^\top\mathbf{v})\mathbf{u}^\top \\ &= \mathbf{G} + (\mathbf{C}\mathbf{v})\mathbf{u}^\top + \mathbf{u}(\mathbf{C}\mathbf{v})^\top + \|\mathbf{v}\|^2\mathbf{u}\mathbf{u}^\top\end{aligned} \tag{30}$$

Let $\mathbf{h} = \mathbf{C}\mathbf{v}$ and scalar $\lambda = \|\mathbf{v}\|^2$, we rewrite the expression as:

$$\tilde{\mathbf{G}} = \mathbf{G} + \mathbf{h}\mathbf{u}^\top + \mathbf{u}\mathbf{h}^\top + \lambda\mathbf{u}\mathbf{u}^\top \tag{31}$$

We define an auxiliary vector $\mathbf{m} = \mathbf{h} + \frac{\lambda}{2}\mathbf{u}$, then we have:

$$\mathbf{h}\mathbf{u}^\top + \mathbf{u}\mathbf{h}^\top + \lambda\mathbf{u}\mathbf{u}^\top = \mathbf{u}\mathbf{m}^\top + \mathbf{m}\mathbf{u}^\top = \frac{1}{2}(\mathbf{u}+\mathbf{m})(\mathbf{u}+\mathbf{m})^\top - \frac{1}{2}(\mathbf{u}-\mathbf{m})(\mathbf{u}-\mathbf{m})^\top \tag{32}$$

Substituting this back into the Gram matrix expression yields:

$$\tilde{\mathbf{G}} = \mathbf{G} + \underbrace{\frac{1}{2}(\mathbf{u}+\mathbf{m})(\mathbf{u}+\mathbf{m})^\top}_{\text{Update}} - \underbrace{\frac{1}{2}(\mathbf{u}-\mathbf{m})(\mathbf{u}-\mathbf{m})^\top}_{\text{Downdate}} \tag{33}$$

Consequently, we do not need to explicitly reconstruct $\tilde{\mathbf{G}}$. Using standard `chol_update` and `chol_downdate` routines, we apply one rank-1 updates and one rank-1 downdate to the original factor $\mathbf{L}$, directly obtaining $\tilde{\mathbf{L}}$ such that $\tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top = \tilde{\mathbf{G}}$.

With the updated factor $\tilde{\mathbf{L}}$, we define the constraint residual $\mathbf{r} = \tilde{\mathbf{C}}\mathbf{x}_0 - \mathbf{d}$. The projection formula is given by:

$$\mathbf{x}_{\text{eq}} = \mathbf{x}_0 - \tilde{\mathbf{C}}^\top((\tilde{\mathbf{L}}^\top)^{-1}(\tilde{\mathbf{L}}^{-1}\mathbf{r})) \tag{34}$$

Using $\tilde{\mathbf{L}}$, we solve this via two triangular substitutions:

1. **Forward Substitution:** $\tilde{\mathbf{L}}\mathbf{k} = \mathbf{r}$;
2. **Backward Substitution:** $\tilde{\mathbf{L}}^\top\mathbf{y} = \mathbf{k}$.

The final projected solution is obtained by:

$$\tilde{\mathbf{x}}_{\text{eq}} = \mathbf{x}_0 - \tilde{\mathbf{C}}^\top\mathbf{y} \tag{35}$$

Next, we construct the new null-space basis $\tilde{\mathbf{Z}}$ such that $\tilde{\mathbf{C}}\tilde{\mathbf{Z}} = \mathbf{0}$, employing a rotation-correction strategy. First, we compute the projection of the perturbation onto the original null space, $\mathbf{w} = \mathbf{Z}^\top\mathbf{v}$. To concentrate the perturbation, we construct a Householder reflection $\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{q}\mathbf{q}^\top}{\|\mathbf{q}\|^2}$ with $\mathbf{q} = \mathbf{w} - \|\mathbf{w}\|\mathbf{e}_1$. Applying this to the original basis yields the transition basis [42]:

$$\mathbf{Z}' = \mathbf{Z}\mathbf{H} = \left[\mathbf{z}'_1, \mathbf{z}'_2, \ldots, \mathbf{z}'_{n-m}\right] \tag{36}$$

By properties of the Householder transformation, columns 2 through $n-m$ of $\mathbf{Z}'$ satisfy $\mathbf{v}^\top\mathbf{z}'_j = 0$. Thus, for $j \geq 2$:

$$\tilde{\mathbf{C}}\mathbf{z}'_j = (\mathbf{C} + \mathbf{u}\mathbf{v}^\top)\mathbf{z}'_j = \mathbf{C}\mathbf{z}'_j + \mathbf{u}(\mathbf{v}^\top\mathbf{z}'_j) = \mathbf{0} \tag{37}$$

This indicates that $\mathbf{Z}'_{:,2:\text{end}}$ directly forms the majority of the new null space. The first column $\mathbf{z}'_1$ produces a non-zero residual $\tilde{\mathbf{C}}\mathbf{z}'_1 = \mathbf{u}(\mathbf{v}^\top\mathbf{z}'_1)$. To eliminate this, we seek a vector $\mathbf{p}$ satisfying $\tilde{\mathbf{C}}\mathbf{p} = \mathbf{u}$. Using the maintained $\tilde{\mathbf{L}}$, we solve $\tilde{\mathbf{C}}\tilde{\mathbf{C}}^\top\mathbf{y} = \mathbf{u}$ via triangular substitution and set $\mathbf{p} = \tilde{\mathbf{C}}^\top\mathbf{y}$. Finally, adding the correction term to $\mathbf{z}'_1$ yields the complete new basis:

$$\tilde{\mathbf{Z}} = \left[\mathbf{z}'_1 - \mathbf{v}^\top\mathbf{z}'_1\mathbf{p}, \quad \mathbf{Z}'_{:,2:\text{end}}\right] \tag{38}$$

B.2 Complexity Analysis

Let $\mathbf{C} \in \mathbb{R}^{n_{\text{eq}} \times n}$ denote the row full-rank equality-constraint matrix, and let $r = n - n_{\text{eq}}$ be the dimension of its null space. The SVD approach requires a complete factorization of the constraint matrix, incurring a time complexity of $\mathcal{O}(n^3)$. In contrast, the proposed framework exploits the low-rank nature of topological perturbations, effectively decoupling the high-dimensional problem into sequential vector-level operations. The computational cost of one complete iteration is decomposed as follows:

- **Factor Maintenance:** We perform one rank-1 update and one rank-1 downdate on the Cholesky factor $\mathbf{L}$ of $\mathbf{G} = \mathbf{C}\mathbf{C}^\top \in \mathbb{R}^{n_{\text{eq}} \times n_{\text{eq}}}$. This step operates purely on the triangular factor, reducing the complexity to $\mathcal{O}(n_{\text{eq}}^2)$, where n_{eq} is the number of equality constraints.
- **Equality Projection:** Computing the projected solution involves matrix-vector multiplications and triangular substitutions using the updated $\tilde{\mathbf{L}}$. The dominant cost is the matrix-vector product, scaling as $\mathcal{O}(n_{\text{eq}} \cdot n)$.
- **Null Space Update:** The subspace rotation involves Householder transformations on the basis matrix $\mathbf{Z} \in \mathbb{R}^{n \times r}$, costing $\mathcal{O}(n \cdot r)$, and the correction requires solving for vector $\mathbf{p}$, adding a $\mathcal{O}(n_{\text{eq}} \cdot n)$ cost regardless of the null-space dimension.

Since both n_{eq} and r are bounded by n, the overall asymptotic complexity of our framework is $\mathcal{O}(n^2)$. This represents a theoretical speedup of order n compared to standard SVD-based methods, transforming the expensive topological adaptation into a computationally lightweight process which is suitable for real-time applications.

B.3 Detailed Analysis for Results on RGGs

To validate the theoretical complexity analysis, we performed tests on Random Geometric Graphs of varying scales. Table 2 presents the overall performance comparison.

Table 2: Computation time comparison between SVD and Cholesky Update on random geometric graphs. The reported complexity values are empirical scaling exponents obtained by least-squares linear regression in log–log space.

Small-Scale ($N = 100 \sim 1000$)					**Large-Scale** ($N = 1500 \sim 5000$)				
N	K	SVD (ms)	Chol. (ms)	Speedup	N	K	SVD (ms)	Chol. (ms)	Speedup
100	10	0.874	0.119	7.36×	1500	38	307.441	8.187	37.55×
200	14	2.906	0.290	10.02×	2000	44	995.480	14.754	67.47×
400	20	11.106	0.575	19.33×	2500	50	2276.658	23.450	97.09×
800	28	49.833	2.257	22.08×	3000	54	4595.625	40.505	113.46×
1000	31	83.666	3.583	23.35×	5000	70	20596.582	122.586	168.02×
Complex.	-	$\mathcal{O}(n^{1.99})$	$\mathcal{O}(n^{1.46})$	-	**Complex.**	-	$\mathcal{O}(n^{3.49})$	$\mathcal{O}(n^{2.28})$	-

Table 2 highlights the significant scalability gap between the SVD approach and our proposed Cholesky Update method. In the small-scale group from 100 to 1000, we observe empirical complexities of $\mathcal{O}(n^{1.99})$ for SVD and $\mathcal{O}(n^{1.46})$ for the Cholesky Update. These growth rates are much lower than their theoretical asymptotic bounds, which can be attributed to specific optimizations in underlying linear algebra libraries that maximize cache efficiency and instruction throughput for smaller matrices. However, even with these library-level enhancements benefiting the baseline, our method maintains superior performance, delivering substantial speedups ranging from 7.36× to 23.35×.

As the number of nodes increases from 1500 to 5000, the SVD computation time exhibits a super-cubic growth trend with a fitted exponent of 3.49 in the large-scale group, reflecting the heavy burden of computation and memory data movement. In contrast, the proposed Cholesky Update maintains a near-quadratic growth rate of approximately 2.28, which closely matches the theoretical lower bound. A remaining limitation is that large-scale matrix operations can become memory-bound, as the increasing matrix size and data movement overhead can partially offset the arithmetic savings and push the observed scaling above the ideal quadratic rate. Nevertheless, the method still maintains a clear efficiency advantage over SVD, achieving over 168 times acceleration at 5000 nodes.

B.4 Numerical Stability under Higher-Rank Contingencies

Sequential Cholesky updates may accumulate numerical errors under finite precision, especially when multiple topology changes are applied consecutively. This issue becomes more visible for higher-rank contingencies and larger systems, where the update and downdate vectors can contain entries with very different magnitudes. To address this numerical concern, we use an adaptive rescaling strategy that preserves the exact low-rank perturbation while improving the conditioning of intermediate update vectors.

For a rank-one perturbation $\Delta\mathbf{C} = \mathbf{u}\mathbf{v}^\top$, the representation is not unique. For any scalar $\alpha > 0$, we have

$$\Delta\mathbf{C} = \mathbf{u}\mathbf{v}^\top = (\alpha\mathbf{u})(\mathbf{v}/\alpha)^\top. \tag{39}$$

In the Gram-matrix update derivation in Appendix B.1, we define $\mathbf{h} = \mathbf{C}\mathbf{v}$, $\lambda = \|\mathbf{v}\|_2^2$, and $\mathbf{m} = \mathbf{h} + \frac{\lambda}{2}\mathbf{u}$. After applying the above rescaling, the corresponding pair becomes $\alpha\mathbf{u}$ and $\mathbf{m}/\alpha$. Therefore, the exact perturbation remains unchanged, but the numerical magnitudes of the two vectors entering the Cholesky update and downdate can be balanced. In practice, we choose

$$\alpha = \sqrt{\frac{\|\mathbf{m}\|_2}{\|\mathbf{u}\|_2}}, \tag{40}$$

which equalizes the norms of $\alpha\mathbf{u}$ and $\mathbf{m}/\alpha$. This adaptive rescaling reduces extreme intermediate values, for example, on the IEEE 300-bus system, the largest intermediate magnitude is reduced from roughly 10^{10} to roughly 10^5.

We evaluate both the original and adaptive Cholesky updates under fp32 and fp64 precision. The experiments cover all N-1 and N-2 contingencies, together with a subset of N-3 contingencies, on the IEEE 57-, 118-, and 300-bus systems. A scenario is counted as a failure if the sequential Cholesky update cannot be completed in the corresponding floating-point precision.

Table 3: Failure counts of original and adaptive Cholesky updates under fp32 and fp64 precision. The percentage in parentheses denotes the failure rate among evaluated contingency scenarios.

System	Contingency	# Scenarios	FP32 Failures		FP64 Failures	
			Original	Adaptive	Original	Adaptive
57-bus	N-1	79	68 (86.1%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
57-bus	N-2	3024	2973 (98.3%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
57-bus	N-3	6048	-	0 (0.0%)	-	0 (0.0%)
118-bus	N-1	177	174 (98.3%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
118-bus	N-2	15502	15501 (99.9%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
118-bus	N-3	31004	-	0 (0.0%)	-	0 (0.0%)
300-bus	N-1	322	303 (94.1%)	1 (0.3%)	2 (0.6%)	0 (0.0%)
300-bus	N-2	51559	51405 (99.7%)	279 (0.5%)	641 (1.2%)	0 (0.0%)
300-bus	N-3	103118	-	558 (0.5%)	-	0 (0.0%)

As shown in Table 3, the original sequential update is highly unstable in fp32 and becomes almost unusable for N-1 and N-2 contingencies. Although fp64 substantially improves stability, the original update still produces a small number of failures on the IEEE 300-bus system. In contrast, the adaptive rescaling removes all observed fp64 failures across all evaluated systems and contingency levels. Under fp32, it also eliminates all failures on the 57- and 118-bus systems and leaves only a small failure rate on the larger 300-bus system. These results indicate that the numerical stability issue of sequential Cholesky updates can be effectively controlled by adaptive rescaling, while the exact low-rank perturbation and the theoretical update structure remain unchanged.

C Minimum-Cost Gas Transport Problem

We further evaluate the proposed method on a minimum-cost gas transport problem using the GasLib-135 benchmark [43]. Given a gas network $G = (V, E)$, each node $v \in V$ represents a supply or demand point, and each edge $e \in E$ is associated with capacity limits and transportation costs. The goal is to find a minimum-cost flow allocation while satisfying nodal flow-balance constraints and edge-capacity constraints. Each test instance corresponds to an operating scenario with different

demand levels and possible topology changes, such as edge removals or failure cases. We evaluate both seen and unseen scenarios to assess generalization across network conditions. The optimization problem is formulated as follows:

$$\begin{aligned}
\min_{\mathbf{g}^{(k)},\,\mathbf{f}^{(k)}} \quad & \sum_{s\in\mathcal{S}} \alpha_s g_s^{(k)} + \sum_{\ell\in\mathcal{L}^{(k)}} \rho_\ell \left| f_\ell^{(k)} \right| \\
\text{s.t.} \quad & \sum_{s\in\mathcal{S}(v)} g_s^{(k)} + \sum_{\ell\in\mathcal{L}^{(k)}} A_{v\ell}^{(k)} f_\ell^{(k)} = d_v, \quad \forall v \in \mathcal{V}\setminus\{v_{\mathrm{ref}}\}, \\
& \sum_{s\in\mathcal{S}} g_s^{(k)} = \sum_{r\in\mathcal{R}} d_r, \\
& 0 \le g_s^{(k)} \le \overline{g}_s, \quad \forall s \in \mathcal{S}, \\
& -\overline{f}_\ell \le f_\ell^{(k)} \le \overline{f}_\ell, \quad \forall \ell \in \mathcal{L}^{(k)}, \\
& k \in \{0\} \cup \mathcal{K}.
\end{aligned} \tag{41}$$

where

$$\mathcal{L}^{(0)} = \mathcal{L}, \qquad \mathcal{L}^{(k)} = \mathcal{L}\setminus\{k\}, \quad k \in \mathcal{K}.$$

Table 4: HGNN-only prediction quality on the minimum-cost gas transport problem. Metrics except forward time are formatted as "Seen / Unseen".

Method	Forward Time (ms)	Max Eq. Viol.	Max Ineq. Viol.	Opt. Gap (%)
HGNN	3.810 ± 2.074	473.6/2337.0	100.5/416.8	0.008/0.009

Table 5: SKM-layer performance on the minimum-cost gas transport problem. The optimality gap is reported as mean value in the format "Seen / Unseen".

Method	Eq. Proj.	Eq. Spd.	Ineq. Iter.	Ineq. Time	Ineq. Spd.	SKM Time	SKM Spd.	Opt. Gap (%)
T-SKM	2.19 ± 1.06	$1.00\times$	15.21 (128)	0.96 ± 1.22	$1.00\times$	3.15	$1.00\times$	0.006/0.019
AT-SKM	0.53 ± 0.35	$4.13\times$	5.33 (78)	0.37 ± 0.71	$2.59\times$	0.90	$3.50\times$	0.006/0.019

Table 4 shows that the HGNN-only predictor can obtain small mean optimality gaps, but it suffers from substantial worst-case equality and inequality constraint violations, especially on unseen scenarios. This indicates that direct neural prediction alone is insufficient for safety-critical constrained optimization.

In contrast, Table 5 shows that both T-SKM-Net and AT-SKM-Net achieve zero equality and inequality constraint violations on both seen and unseen scenarios. Compared with T-SKM-Net, AT-SKM-Net reduces the equality projection time from 2.19 ms to 0.53 ms, yielding a $4.13\times$ speedup, and reduces the inequality iteration time from 0.96 ms to 0.37 ms, yielding a $2.59\times$ speedup. Overall, AT-SKM-Net reduces the total SKM computation time from 3.15 ms to 0.90 ms, corresponding to a $3.50\times$ acceleration while preserving strict feasibility and maintaining comparable mean optimality gaps.

D Experimental Settings and Additional Results for N-1 Security-Constrained DC-OPF

D.1 Mathematical Formulation of N-1 Security-Constrained DC-OPF

The N-1 Security-Constrained DC Optimal Power Flow problem aims to determine the optimal generation dispatch that minimizes total operating costs while ensuring system feasibility under both normal conditions and any single transmission line outage. Let $\mathcal{K}$ denote the set of all considered scenarios, where $k = 0$ represents the base case and each remaining $k \in \mathcal{K}$ corresponds to a specific line failure contingency. Under a given scenario k, the network topology changes accordingly; we define $\mathcal{L}^{(k)}$ as the set of remaining active lines and $\mathcal{N}^{(k)}(i)$ as the set of neighboring buses for bus i in that topology. The optimization problem is formulated as follows:

$$
\begin{aligned}
\min_{\mathbf{P}_{\mathcal{G}},\boldsymbol{\delta}} \quad & \sum_{i\in\mathcal{G}} \left(\frac{1}{2}c_i P_{G,i}^2 + b_i P_{G,i}\right) \\
\text{s.t.} \quad & P_{G,i} - P_{D,i} = \sum_{j\in\mathcal{N}^{(k)}(i)} B_{ij}(\delta_i^{(k)} - \delta_j^{(k)}), \quad \forall i \in \mathcal{B}, \forall k \in \{0\}\cup\mathcal{K} \\
& P_{G,i}^{\min} \le P_{G,i} \le P_{G,i}^{\max}, \quad \forall i \in \mathcal{G} \\
& -P_{ij}^{\max} \le B_{ij}(\delta_i^{(k)} - \delta_j^{(k)}) \le P_{ij}^{\max}, \quad \forall (i,j) \in \mathcal{L}^{(k)}, \forall k \in \{0\}\cup\mathcal{K}
\end{aligned}
\tag{42}
$$

where the sets $\mathcal{B}$, $\mathcal{G} \subseteq \mathcal{B}$, and $\mathcal{L}$ represent buses, generators, and transmission lines, respectively. The decision variables consist of the active power outputs $\mathbf{P}_{\mathcal{G}} = \{P_{G,i}\}_{i\in\mathcal{G}}$ and the voltage phase angles $\boldsymbol{\delta}^{(k)} = \{\delta_i^{(k)}\}_{i\in\mathcal{B}}$ for each scenario. It is important to note that the generation dispatch P_G is preventive and shared across all scenarios, whereas the system state $\delta^{(k)}$ adapts to each specific topology. In the objective function, c_i and b_i denote the quadratic and linear cost coefficients for generator i. The parameter $P_{D,i}$ represents the active power demand at bus i, and B_{ij} is the line susceptance defined as a positive value $1/X_{ij}$. For non-generator buses $i \in \mathcal{B} \setminus \mathcal{G}$, we set $P_{G,i} = 0$. The terms $P_{G,i}^{\min\,/\,\max}$ and $P_{ij}^{\max}$ define the generator capacity limits and transmission line thermal limits, respectively.

D.2 N-1 Security-Constrainted DC-OPF Dataset Generation

To validate the effectiveness of the proposed method, we employ the IEEE 57, 118, and 300-bus systems as experimental benchmarks. In constructing the N-1 contingency dataset, we first screen all potential single-line outage scenarios, retaining only those that are physically feasible. Here, a feasible scenario is defined as one where the post-contingency network topology remains connected without forming islands, and the OPF solver successfully converges to a feasible solution. Table 6 summarizes the statistics for each test system, including the number of buses, total branches, and the count of valid N-1 contingencies with the original scenario after screening.

For each selected topological scenario, we generate 100 independent operating instances by introducing random perturbations to the nodal loads. Specifically, the load vectors are obtained by uniformly scaling the baseline load profile:

$$
P_D^{(i)} = \alpha_i \cdot P_D^{\text{base}}, \quad \alpha_i \sim \mathcal{U}(0.8, 1.2), \tag{43}
$$

where P_D^{base} represents the baseline load values, and α_i is a random scaling factor drawn from a uniform distribution over the interval $[0.8, 1.2]$. This generation procedure ensures that the dataset encompasses a wide range of load fluctuations.

Regarding dataset partitioning, to evaluate the model's generalization capability to unseen contingencies, we randomly divide all valid N-1 scenarios into two subsets: 70% are designated as *seen* scenarios for the training phase, while the remaining 30% serve as *unseen* scenarios. The training set consists exclusively of samples generated from the seen scenarios, whereas the test set comprehensively covers instances from both seen and unseen topologies.

Table 6: Settings of N-1 contingency Dataset and Hyperparameters of SKM Sampling

Test System	Buses	Branches	Removable	Scenarios	β	ρ
IEEE 57-Bus	57	80	79	80	10	0.8
IEEE 118-Bus	118	186	177	178	50	0.8
IEEE 300-Bus	300	411	322	323	60	0.8

D.3 Experimental Hyperparameters and Loss Function Configuration

In terms of model architecture and training configuration, we employ a Heterogeneous Graph Neural Network (HGNN) consisting of two message-passing layers, with the hidden dimension uniformly set to 128. We utilize the Adam optimizer for training, initialized with a learning rate of $5e{-}4$. A StepLR scheduler is applied to implement stepwise decay:

$$
\eta_t = \eta_0 \cdot \gamma^{\lfloor t/S \rfloor}, \tag{44}
$$

where η_0 denotes the initial learning rate, $S = 25$ represents the step size, and $\gamma = 0.8$ is the decay factor. The batch size is set to 64 during the training phase. Conversely, the testing phase utilizes a batch size of 1 to simulate real-world scenarios requiring real-time inference for individual samples.

The composite loss function comprises two components: regression loss and classification loss. Following the methodology of T-SKM [15], the regression loss adopts the Mean Squared Error (MSE) formulation, calculated as a weighted sum of the pre-projection and post-projection losses:

$$L_{reg} = 0.9 \cdot \text{MSE}(\hat{\mathbf{y}}_{pre}, \mathbf{y}) + 0.1 \cdot \text{MSE}(\hat{\mathbf{y}}_{proj}, \mathbf{y}), \tag{45}$$

where $\hat{\mathbf{y}}_{pre}$ represents the raw model output and $\hat{\mathbf{y}}_{proj}$ denotes the output after constraint projection. This combination enables the model to learn the primary feature mapping while simultaneously encouraging satisfaction of physical constraints.

For the active-set prediction task, we employ the Binary Focal Loss to address the issue of positive-negative sample imbalance [44]:

$$L_{\text{focal}}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t), \tag{46}$$

where p_t is the model's estimated probability for the ground-truth class. The focusing parameter $\gamma = 2.0$ down-weights easy-to-classify examples, while the balancing parameter $\alpha = 0.5$ adjusts the relative importance of positive and negative samples. The total loss function is expressed as:

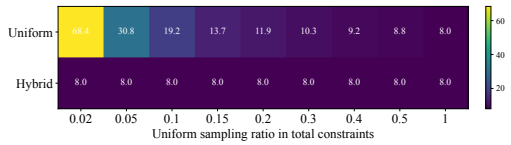
$$L_{\text{total}} = L_{\text{reg}} + \lambda(t) \cdot L_{\text{cls}}. \tag{47}$$

Here, $\lambda(t)$ follows a linear warmup schedule, increasing linearly from 0 to 1 over the first 40 training epochs. This strategy allows the model to prioritize the regression task during the initial training phase before gradually integrating the classification objective.

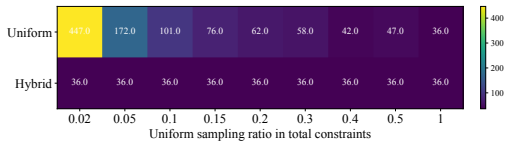
Active-set labels are generated using a relative slack criterion. For a two-sided inequality $l_i \leq g_i(x) \leq u_i$, the constraint is labeled active if its slack to either bound is less than 1% of the feasible interval $u_i - l_i$. This threshold is used only for training the active-set predictor. In each iteration, the number of sampled constraints corresponds to approximately 10% of the total constraint set. The specific configurations for the sampling batch size β and the hybrid sampling ratio ρ across different test systems are detailed in Table 6.

D.4 Hyperparameter Sensitivity of Hybrid Sampling Strategy

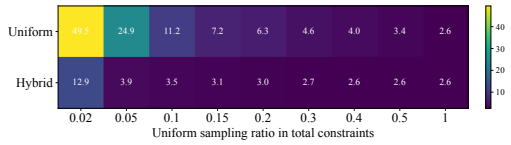
To analyze the sensitivity of our proposed active-set predict method to the batch size parameter, we conducted a controlled experiment using ground truth data. In this setup, we assume the availability of a perfect predictor by directly utilizing the actual active sets derived from the optimal solutions to guide the sampling process. We compared the convergence performance of the standard uniform sampling against the ground truth guided hybrid sampling across varying sampling ratios on IEEE 118 bus systems.



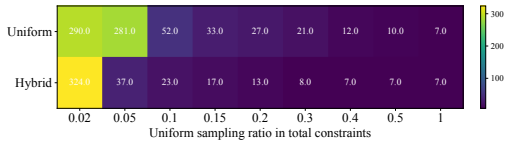
(a) IEEE 118 Mean Iterations



(b) IEEE 118 Max Iterations



(c) IEEE 300 Mean Iterations



(d) IEEE 300 Max Iterations

Figure 3: Sensitivity analysis of sampling strategies with a perfect predictor on IEEE 118-bus and 300-bus systems. Comparison between uniform sampling and ground truth guided hybrid sampling across varying sampling ratios.

Figure 3 presents the sensitivity analysis where the SKM iterative process is initialized using the solution predicted by the HGNN.

For the IEEE 118 bus system shown in Figure 3a and Figure 3b, the results demonstrate that the hybrid sampling strategy exhibits remarkable stability when the active set is accurately identified. Regardless of the proportion of additional random sampling, both the mean and maximum iteration counts remain at a consistently low level. This implies that once the true active constraints are covered, the algorithm is effectively decoupled from the interference of massive inactive constraints.

In the larger IEEE 300 bus system shown in Figure 3c and Figure 3d, the hybrid strategy shows a significantly sharper improvement in convergence speed compared to uniform sampling. With only a minimal amount of random sampling included, the hybrid approach allows the iteration count to drop rapidly to its optimal level. Specifically, Figure 3d reveals that the hybrid strategy effectively suppresses worst case scenarios, achieving performance with about a tenth of the sampling effort that would otherwise require full set scanning in the uniform approach.

D.5 Detailed Time Breakdown for AT-SKM-Net on SC DC-OPF Problems

Using the same models and experimental settings as in the main SC DC-OPF experiments, we provide a detailed runtime breakdown of T-SKM-Net and AT-SKM-Net. Specifically, we separately measure the time spent in the equality projection stage and the inequality SKM iteration stage, which correspond to the two main computational components accelerated by the proposed Cholesky update and active-set-guided sampling strategy, respectively.

Table 7: Detailed runtime breakdown of T-SKM-Net and AT-SKM-Net on SC DC-OPF problems. AT-SKM-AC denotes the full AT-SKM-Net with both active-set-guided sampling and Cholesky update. Runtime values are reported in milliseconds.

System	Method	Eq. Proj. Time	Eq. Proj. Spd.	Ineq. Iter. Time	Ineq. Iter. Spd.	Total SKM Spd.
57-bus	T-SKM-Net	0.56 ± 0.34	$1.00\times$	0.41 ± 1.20	$1.00\times$	$1.00\times$
57-bus	AT-SKM-AC	0.18 ± 0.02	$3.11\times$	0.06 ± 0.07	$6.83\times$	$4.04\times$
118-bus	T-SKM-Net	1.56 ± 0.06	$1.00\times$	2.13 ± 2.51	$1.00\times$	$1.00\times$
118-bus	AT-SKM-AC	0.30 ± 0.03	$5.20\times$	1.01 ± 1.38	$2.11\times$	$2.81\times$
300-bus	T-SKM-Net	7.65 ± 1.54	$1.00\times$	2.07 ± 4.59	$1.00\times$	$1.00\times$
300-bus	AT-SKM-AC	0.70 ± 0.02	$10.92\times$	0.32 ± 0.20	$6.46\times$	$9.53\times$

As shown in Table 7, AT-SKM-Net substantially reduces the equality projection time compared with T-SKM-Net. Moreover, the acceleration becomes more pronounced as the network size increases: the equality projection speedup grows from $3.11\times$ on the 57-bus system to $10.92\times$ on the 300-bus system. This trend is consistent with the theoretical motivation of the Cholesky update, whose advantage becomes more significant when the equality projection involves larger constraint matrices.

The inequality SKM iteration stage is also accelerated. Compared with T-SKM-Net, AT-SKM-Net reduces the inequality iteration time by $6.83\times$, $2.11\times$, and $6.46\times$ on the 57-, 118-, and 300-bus systems, respectively. These reductions reflect the benefit of active-set-guided sampling, which avoids spending excessive iterations on inactive or redundant constraints. Overall, AT-SKM-Net achieves total SKM speedups of $4.04\times$, $2.81\times$, and $9.53\times$ across the three systems, which is consistent with the main-paper conclusion that the proposed framework accelerates both equality projection and inequality iteration while preserving strict feasibility.

D.6 Robustness under High-Load Operating Conditions

To further evaluate the robustness of proposed AT-SKM-Net, we construct an additional high-load test regime for the SC DC-OPF problem. In the main experiments, load perturbations are sampled from the range $[0.8, 1.2]$ around the nominal demand profile. Here, we shift the load range to $[1.0, 1.2]$, thereby concentrating the test instances in more heavily loaded operating conditions. Such cases are more challenging because line-flow and generation-capacity constraints are more likely to approach their limits, making the active-set pattern more sensitive to both load variations and topology changes.

As shown in Table 8, AT-SKM-Net continues to achieve clear acceleration over T-SKM-Net under the high-load regime. The equality projection stage is consistently accelerated, with the speedup increasing from $2.33\times$ on the 57-bus system to $9.93\times$ on the 300-bus system. This again confirms that the Cholesky update becomes increasingly beneficial as the system size grows.

Table 8: Performance of T-SKM-Net and AT-SKM-Net under high-load SC DC-OPF conditions. Load perturbations are sampled from $[1.0, 1.2]$. Runtime values are reported in milliseconds. The inequality iteration column reports the mean number of iterations, with the maximum shown in parentheses.

System	Method	Eq. Proj. Time	Eq. Spd.	Ineq. Iter.	Ineq. Time	Ineq. Spd.	SKM Spd.	Mean Gap (%)
57-bus	T-SKM-Net	0.42 ± 0.14	$1.00\times$	$16.00\,(133)$	0.70 ± 1.22	$1.00\times$	$1.00\times$	0.014
57-bus	AT-SKM-AC	0.18 ± 0.04	$2.33\times$	$2.75\,(12)$	0.13 ± 0.14	$5.38\times$	$3.61\times$	0.010
118-bus	T-SKM-Net	1.81 ± 0.59	$1.00\times$	$18.59\,(160)$	1.45 ± 2.80	$1.00\times$	$1.00\times$	0.004
118-bus	AT-SKM-AC	0.37 ± 0.22	$4.89\times$	$9.21\,(56)$	0.82 ± 1.56	$1.77\times$	$2.74\times$	0.004
300-bus	T-SKM-Net	7.05 ± 0.51	$1.00\times$	$25.49\,(248)$	1.78 ± 3.02	$1.00\times$	$1.00\times$	0.136
300-bus	AT-SKM-AC	0.71 ± 0.02	$9.93\times$	$5.34\,(28)$	0.42 ± 0.58	$4.24\times$	$7.81\times$	0.136

The inequality iteration stage also remains robust under heavier loading. Compared with T-SKM-Net, AT-SKM-Net reduces the mean number of inequality iterations from 16.00 to 2.75 on the 57-bus system, from 18.59 to 9.21 on the 118-bus system, and from 25.49 to 5.34 on the 300-bus system. The maximum iteration count is also substantially reduced across all systems, indicating that active-set-guided sampling mitigates difficult tail cases even when more constraints are close to becoming active.

These results suggest that the HGNN predictor remains effective in more stressed operating conditions. The warm-start prediction continues to provide useful initialization, while the active-set predictor still concentrates sampling on the most relevant constraints. As a result, AT-SKM-Net achieves total SKM speedups of $3.61\times$, $2.74\times$, and $7.81\times$ on the 57-, 118-, and 300-bus systems, respectively, while maintaining comparable mean optimality gaps to T-SKM-Net.

D.7 Active-Set Prediction Quality

Table 9: Active set prediction performance on test "seen / unseen" scenarios.

System	Accuracy	Precision	Recall
IEEE 57-Bus	0.9995 / 0.9995	0.9814 / 0.9802	0.9804 / 0.9815
IEEE 118-Bus	0.9995 / 0.9995	0.9988 / 0.9989	0.9948 / 0.9940
IEEE 300-Bus	0.9991 / 0.9991	0.9647 / 0.9649	0.9834 / 0.9834

To validate that the theoretical acceleration potential established in the sensitivity analysis is achievable in practice, we evaluate the actual classification performance of the proposed HGNN predictor.

Table 9 details the Accuracy, Precision, and Recall metrics across three test systems. We explicitly distinguish between "seen" and "unseen" scenarios to assess the model's generalization capability.

Given that active constraints constitute only a tiny fraction of the total set yet strictly define the optimal solution, missing even a single binding constraint can severely stall convergence. Consequently, the recall rate is the most important metric. Table 9 shows that the HGNN consistently achieves a Recall exceeding 98% across all systems. This high coverage guarantees that the predictor captures nearly all binding constraints required to define the active manifold, allowing SKM to focus on the feasible boundary and minimizing the reliance on random exploration to recover missed information. These recall values provide empirical support for the coverage condition in Assumption 2, namely $\mathcal{A}^* \subseteq \hat{\mathcal{A}}$, up to the small prediction error observed in finite test samples.

In addition to reliability, the method maintains consistently high precision from 96.5% to 99.9%, ensuring that the predicted candidate sets are compact with minimal redundancy. Such compactness is essential for efficiency as it prevents the dilution of sampling probability. Moreover, the framework demonstrates that the HGNN remains robust to topological changes. The negligible performance gap between seen and unseen scenarios indicates that the HGNN has captured the underlying physical laws governing constraint activation, rather than overfitting to specific patterns. This robustness ensures that AT-SKM sustains its acceleration performance even under unexpected line failures.

D.8 Robustness to Active-Set Perturbations

We further analyze the robustness of the proposed active-set-guided sampling strategy when the predicted active set is imperfect. In practice, the prediction errors of an active-set predictor can be divided into two types: false negatives, where truly active constraints are dropped from the predicted pool, and false positives, where inactive constraints are added to the predicted pool. Both types of errors weaken the sampling guidance, but their effects are not symmetric.

To isolate these two effects, we conduct a two-dimensional perturbation experiment using the oracle active set. Starting from the true active set, we randomly drop a prescribed fraction of true active constraints and add a prescribed number of inactive constraints. Both the add and drop ratios are normalized by the size of the true active set. We then run the SKM layer with the perturbed active-set pool and report the mean number of inequality iterations.

Table 10: Mean number of SKM inequality iterations under active-set perturbations on the IEEE 118-bus system. Rows denote the ratio of added inactive constraints, and columns denote the ratio of dropped true active constraints. Both ratios are normalized by the true active-set size.

ADD RATIO	DROP RATIO						
	0.00	0.05	0.10	0.15	0.20	0.25	0.50
0.0	15.68	18.04	20.02	22.71	24.41	24.71	29.61
0.5	15.85	20.64	29.60	33.44	39.12	39.71	42.53
1.0	16.37	21.41	29.04	35.27	37.91	42.79	64.02
1.5	17.04	21.34	27.89	35.81	39.81	41.90	63.72
2.0	17.64	22.10	30.32	36.28	41.06	42.42	63.08
3.0	19.12	24.32	31.16	36.93	41.80	44.11	65.27

Table 11: Mean number of SKM inequality iterations under active-set perturbations on the IEEE 300-bus system. Rows denote the ratio of added inactive constraints, and columns denote the ratio of dropped true active constraints.

ADD RATIO	DROP RATIO						
	0.00	0.05	0.10	0.15	0.20	0.25	0.50
0.0	3.69	4.15	7.68	8.88	10.99	13.96	19.24
0.5	3.69	4.51	9.65	10.39	14.43	16.55	23.97
1.0	3.72	4.78	11.53	12.50	17.17	23.93	33.14
1.5	3.77	5.25	12.03	13.27	20.15	24.80	39.51
2.0	3.80	4.79	13.36	15.24	23.41	28.90	43.07
3.0	4.04	5.79	17.90	21.70	34.99	40.32	69.75

Tables 10 and 11 show that both false positives and false negatives degrade the efficiency of active-set-guided sampling. However, dropping true active constraints is generally more harmful than adding inactive constraints. On the IEEE 118-bus system, when no inactive constraints are added, increasing the drop ratio from 0 to 0.5 increases the mean iteration count from 15.68 to 29.61. In contrast, when no true active constraints are dropped, increasing the add ratio from 0 to 3.0 only increases the mean iteration count from 15.68 to 19.12. A similar trend appears on the IEEE 300-bus system, where the iteration count increases from 3.69 to 19.24 as the drop ratio grows from 0 to 0.5, but only from 3.69 to 4.04 as the add ratio grows from 0 to 3.0 without dropping true active constraints. For reference, the original T-SKM-Net requires 34.60 inequality iterations on the IEEE 118-bus system and 24.72 iterations on the IEEE 300-bus system.

This behavior is consistent with the SKM update rule. In each iteration, SKM samples a subset of constraints and projects onto the most violated constraint within that sampled subset. Therefore, inactive false-positive constraints do not necessarily cause harmful projections; their main effect is to dilute the probability of sampling truly active constraints. In contrast, false negatives remove truly active constraints from the guided sampling pool, forcing them to be recovered only through the uniform sampling branch.

Let m denote the total number of constraints, s the size of the predicted active-set pool, and β_g and β_u the guided and uniform sample sizes, respectively. For a true active constraint that remains in the

predicted pool, its probability of being sampled is

$$p_{\text{keep}} = \frac{\beta_g}{s} + \left(1 - \frac{\beta_g}{s}\right) \frac{\beta_u}{m - \beta_g} > \frac{\beta_g}{s}. \tag{48}$$

If this active constraint is mistakenly dropped from the predicted pool, it can only be selected by the uniform branch:

$$p_{\text{drop}} = \frac{\beta_u}{m - \beta_g}. \tag{49}$$

Thus, adding Δ inactive constraints mainly dilutes the guided sampling probability from β_g/s to $\beta_g/(s + \Delta)$. By contrast, dropping a true active constraint moves it from the guided branch to the uniform branch, reducing its sampling probability from approximately β_g/s to $\beta_u/(m - \beta_g)$. Since in our setting the predicted active-set pool is much smaller than the full constraint set and the guided sampling ratio satisfies $\rho > 1/2$, we have

$$\frac{\beta_u/(m - \beta_g)}{\beta_g/(s + \Delta)} = \frac{1 - \rho}{\rho} \cdot \frac{s + \Delta}{m - \beta_g} \ll 1. \tag{50}$$

This explains why dropping true active constraints is more damaging than adding inactive constraints. At the same time, the perturbation results show that AT-SKM-Net remains robust under mild active-set prediction errors, and that the guided sampling strategy continues to provide acceleration even when the predicted pool contains moderate false positives.

D.9 Comparison with additional SKM variants

We compare Greedy KM, randomized Block SKM, heavy-ball SKM (HB-SKM), and Nesterov-SKM on the IEEE 57-, 118-, and 300-bus systems. For the latter three methods, we report results both without active-set prediction (ASP) and with the same active-set prediction mechanism. The Block SKM block sizes are 4, 8, and 16 for the 57-, 118-, and 300-bus systems, respectively. The momentum coefficient is 0.5 for both HB-SKM and Nesterov-SKM.

Table 12: Comparison with additional SKM variants. Times refer to the inequality iteration stage. ASP speedup is the ratio of mean runtime without ASP to that with ASP for the same method.

Method	IEEE 57-Bus		IEEE 118-Bus		IEEE 300-Bus	
	Time (ms)	Speedup	Time (ms)	Speedup	Time (ms)	Speedup
Greedy KM	0.09 ± 0.02	–	6.77 ± 0.88	–	0.46 ± 0.07	–
Block SKM w/o ASP	1.15 ± 0.21	–	16.46 ± 2.32	–	6.75 ± 0.94	–
Block SKM w/ ASP	0.08 ± 0.02	$14.38\times$	5.83 ± 0.84	$2.82\times$	0.25 ± 0.04	$27.00\times$
HB-SKM w/o ASP	0.37 ± 0.08	–	2.83 ± 0.39	–	2.23 ± 0.26	–
HB-SKM w/ ASP	0.08 ± 0.02	$4.63\times$	1.76 ± 0.35	$1.61\times$	0.39 ± 0.06	$5.72\times$
Nesterov-SKM w/o ASP	0.34 ± 0.07	–	2.72 ± 0.35	–	2.21 ± 0.33	–
Nesterov-SKM w/ ASP	0.06 ± 0.01	$5.67\times$	1.66 ± 0.24	$1.64\times$	0.32 ± 0.05	$6.91\times$

Table 12 shows that active-set prediction reduces the inequality iteration runtime across all tested SKM variants and systems. On the IEEE 118- and 300-bus systems, all three ASP-enhanced variants are faster than Greedy KM. On the IEEE 300-bus system, ASP accelerates Block SKM, HB-SKM, and Nesterov-SKM by $27.00\times$, $5.72\times$, and $6.91\times$ relative to their corresponding versions without ASP.

These results support the modularity of AT-SKM-Net. The active-set prediction head guides the construction of the candidate constraint set, while Block SKM, HB-SKM, and Nesterov-SKM apply their respective block projections or momentum updates after the candidate constraints have been sampled.

D.10 Transfer Learning and Pseudo-Label Active-Set Training

Although the experiments in the main paper use supervised training with solver-generated reference solutions, AT-SKM-Net does not fundamentally require exact optimal solutions for all training

instances. The framework relies on two acceleration components: a warm-start candidate for equality projection and SKM iteration, and active-set scores for guided inequality sampling. These components can be trained or adapted with weaker supervision.

For the warm-start branch, prior work on T-SKM-Net has shown that the predictor can be trained without solver-generated ground-truth solutions by directly optimizing the objective together with constraint-violation penalties. For the active-set prediction branch, exact solver-derived active-set labels are also not strictly necessary. Since this branch is only used to bias the SKM sampling distribution toward likely active constraints, pseudo labels can be constructed from corrected solutions produced by the model itself. Specifically, for each target sample, the transferred model first produces a warm-start solution, the SKM layer then corrects it to a feasible solution, and the active constraints of this corrected solution are used as pseudo labels for retraining the active-set heads.

To evaluate this possibility, we consider two transfer settings: from the IEEE 57-bus system to the IEEE 118-bus system, and from the IEEE 118-bus system to the IEEE 300-bus system. In each setting, we first train a source model on the smaller system and transfer it to the larger target system. We then freeze the shared HGNN backbone and the warm-start output branch, and retrain only the active-set prediction heads using 5% of the target N-1 scenarios. Importantly, these active-set heads are not trained with solver-derived active-set labels; instead, they use pseudo labels generated from the model's own corrected SKM outputs.

Table 13: Transfer learning results with pseudo-label active-set training. Runtime values are reported in milliseconds. The notation $57\rightarrow118$ denotes transfer from the IEEE 57-bus system to the IEEE 118-bus system.

TRANSFER	METHOD	EQ. PROJ.	EQ. SPD.	INEQ. ITER.	INEQ. TIME	INEQ. SPD.	SKM SPD.	GAP (%)
$57\rightarrow118$	T-SKM-NET	1.54 ± 0.02	$1.00\times$	$57.31\,(148)$	5.93 ± 7.07	$1.00\times$	$1.00\times$	0.009
$57\rightarrow118$	AT-SKM-AC	0.28 ± 0.02	$5.50\times$	$29.28\,(59)$	1.96 ± 1.58	$3.02\times$	$3.33\times$	0.009
$118\rightarrow300$	T-SKM-NET	7.01 ± 0.08	$1.00\times$	$30.83\,(218)$	2.06 ± 2.53	$1.00\times$	$1.00\times$	0.618
$118\rightarrow300$	AT-SKM-AC	0.74 ± 0.05	$9.47\times$	$6.30\,(18)$	0.51 ± 0.54	$4.03\times$	$7.26\times$	0.590

As shown in Table 13, AT-SKM-AC retains substantial acceleration even when the active-set heads are trained only with pseudo labels. In the $57\rightarrow118$ transfer setting, the average number of inequality iterations decreases from 57.31 to 29.28, and the worst-case iteration count decreases from 148 to 59. In the more challenging $118\rightarrow300$ setting, the average iteration count decreases from 30.83 to 6.30, and the worst-case iteration count decreases from 218 to 18. These reductions show that pseudo-label active-set learning still provides useful sampling guidance after transfer to a larger system.

The runtime results show the same trend. In the $57\rightarrow118$ setting, AT-SKM-AC achieves a $5.50\times$ speedup in equality projection, a $3.02\times$ speedup in inequality iteration, and a $3.33\times$ total SKM speedup. In the $118\rightarrow300$ setting, the corresponding speedups are $9.47\times$, $4.03\times$, and $7.26\times$. Meanwhile, the optimality gap remains comparable to T-SKM-Net: 0.009% versus 0.009% for $57\rightarrow118$, and 0.590% versus 0.618% for $118\rightarrow300$. Since the final outputs are corrected by the projection and SKM layers, both methods preserve strict constraint feasibility after correction.

These results suggest that supervised ground-truth solutions are a training choice in the current implementation rather than a fundamental requirement of AT-SKM-Net. When exact solver-generated labels are expensive or insufficient on larger systems, the model can still transfer from a smaller system and use self-generated pseudo labels to train the active-set predictor, thereby retaining much of the acceleration benefit while preserving strict feasibility.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are supported by the results in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix B.3 discusses that although the Cholesky low-rank update theoretically reduces the complexity to $O(n^2)$, memory limitations may become an important factor affecting algorithmic speed on larger networks.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Appendix A.2 provides complete proofs of the proposed theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 and Appendix D provide the computing hardware and relevant parameter settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the code and data as supplemental material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix D provides the details of the experimental settings, including data splits, hyperparameters, and optimizer types.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Appendix D.5 reports the mean and standard deviation of the runtime.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5 reports the hardware used in our experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The data we used is all open source, including mathematics and code, and does not contain any content that violates ethical requirements.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our paper does not specifically discuss potential positive or negative societal impacts, as its primary focus is on methodological innovation and technical evaluation. This section is not directly applicable to our work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All public datasets and resources used in this work are properly cited and credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.