Multi-ResNets for Low Rank Preconditioning in Constrained Optimization

Anonymous Authors[1]

Abstract

We propose Multi-ResNets for optimization problems that encode inductive biases toward low-rank constraint set approximations. By leveraging compressed representations and embedding a priori structure, the method enables effective low-rank preconditioning, improving convergence and reducing sensitivity to local minima. This hierarchical formulation supports greedy refinement, where low-rank solutions initialize and guide optimization in the full space. Empirically, Multi-ResNets achieve 2–6× reductions in primary constraint violation and improved convergence across convex and non-convex benchmarks.

1. Introduction

Algorithms for training neural networks to solve optimization problems have recently seen rapid development (Donti et al., 2021). However, these approaches often fail to incorporate the structure of the underlying optimization problem into the network architecture. This is in contrast to traditional methods, which exploit techniques such as Galerkin projection, compression, and preconditioning to guide the search towards high-quality solutions. Such methods often rely on low-rank approximations that capture much of the problem structure, enabling exploration in a reduced setting before initializing the full model. At present, there is little connection between this strand of neural network research and the traditional methods that employ these ideas.

We develop neural networks with an inductive bias induced by low-rank approximations of the constraint operator, yielding structured solution manifolds that extend to the full space. When the constraint operator admits a low-rank representation, the network exploits this structure to improve search quality. Furthermore, a priori problem structure can be embedded in the architecture, enabling low-rank preconditioning and mitigating convergence to poor local minima.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Low Rank Preconditioning for Optimization

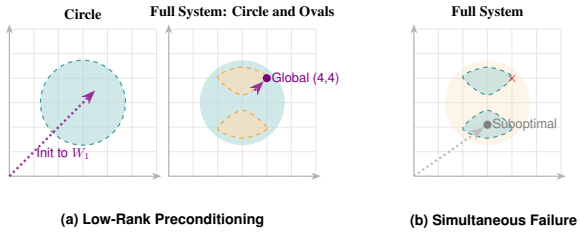

Figure 1. Two constraint system of circle and ovals feasible regions: Low rank navigates via preconditioning on circle constraint, simultaneous application falls to local minima, see Example A.3.

2.1. Neural Network Proposals for Optimization

We train a neural network $N_\theta : \mathbb{V} \mapsto \mathbb{W}$, with $\theta \in \mathbb{R}^p$, that maps v to a proposed optimal solution $w = N_\theta(v)$ for:

$$\min_{w \in \mathbb{W}} f(w \mid v) \quad \text{subject to} \quad g(w \mid v) \leq 0, \quad h(w \mid v) = 0,$$

where $f > 0$ is a coercive function, $g : \mathbb{W} \times \mathbb{V} \to \mathbb{R}^{m_1}$, and $h : \mathbb{W} \times \mathbb{V} \to \mathbb{R}^{m_2}$. For now, $v \in \mathbb{V} = \mathbb{R}^n$, $\mathbb{W} = \mathbb{R}^d$; later we use graph domain and co-domains, the main topic of our computational experiments. Assuming continuous f, g, and h, N_θ can be generalized over a given problem type.

The DC3 framework (Donti et al., 2021) employs a neural network $N_\theta : \mathbb{R}^n \to \mathbb{R}^D$ to propose a candidate solution $z \in \mathbb{R}^D$ with reduced dimensionality $D < d$ (e.g., $D = d - m_2$). This reduction constrains the networks output to match the solution manifold's degrees of freedom. Equality constraints h are satisfied by completing the partial proposal $\tilde{w} = z$ via a projection operator $\mathcal{T}(z) = w$ onto the solution manifold. Subsequently, inequality constraints are enforced by performing gradient descent directly along this manifold to obtain $\hat{w} = \Pi(\tilde{w})$, as detailed in Algorithm 3. The process comprises three stages: a neural network proposal, numerical completion, and gradient-based correction. Implicit differentiation enables backpropagation through the solver, while gradient descent operates along the solution manifold to maintain equality constraints within the fine-mesh limit of the learning step size (Donti et al., 2021). This approach assumes that generated points are both *completable* and *correctable*. For this, DC3 relies on these proposals falling within a local basin of attraction; failing this, the procedure risks divergence or entrapment in poor local optima, see Figure 4 for example.

More broadly, several constraint-enforcement paradigms have been proposed for ML-based optimization. Penalty methods (Amos & Kolter, 2017; Agrawal et al., 2019) add constraint violations to the loss but offer no feasibility guarantees. Projection-based methods map outputs onto the feasible set but are expensive for general non-convex constraints. Iterative methods, including DC3 (Donti et al., 2021) and FSNet (Nguyen & Donti, 2025), integrate completion and correction steps directly into training, achieving feasibility through differentiable solvers. Our work is complementary: rather than proposing a new solver or correction mechanism, we introduce an architectural modification that decomposes constraints by priority, enabling low-rank preconditioning within existing predict-complete-correct pipelines. This is inspired by classical preconditioning strategies—multigrid and Galerkin methods—where coarse approximations guide the search before fine-scale refinement, an idea with recent success in multi-resolution neural architectures (Williams et al., 2023; Falck et al., 2022).

2.2. Lexicographic Constraint Importance

Constructing our low rank constraint manifold requires ranking constraints by priority. To do this, we may unify equality and inequality constraints through $C(w \mid v) = 0$, where $C(w \mid v) = \max(0, g(w \mid v))$ for inequalities and $C(w \mid v) = h(w \mid v)$ otherwise. We will construct a sequence of low rank approximation spaces $\{W_i\}_{i=1}^r$ which enables greedy sequential refinement in our chosen constraint ordering, using low-rank solutions as a preconditioner to initialize the search in the full solution space.

2.3. Low Rank Subspace Proposal Solutions

The performance of the optimization heavily depends on the quality of the initial neural network proposal. At initialization, the model's inductive bias dictates search efficacy; a poorly initialized network may fail to explore the space adequately, hindering the discovery of feasible solutions that satisfy domain-specific performance metrics. To address this, we propose that the initial neural network proposal should be a low rank approximation of the solution (Section A.1.1). In non-optimization contexts, various neural network designs (Dimola et al., 2026; Han & Lee, 2023; Chen & Koltun, 2017; Franco et al., 2023), such as the U-Net (Ronneberger et al., 2015), leverage a sequence of lower rank approximation spaces—a filtration—that guides the approximation mechanism, allowing the network to transition from low rank approximations toward the desired manifold.

These approximation spaces encode ordinal priority, made from the choice of low rank initialization made. We operationalise this principle via a Multi-ResNet architecture (Definition 3 (Williams et al., 2023)), where the descending filtration $\{W_i\}_{i=1}^r$ enacts low-rank preconditioning: the network first solves on a higher-dimensional subspace defined by fewer constraints before refining to the full constraint set. Specifically, W_1 serves as the initial low rank approximation, conditioning the optimization to resolve high-fidelity solutions within W_r. Multi-ResNets are applicable by describing our target feasible sets W_r as embedded within the following nested structure for constraints (either equality or inequality) $\{C_j\}_{j=1}^n$:

$$W_r \subset W_{r-1} \subset \cdots \subset W_1 \subset W_0 = \mathbb{W}, \quad (1)$$

$$W_i := \{w \in \mathbb{W} \mid C_j(w) = 0 \text{ for all } j \leq i\}, \quad (2)$$

where $\mathbb{W}$ is the ambient solution space. This construction of the $\{W_i\}_{i=0}^r$ allows for Definition 1 (Williams et al., 2023) to be utilized. By explicitly aligning the neural architecture with this filtration, the Multi-ResNet embeds an inductive bias that prioritizes low rank feasibility. Moreover, this structural priority facilitates robust optimization even when higher-order constraints are intractable. We illustrate this advantage in Example A.3 (Figure 4): while simultaneous constraint enforcement succumbs to basin trapping induced by non-convexity and disjoint regions, our hierarchical approach successfully recovers the global optimum.

3. Multi-ResNets for Low Rank Proposals

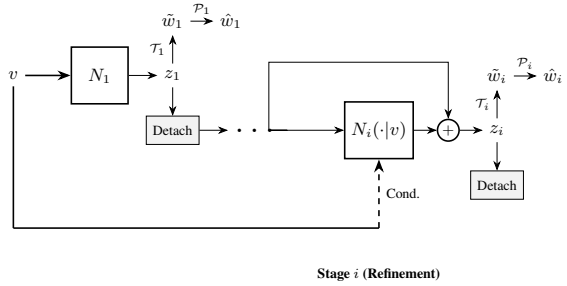

Figure 2. MRes-LR architecture, a form of U-Net, see Figure 8

3.1. Multi-ResNets:Sequential Low Rank Proposals

To internalize the lexicographic hierarchy as an inductive bias and initialize our low rank preconditioning, (Williams et al., 2023) structures the neural network as a composition of sequential operators. In the Multi-ResNet framework, each stage refines the previous estimate to satisfy a progressively more restrictive level of the constraint filtration $W_1 \supseteq W_2 \supseteq \cdots \supseteq W_r$. This framework formalizes this refinement using a sequence of *Conditional ResNets* $\mathcal{R} : \mathbb{W} \times \mathbb{V} \to \mathbb{W}$ (Defn. A.1) acting through

$$w^{(k)} := w^{(k-1)} + \mathcal{R}_{\text{res}}^{(k)}(w^{(k-1)} \mid v), \quad w^1 := \mathcal{R}_{\text{res}}^{(1)}(v),$$

where $\mathcal{R}_{\text{res}} : \mathbb{W} \times \mathbb{V} \to \mathbb{W}$ is the residual of $\mathcal{R}$. In the context of low rank preconditioning, this residual structure functions not merely as a device to facilitate gradient flow,

but as a formal mechanism for sequential projection onto an initial lower rank manifold. By initializing the forward pass with a base mapping projecting directly onto the manifold of (W_1), we ensure that all subsequent residual blocks act as corrective adjustments. Specifically, for an r-stage network, the low-rank state $w^{(k-1)}$ serves as the preconditioner for stage k. This architecture enforces the W_i sequence through three mechanisms:

First, the network provides physically feasible preconditioning via the initial stage $\mathcal{R}^{(1)}$, which is trained to satisfy inviolable constraints $C_1(w) = 0$, thereby initializing our search within the valid low rank manifold W_1. Second, it performs refinement, where each residual $\mathcal{R}^{(k)}_{\text{res}}$ is responsible for the transition from W_{k-1} to W_k (e.g., enforcing a low rank residual problem). Finally, the architecture enables a safe low rank solution: if a stage k encounters an overdetermined or infeasible operational constraint, the lower rank valid solution $w^{(k-1)}$ established by the previous, higher-priority preconditioner, may be utilized.

3.2. Multi-ResNet Low Rank (MRes-LR) Architecture

Algorithm 1 Forward Pass

Input: v, $\{\theta_i\}_{i=1}^r$
$z_1 \leftarrow N_{\theta_1}(v)$, $\hat{w}_1 \leftarrow \mathcal{P}_1(\mathcal{T}_1(z_1))$, $\mathcal{L}_{tot} \leftarrow \mathcal{L}_1(\hat{w}_1)$
for $i = 2$ **to** n **do**
  Detach z_{i-1}
  $z_i \leftarrow z_{i-1} + N_{\theta_i}(z_{i-1} \mid v)$
  $\hat{w}_i \leftarrow \mathcal{P}_i(\mathcal{T}_i(z_i))$, $\mathcal{L}_{tot} \mathrel{+}= \mathcal{L}_i(\hat{w}_i)$
end for
Return $\mathcal{L}_{tot}$

Our architecture, Multi-ResNet Low Rank (MRes-LR), integrates the implicit constraint solvers of DC3 with the Multi-ResNet framework to solve optimization problems with low-rank preconditioning. A non-residual anchoring step, $z_1 = N_1(v)$, first captures the first low rank solution on W_1. This establishes a valid initial preconditioning for subsequent stages to build hierarchical residuals, embedding this solution as a structural inductive bias.

For $i \in \{2, \ldots, n\}$, the architecture uses a conditional ResNet update: $z_i = \text{Detach}(z_{i-1}) + N_i(\text{Detach}(z_{i-1}) \mid v)$. This stop-gradient operation enforces a greedy priority, preventing lower-level operational constraints from compromising the physical consistency of the foundational manifold W_1. Latent estimates z_i are then mapped to feasible space via sequence-specific completion and correction operators: $\tilde{w}_i = \mathcal{T}_i(z_i)$ and $\hat{w}_i = \mathcal{P}_i(\tilde{w}_i)$, ensuring $\hat{w}_i \in W_i$.

The completion operator $\mathcal{T}_i$ generates the auxiliary variables for the i-th level, while the correction operator $\mathcal{P}_i$ enforces strict satisfaction of $C_i(\hat{w}_i) = 0$. As $W_i \subseteq W_{i-1}$, each refinement preserves the feasibility of all higher-priority stages, providing a fallback: if a later stage fails, the low rank output from the previous stage remains available.

3.3. Multi-Stage Low Rank Subspace Training

Algorithm 2 Multi-stage Low Rank Preconditioning

In: $(\mathcal{V}, \mathcal{W})$, prior μ, stages K
$\hat{w}_0 \leftarrow \mu(\mathcal{V})$ ▷ Anchor W_1
for $k = 1 \ldots K$ **do**
  $E \leftarrow \mathcal{W} - \hat{w}_{k-1}$; $\theta_k \leftarrow$ init
  $\theta_k \leftarrow$ optimize $\mathcal{L}_k(E - \mathcal{R}_{\text{res},k}(\text{Detach}(\hat{w}_{k-1}))$
  $\hat{w}_k \leftarrow \hat{w}_{k-1} + \mathcal{R}_{\text{res},k}(\text{Detach}(\hat{w}_{k-1}), \mathcal{V}; \theta_k)$
end for
Out: $\mathcal{F}_K = \mathcal{R}_K \circ \cdots \circ \mathcal{R}_1$

Our staged training paradigm adopts the staged training procedure of Algorithm 1 (Williams et al., 2023), which treats the transition between subspaces $W_k \to W_{k+1}$ as a sequential refinement of the residual error. By strictly detaching these prior stages during optimization, each residual block $\mathcal{R}_{\text{res},k}$ isolates the resolution of remaining operational violations, thereby preserving the topological anchoring of the fundamental power flow manifold (W_1).

4. Experiments

In this section, we evaluate the proposed low-rank preconditioning framework across benchmarks testing global optimality, robustness under infeasibility, and scalability to high-dimensional physical systems. We demonstrate how the structural inductive bias of the MRes-LR architecture outperforms standard simultaneous satisfaction methods.

4.1. Bimodal Basin Trapping

Motivated by Figure 4, we validate the inductive bias on a synthetic benchmark with optimum $w^* = (4, 4)$ under manifolds W_{circle} and W_{ovals} (see Example A.3). Compared to an MLP with weighted penalty loss, the baseline is consistently trapped in a local basin, whereas low-rank preconditioning avoids this by anchoring N_1 to W_1 and refining via residual stages. This gravity well test highlights distributional collapse: Table 4 shows 100% trapping for the baseline and 100% success for our method.

4.2. Low Rank Fallback in Over-constrained Scenarios

Consider an over-constrained problem, e.g. a physical system with policy constraints, where only the physical component is guaranteed feasible. A low-rank approximation then yields a meaningful physical solution, whereas solving all constraints simultaneously may fail to converge.

First, we simulate $W_1 \cap W_2 = \emptyset$ (Example A.4). As predicted by Proposition A.2, the weighted penalty baseline

converges to solutions violating both constraints, whereas low-rank preconditioning exhibits a fallback: N_1 is greedily optimized and preserved via detachment, maintaining near-zero violation of W_1 even when N_2 fails.

Second, we consider a physical system with a policy constraint via AC Optimal Power Flow (ACOPF), which minimizes generation cost subject to non-linear power flow and operational constraints, yielding a non-convex feasible set (full formulation in Section A.1.2). The $2n_b$ power balance equalities constrain the solution to a manifold of intrinsic dimension $D \ll d$, where $d = 2n_g + 2n_b$. The neural network predicts only the D independent variables, and a Newton power flow solver completes the remaining $d - D$ dependent variables. This low-rank structure motivates the staged approach—Stage 1 operates on the D-dimensional equality manifold intersected with box bounds (W_1), before Stage 2 further restricts to thermal limits ($W_2 \subseteq W_1$). We use a minimal 3-bus system (2 generators, 3 branches) to visualise nested feasibility: Figure 3 shows max line flow over (P_{g_2}, V_{m_2}) with planes indicating thermal limits $\bar{S}$.

This allows us to study the over-constrained regime: while physical constraints admit feasible solutions, the thermal limit $\bar{S} = 38$ MVA renders the problem infeasible. We train DC3 and MRes-LR on this system under tightened bounds ($P_g \in [80, 200]$ MW, $Q_g \in [-30, 80]$ MVAr, $V_m \in [0.95, 1.05]$ p.u.). All models use 2 layers with 16 hidden units, trained for 500 epochs on 2000 samples.

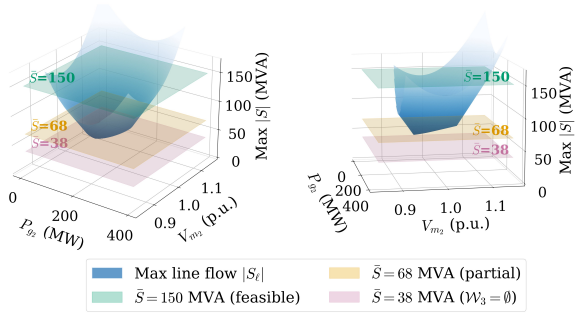

Figure 3. 3-bus ACOPF feasibility. **Left**: front view. **Right**: side view. Blue surface: max line flow $|S_\ell|$ over (P_{g_2}, V_{m_2}). Horizontal planes show thermal limits: $\bar{S} = 150$ MVA (teal) intersects ($\mathcal{W}_3$ exists); $\bar{S} = 68$ MVA (amber) leaves a small feasible region; $\bar{S} = 38$ MVA (pink) is infeasible ($\mathcal{W}_3 = \emptyset$).

Table 1. 3-bus ACOPF results (mean ± std over 3 seeds). Equality violations are near-zero for both methods ($\sim 10^{-7}$). MRes-LR preserves Tier-1 constraints 3.1× better than DC3.

Method	T1 box (p.u.)	T2 flow (p.u.)
DC3	0.0054 ± 0.0012	$\mathbf{0.558 \pm 0.003}$
MRes-LR	$\mathbf{0.0017 \pm 0.0002}$	0.574 ± 0.003

4.3. Synthetic Quadratic Programming Results

We consider a quadratic program with two-tier constraints to isolate the effect of low-rank preconditioning:

$$\min_{w \in \mathbb{R}^d} \quad f(w) = \tfrac{1}{2} w^\top Q w + c^\top w, \tag{3}$$

subject to an equality constraint (enforced by completion) and two constraint tiers defining feasible sets W_1 and $W_2 \subseteq W_1$. Performance is reported as mean ± std. constraint violation (p.u.) over 4 seeds (2026–2029). All methods use budget 10 and are trained for 500 epochs.

Linear (Convex) Constraints. Tier-1 and Tier-2 are linear, yielding convex $W_2 \subseteq W_1$. In the convex setting, where the problem is well-behaved, MRes-LR enforces near-perfect Tier-1 feasibility but degrades Tier-2 due to gradient decoupling, while DC3 shows intermediate violations on both tiers.

Table 2. Convex (linear) synthetic results (4 seeds, budget 10).

Method	Tier-1 (p.u.)	Tier-2 (p.u.)
DC3	0.0530 ± 0.044	$\mathbf{0.1885 \pm 0.125}$
MRes-LR	$\mathbf{0.0005 \pm 0.000}$	0.2035 ± 0.125

Non-linear (Non-Convex) Constraints. Tier-2 is non-linear, introducing additional structure in the constraint landscape. In the non-convex setting, MRes-LR outperforms DC3 on both tiers, demonstrating that the low-rank preconditioning benefit extends beyond the convex case.

Table 3. Non-convex synthetic results (4 seeds, budget 10).

Method	Tier-1 (p.u.)	Tier-2 (p.u.)
DC3	0.0343 ± 0.021	0.0356 ± 0.002
MRes-LR	$\mathbf{0.0054 \pm 0.001}$	$\mathbf{0.0136 \pm 0.006}$

5. Conclusion

We utilized Multi-ResNets for optimization, a hierarchical architecture that encodes low-rank inductive biases for constrained optimization. By leveraging low-rank preconditioning the method improves convergence and robustness to local minima. Empirically, it achieves consistent gains across convex and non-convex settings. In over-constrained cases, it exhibits a low-rank fallback, preserving primary feasibility when full feasibility is unattainable.

Limitations. The constraint ordering requires domain knowledge and is not learned automatically. The detach operator is effective in convex settings but degrades under tightly coupled nonlinear constraints. Our current implementation uses identity projections which can later also be learned. Experiments are small-scale to isolate the low-rank preconditioning effect; larger ACOPF and broader-domain validation remain future work.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., and Kolter, J. Z. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Amos, B. and Kolter, J. Z. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning (ICML)*, pp. 136–145, 2017.

Chen, Q. and Koltun, V. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1520–1529, 2017.

Dimola, N., Rares Franco, N., and Zunino, P. Numerical solution of mixed-dimensional pdes using a neural preconditioner. *Computers & Mathematics with Applications*, 206:58–79, 2026. ISSN 0898-1221. doi: https://doi.org/10.1016/j.camwa.2025.12.013. URL https://www.sciencedirect.com/science/article/pii/S0898122125005255.

Donti, P., Rolnick, D., and Kolter, J. Z. DC3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations (ICLR)*, 2021.

Falck, F., Williams, C., Danks, D., Deligiannidis, G., Yau, C., Holmes, C., Doucet, A., and Willetts, M. A multi-resolution framework for U-Nets with applications to hierarchical VAEs. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15529–15544, 2022.

Franco, N. R., Manzoni, A., and Zunino, P. Mesh-informed neural networks for operator learning in finite element spaces. *Journal of Scientific Computing*, 97(2):35, 2023.

Han, J. and Lee, Y. Hierarchical learning to solve pdes using physics-informed neural networks. In Mikyška, J., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V. V., Dongarra, J. J., and Sloot, P. M. (eds.), *Computational Science – ICCS 2023*, pp. 548–562, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-36024-4.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Nguyen, H. T. and Donti, P. L. FSNet: Feasibility-seeking neural network for constrained optimization with guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Williams, C., Falck, F., Deligiannidis, G., Holmes, C., Doucet, A., and Syed, S. A unified framework for U-Net design and analysis. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A. Appendix

A.1. Delayed Definitions

A.1.1. Low-Rank Constraint Operators

We formalize the notion of low-rank structure in the *constraint operator* of an optimization problem. Consider constraints defined by $g : \mathbb{R}^n \to \mathbb{R}^m$, with feasible set

$$\mathcal{V} = \{x \in \mathbb{R}^n : g(x) = 0\}. \tag{4}$$

In the linear case, where $g(x) = Ax - b$, the rank of the constraint operator is $\operatorname{rank}(A)$, and the feasible set is an affine subspace of dimension $n - \operatorname{rank}(A)$. A low-rank constraint operator ($\operatorname{rank}(A) \ll \min(m, n)$) therefore restricts solutions to a low-dimensional subspace.

In the nonlinear setting, we extend this notion by considering the intrinsic dimension of the feasible set induced by the constraints. We say that the *constraint operator admits a low-rank representation* if $\mathcal{V}$ has low intrinsic dimension relative to the ambient space, i.e.,

$$\dim(\mathcal{V}) \ll n. \tag{5}$$

Locally, this corresponds to the Jacobian $\nabla g(x)$ having low rank.

In practice, we approximate the constraint operator (or its linearization) with a low-rank representation, thereby inducing a reduced feasible manifold that captures the dominant constraint structure and guides optimization.

A.1.2. ACOPF Problem Formulation

The AC Optimal Power Flow (ACOPF) problem minimizes generation cost subject to nonlinear power flow physics and operational limits. For a network with n_b buses, n_g generators, and n_ℓ branches:

$$\min_{P_g, Q_g, V_m, V_a} \quad \sum_{i=1}^{n_g} \left(c_{2,i} P_{g,i}^2 + c_{1,i} P_{g,i}\right) \tag{6}$$

$$\text{s.t.} \quad P_{g,i} - P_{d,i} = \sum_{k=1}^{n_b} V_{m,i} V_{m,k} (G_{ik} \cos\theta_{ik} + B_{ik} \sin\theta_{ik}), \tag{7}$$

$$Q_{g,i} - Q_{d,i} = \sum_{k=1}^{n_b} V_{m,i} V_{m,k} (G_{ik} \sin\theta_{ik} - B_{ik} \cos\theta_{ik}), \tag{8}$$

$$P_{g,i}^{\min} \le P_{g,i} \le P_{g,i}^{\max}, \quad Q_{g,i}^{\min} \le Q_{g,i} \le Q_{g,i}^{\max}, \tag{9}$$

$$V_{m,i}^{\min} \le V_{m,i} \le V_{m,i}^{\max}, \tag{10}$$

$$|S_\ell^f|^2, |S_\ell^t|^2 \le (\bar{S}_\ell)^2, \quad \forall \ell \in \{1, \ldots, n_\ell\}, \tag{11}$$

where $\theta_{ik} = V_{a,i} - V_{a,k}$, $G + jB = Y_{\text{bus}}$ is the network admittance matrix, and S_ℓ^f, S_ℓ^t are the complex power flows at the from/to ends of branch ℓ.

The full solution vector is $w = [P_g, Q_g, V_m, V_a] \in \mathbb{R}^d$ with $d = 2n_g + 2n_b$. The equality constraints (7)–(8) define $m_{\text{eq}} = 2n_b$ equations. Following DC3 (Donti et al., 2021), the neural network predicts only the *partial* (independent) variables $z = [P_g^{\text{PV}}, V_m^{\text{slack}\cup\text{PV}}] \in \mathbb{R}^D$, and a Newton power flow solver $\mathcal{T}$ completes the remaining (dependent) variables $[V_a^{\text{PV}\cup\text{PQ}}, V_m^{\text{PQ}}, P_g^{\text{slack}}, Q_g]$ to satisfy (7)–(8). The equality manifold therefore has intrinsic dimension $D \ll d$. In realistic transmission grids, load buses far outnumber generators ($n_g \ll n_b$), making this reduction substantial: for the IEEE 30-bus system ($n_g = 6$, $n_b = 30$), $D = 11$ out of $d = 72$, so the neural network learns only 15% of the full solution vector. For our 3-bus test system, $D = 3$ out of $d = 10$.

Tier hierarchy. The inequality constraints are partitioned into two tiers reflecting operational priority:

- **Tier 1** (box bounds): generator limits (9) and voltage limits (10),

- **Tier 2** (line flows): thermal limits (11).

This yields the nested feasible sets $W_1 \supseteq W_2$, where W_1 is the equality manifold intersected with box bounds, and $W_2 = W_1 \cap \{w : |S_\ell| \leq \bar{S}_\ell\}$. Our 3-bus experiments use the system with $D = 3$, $d = 10$.

A.1.3. Neural Network Architecture Definitions

Definition A.1 (ResNet, Conditioning ResNet, (He et al., 2016; Williams et al., 2023))**.** A mapping $\mathcal{R} : \mathbb{W} \times \mathbb{V} \to \mathbb{W}$ is defined as a *ResNet* preconditioned on $\mathcal{R}_{\text{pre}} : \mathbb{V} \to \mathbb{W}$ if: $\mathcal{R}(w \mid v) = \mathcal{R}_{\text{pre}}(v) + \mathcal{R}_{\text{res}}(w \mid v)$. where $\mathcal{R}_{\text{res}} : \mathbb{W} \times \mathbb{V} \to \mathbb{W}$ is the residual of $\mathcal{R}$. A *Conditional ResNet* is conditioned on the input $v \in \mathbb{V}$ and the solution estimate from the previous stage $w^{(k-1)} \in \mathbb{W}$: $w^{(k)} := w^{(k-1)} + \mathcal{R}_{\text{res}}^{(k)}(w^{(k-1)} \mid v)$ and $w^1 := \mathcal{R}_{\text{res}}^{(1)}(v)$.

A.1.4. Delayed Proofs

Proposition A.2 (Infeasibility of Weighted Penalties for Disjoint Sets)**.** *Let $W_1, W_2 \subset \mathbb{W}$ be two disjoint, non-empty closed sets. Let $d(w, W_i) = \inf_{z \in W_i} \|w - z\|_2$ denote the distance function to each set. For any convex combination of the penalty terms $\mathcal{L}(w) = \alpha d(w, W_1)^2 + (1-\alpha) d(w, W_2)^2$ with $\alpha \in (0, 1)$, the minimizer $w^* = \arg\min_w \mathcal{L}(w)$ satisfies $w^* \notin W_1 \cup W_2$.*

Proof. The gradient of the squared distance function to a closed set is $\nabla d(w, W_i)^2 = 2(y - \text{Proj}_{W_i}(w))$. The first-order optimality condition for $\mathcal{L}(w)$ implies:

$$\alpha(w^* - \text{Proj}_{W_1}(w^*)) + (1-\alpha)(w^* - \text{Proj}_{W_2}(w^*)) = 0. \tag{12}$$

Rearranging shows that w^* is a convex combination of its projections onto the two sets. Since the sets are disjoint, the distance between the projections is strictly positive, forcing w^* to lie strictly between the sets, thus violating both constraints. □

A.2. Subspace Preconditioning Examples

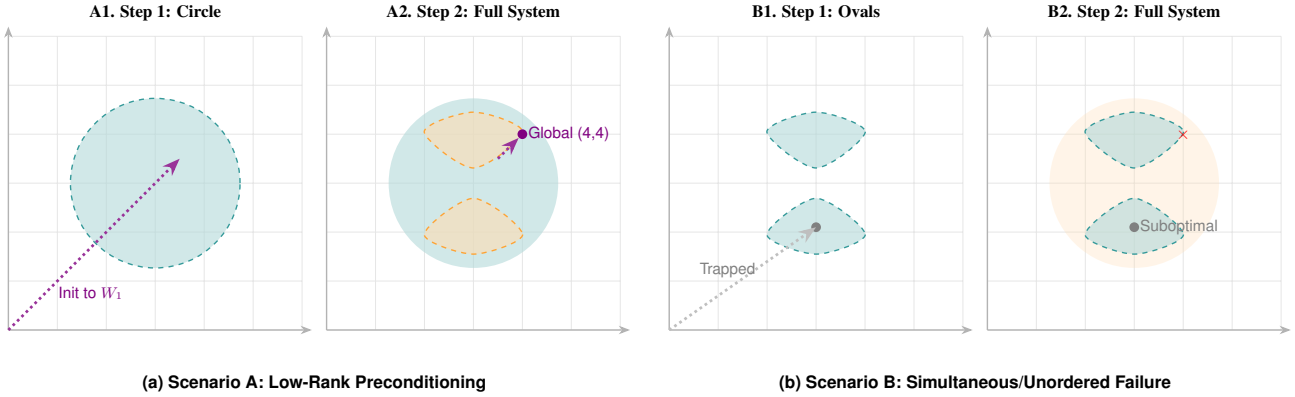

Figure 4. Visualization of basin trapping: Low-rank preconditioning (left) navigates global topology via lexicographic priority, whereas simultaneous or incorrectly ordered learning (right) succumbs to local entrapment.

Example A.3 (Bimodal Basin Trapping)**.** Consider finding a feasible point for $f(w) = \|w - (4,4)\|_2^2$ in $\mathbb{R}^2$, initialized at $(0,0)$. The system is governed by a physical constraint $C_1(w) = \max(0, (w_1-3)^2 + (w_2-3)^2 - 3)$ and an operational constraint $C_2(w) = \max(0, (w_1-3)^2 + ((w_2-3)^2 - 1.1)^2 - 1.01)$. Simultaneous enforcement of $\{C_1, C_2\}$, or prioritising C_2, causes the zero-centered initialization to descend into a suboptimal local basin near $(3, 2.1)$. Because the manifold $W_1 \cap W_2$ is non-convex and disconnected, the solver becomes trapped. Conversely, as shown in Figure 4, adopting the filtration $W_1 \subset W_0$ first relaxes the search to the circle defined by C_1. This acts as a global preconditioner, anchoring the solution in the upper quadrant. Once C_2 is introduced, the model is already positioned within the correct basin to reach the global optimum at $(4,4)$.

Table 4. Bimodal Basin Trapping ($N = 500$ test samples).

Method	Cost	L_1	Success Rate (%)
Simultaneous	3.4841	0.0000	0.0%
MRes-LR	0.0320	0.0000	100.0%

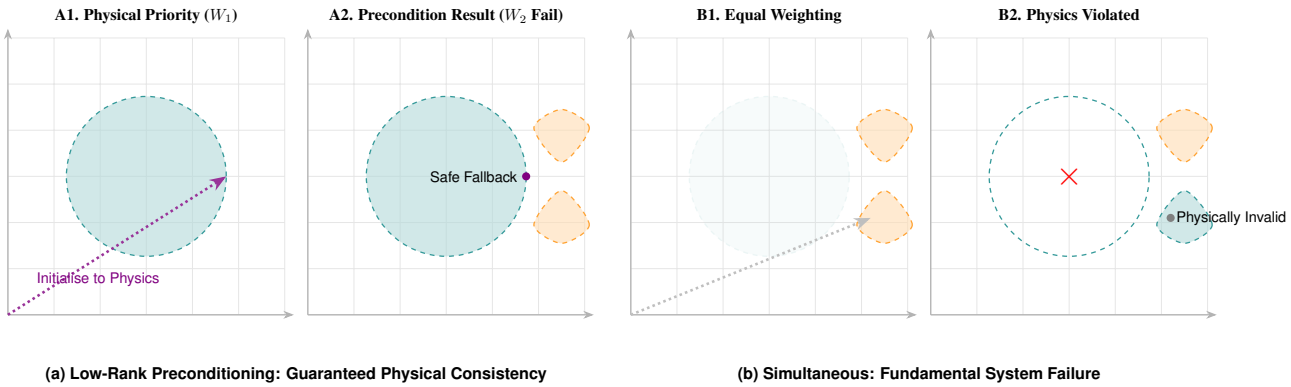

Figure 5. A comparison of low-rank preconditioning and simultaneous satisfaction under over-constrained conditions. In Scenario A, low-rank preconditioning ensures the model defaults to a physically valid solution. In Scenario B, simultaneous satisfaction allows operational goals to pull the solution out of the physical manifold, resulting in an invalid system state.

Example A.4 (Basin Trapping and Physical Fallback)**.** Consider finding a feasible point for $f(y) = \|w - (4,4)\|_2^2$ in $\mathbb{R}^2$, initialized at $(0,0)$. The system is governed by a physical constraint $C_1(w) = \max(0, (w_1-3)^2 + (w_2-3)^2 - 3)$ and an operational constraint $C_2(w) = \max(0, (w_1-3)^2 + ((w_2-3)^2 - 1.1)^2 - 1.01)$. As illustrated in Figure 5, low-rank preconditioning successfully identifies a physically feasible solution even when no global feasible solution exists. In scenarios where operational requirements are incompatible with physics ($W_1 \cap W_2 = \emptyset$), the model defaults to a *safe fallback* on the physical manifold because it prioritizes the satisfaction of W_1. Conversely, a simultaneous enforcement approach often yields a solution that satisfies neither constraint, violating fundamental physical laws in a failed attempt to reach the operational basin.

Table 5. Comparison of Mean and Maximum Constraint Violations across 500 Samples.

Method / Stage	Metric	Average Error	Max Error
Simultaneous	L_1	6.99906	7.38585
	L_2	7.00127	7.15303
MRes-LR Stage 1	L_1	0.00000	0.00000
MRes-LR Stage 2	L_1	7.00254	7.29045
	L_2	6.99857	7.18668

A.3. Antecedents of the DC3 Multi-ResNet Architecture

A.3.1. DC3 Hard Constraint Networks

The DC3 (Deep Constraint Completion and Correction) framework is a differentiable approach designed to solve constrained optimization problems structured as $\min_w f_v(w)$ subject to inequality constraints $g_v(w) \leq 0$ and equality constraints $h_v(w) = 0$. By integrating these constraints directly into the neural network architecture, the method enforces equality constraints through completion and attempts to reduce inequality violations through unrolled correction, in two sequential stages.

In the first stage, known as equality completion, the framework satisfies the equality constraints $h_v(w) = 0$ by partitioning the decision variable w into independent variables z and dependent variables $\phi_v(z)$. During the forward pass, the neural network N_θ predicts the independent component z, and a completion function $\mathcal{T}(z)$ solves for $\phi_v(z)$ such that the equality is maintained. To facilitate end-to-end training, the backward pass propagates gradients through this implicit solve using the

Algorithm 3 DC3 Training

Initialise $N_\theta : \mathbb{R}^n \to \mathbb{R}^D$ $\{D < d\}$
while not converged **do**
  for $v \in \mathcal{V}$ **do**
    Propose $z = N_\theta(v)$
    Complete $\tilde{w} = \mathcal{T}(z) = \left[z, \phi_v(z)\right]^T$ {Project to equality constraints}
    Correct $\hat{w} = \Pi(\tilde{w})$ {Project to inequality constraints}
    Compute regularized loss $f(\hat{w} \mid v)$
    Update θ using $\nabla_\theta f(\hat{w} \mid v)$
  end for
end while

Implicit Function Theorem, which defines the Jacobian as:

$$\frac{\partial \phi_v(z)}{\partial z} = -\left[\frac{\partial h_v}{\partial \phi_v}\right]^{-1} \frac{\partial h_v}{\partial z}. \tag{13}$$

The second stage, inequality correction, addresses the constraints $g_v(w) \leq 0$ through a differentiable procedure that adjusts the initial completed solution $\tilde{w}$. This is achieved by performing fixed gradient-based update steps to minimize the magnitude of any inequality violations:

$$\hat{w} = \tilde{w} - \gamma \nabla_w \| \max(0, g_v(\tilde{w})) \|_2^2. \tag{14}$$

By unrolling these iterations during training, the network learns to predict an initial z that requires minimal correction, effectively internalizing the feasible region's boundaries.

The overall training procedure begins with the initialization of the neural network parameters θ, defined as $N_\theta : \mathbb{R}^n \to \mathbb{R}^D$ where $D < d$. For each input v in the dataset $\mathcal{V}$, the model proposes a latent vector $z = N_\theta(v)$. This proposal is then completed into a full solution $\tilde{w} = \mathcal{T}(z) = [z, \phi_v(z)]^T$ via the equality projection. Subsequently, the solution is corrected toward inequality feasibility via $\hat{w} = \Pi(\tilde{w})$. A regularized loss $f(\hat{w} \mid v)$ is then computed, and the parameters θ are updated using the gradient $\nabla_\theta f(\hat{w} \mid v)$. This entire pipeline remains differentiable, allowing the model to optimize its weights based on the final, feasible output rather than a raw, unconstrained prediction.

A.3.2. U-Nets, Multi-ResNet and Conditioned ResNets

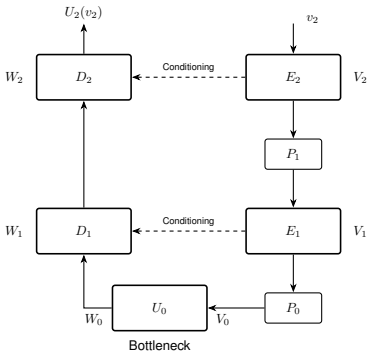

Figure 6. General U-Net (Ronneberger et al., 2015; Williams et al., 2023)

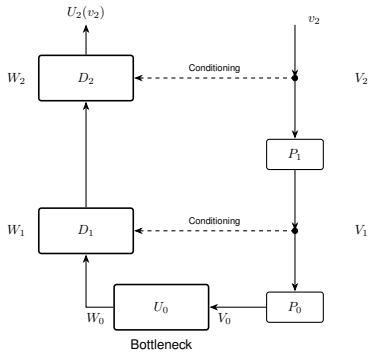

Figure 7. Multi-ResNet (U-Net specified with Identity Encoder)

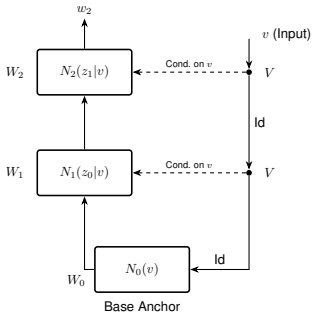

Figure 8. Conditioned ResNet (Multi-ResNet specified with identity projections)

Formally, this architecture instantiates a U-Net for low-rank preconditioning by fixing the encoder and projection operators to the identity map, thereby preserving the full input state at every refinement stage. This is the simplest form of U-Net. Currently we do not do any feature compression, but this would need further investigation.