GSNR: Graph Smooth Null-Space Representation for Inverse Problems

Supplementary Material

Appendix Contents

A1. Proof of Theorem 1

Theorem 1 (Coverage for graph smooth null-space) *Consider the construction of* $\mathbf{T}$ *Eq* (5)*, the covariance of the null-space* $\mathrm{Cov}(\mathbf{x}_n)$ *is a spectral function of* $\mathbf{T}$*, i.e.,* $\mathrm{Cov}(\mathbf{x}_n) = \mathbf{V}\mathrm{diag}(\lambda_1, \dots, \lambda_q)\mathbf{V}^\top$*, where* $\lambda_i = \frac{1}{\alpha\mu_i+\epsilon}$*,* $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_q$*. With the construction of* $\mathbf{S}$ *in* (6)*, the coverage of the null-space using GMRF with a Laplacian matrix* $\mathbf{L}$ *as* $C_L(p)$ *and the coverage when* $\mathbf{L} = \mathbf{I}$ *is denoted* $C_I(p)$*, satisfies for every* $p = 1, \dots, q$

$$C_L(p) \;\geq\; C_I(p) \tag{15}$$

Using cyclicity of the trace and $P(p) = \mathbf{V}_p\mathbf{V}_p^\top$,

$$\begin{aligned}\mathrm{tr}\big(P(p)\,\mathrm{Cov}\big) &= \mathrm{tr}\Big(\mathbf{V}_p\mathbf{V}_p^\top\,\mathbf{V}\,\mathrm{diag}(\lambda)\,\mathbf{V}^\top\Big)\\ &= \mathrm{tr}\Big((\mathbf{V}^\top\mathbf{V}_p)(\mathbf{V}^\top\mathbf{V}_p)^\top\,\mathrm{diag}(\lambda)\Big)\\ &= \mathrm{tr}\big(\mathrm{diag}(\lambda_1, \dots, \lambda_p)\big) = \sum_{i=1}^{p}\lambda_i.\end{aligned}$$

Also $\mathrm{tr}(\mathrm{Cov}) = \sum_{i=1}^{q}\lambda_i$. Hence

$$C_L(p) \;=\; \frac{\sum_{i=1}^{p}\lambda_i}{\sum_{i=1}^{q}\lambda_i}. \tag{16}$$

Consider the case $\mathbf{L} = \mathbf{I} \to \mathbf{T} = \mathbf{P}_n$ due to the symmetric and idempotent property of the null-space projector. Any symmetric and idempotent matrix is diagonalizable with eigenvalues only in $\{0, 1\}$. Then, any eigenvector and eigenvalue pair $(\mathbf{v}_i, \lambda_i)$ satisfy

$$\lambda_i = \begin{cases}1, & \text{if } \mathbf{v}_i \in \mathrm{Null}(\mathbf{H}),\\ 0, & \text{if } \mathbf{v}_i \in \mathrm{Range}(\mathbf{H}^\top).\end{cases}$$

Under this analysis, replacing the eigenvalues λ_i in (16), we have

$$C_I(p) = \frac{p}{q}. \tag{17}$$

Since (λ_i) is nonincreasing, the mean of the first p terms is at least the global mean:

$$\frac{1}{p}\sum_{i=1}^{p}\lambda_i \;\geq\; \frac{1}{q}\sum_{i=1}^{q}\lambda_i \quad\Longleftrightarrow\quad \frac{\sum_{i=1}^{p}\lambda_i}{\sum_{i=1}^{q}\lambda_i} \;\geq\; \frac{p}{q}.$$

Combining with (16)–(17) gives $C_L(p) \geq C_I(p)$ for all p, with strict inequality whenever (λ_i) is not constant.

Corollary 1 *Using* $\lambda_i = \frac{1}{\alpha\mu_i+\varepsilon}$*,* $\mu_1 \leq \cdots \leq \mu_q$*, and the bounds* $\sum_{i\leq p}\lambda_i \geq p\,\lambda_p$ *and* $\sum_{i\leq q}\lambda_i \leq p\,\lambda_1 + (q-p)\,\lambda_{p+1}$*, we obtain*

$$C_L(p) \geq \frac{p\,\lambda_p}{p\,\lambda_1 + (q-p)\,\lambda_{p+1}} \tag{18}$$

$$= \frac{1}{\frac{\alpha\mu_p+\varepsilon}{\alpha\mu_1+\varepsilon} + \frac{q-p}{p}\,\frac{\alpha\mu_p+\varepsilon}{\alpha\mu_{p+1}+\varepsilon}} \;\geq\; \frac{p}{q}, \tag{19}$$

with the last inequality again strict unless the spectrum is flat. This makes explicit how fast coverage rises for Laplacians whose $\{\mu_i\}$ increase rapidly (grids/graphs) compared to $\mathbf{L} = \mathbf{I}$.

A2. Proof of Theorem 2

Theorem 2 (Minimax optimality) *Let $V_p = \mathrm{span}\{\mathbf{v}_1, \dots, \mathbf{v}_p\}$ and P_{V_p} be its projector. Then, among all p-dimensional $V \subset \mathrm{Null}(\mathbf{H})$, provided $\mu_{p+1} > 0$,*

$$\min_{\dim(V)=p} \sup_{\mathbf{x}_n \in \mathcal{M}_\tau} \|(\mathbf{I} - P_V)\mathbf{x}_n\|_2^2 = \sup_{\mathbf{x}_n \in \mathcal{M}_\tau} \|(\mathbf{I} - P_{V_p})\mathbf{x}_n\|_2^2 = \frac{\tau}{\mu_{p+1}}.$$

Proof.
Expand any $\mathbf{x}_n \in \mathrm{Null}(\mathbf{H})$ in the eigenbasis of $\mathbf{T}$: $\mathbf{x}_n = \sum_{j=1}^{q} a_j \mathbf{v}_j$ with $\|\mathbf{x}_n\|_2^2 = \sum_j a_j^2$ and energy constraint $\mathbf{x}_n^\top \mathbf{T} \mathbf{x}_n = \sum_j \mu_j a_j^2 \le \tau$. This is the Rayleigh–Ritz parameterization for a symmetric PSD matrix. Take $V_p = \mathrm{span}\{\mathbf{v}_1, \dots, \mathbf{v}_p\}$. Then $(\mathbf{I} - P_{V_p})\mathbf{x}_n = \sum_{j \ge p+1} a_j \mathbf{v}_j$, hence

$$\|(\mathbf{I} - P_{V_p})\mathbf{x}_n\|_2^2 = \sum_{j \ge p+1} a_j^2 \le \frac{1}{\mu_{p+1}} \sum_{j \ge p+1} \mu_j a_j^2 \le \frac{\tau}{\mu_{p+1}}.$$

The first inequality uses $\mu_j \ge \mu_{p+1}$ for all $j \ge p+1$. Equality is attained by $\mathbf{x}_n^\star = \sqrt{\tau/\mu_{p+1}}\, \mathbf{v}_{p+1}$, so

$$\sup_{\mathbf{x}_n \in \mathcal{M}_\tau} \|(\mathbf{I} - P_{V_p})\mathbf{x}_n\|_2^2 = \frac{\tau}{\mu_{p+1}}. \qquad (*)$$

Let V be any p–dimensional subspace of $\mathrm{Null}(\mathbf{H})$ and set $W := V^\perp$ (within $\mathrm{Null}(\mathbf{H})$), so $\dim W = q - p$. Consider the restricted eigenproblem for $\mathbf{T}$ on W; denote its smallest eigenvalue by

$$\widetilde{\mu}_{\min}(W) = \min_{\mathbf{u} \in W,\ \|\mathbf{u}\|=1} \mathbf{u}^\top \mathbf{T} \mathbf{u}.$$

By the Courant–Fischer min–max theorem,

$$\mu_{p+1} = \max_{\dim(S)=q-p} \min_{\mathbf{u} \in S,\ \|\mathbf{u}\|=1} \mathbf{u}^\top \mathbf{T} \mathbf{u} \ge \min_{\mathbf{u} \in W,\ \|\mathbf{u}\|=1} \mathbf{u}^\top \mathbf{T} \mathbf{u} = \widetilde{\mu}_{\min}(W). \qquad (20)$$

Thus $\widetilde{\mu}_{\min}(W) \le \mu_{p+1}$.

Now maximize the residual under the energy constraint *within* W, where the projection vanishes: for any $\mathbf{x}_n \in W$, $(\mathbf{I} - P_V)\mathbf{x}_n = \mathbf{x}_n$. The maximizer aligns with the eigenvector of $\mathbf{T}|_W$ for $\widetilde{\mu}_{\min}(W)$, yielding

$$\sup_{\mathbf{x}_n \in \mathcal{M}_\tau} \|(\mathbf{I} - P_V)\mathbf{x}_n\|_2^2 \ge \sup_{\substack{\mathbf{x}_n \in W \\ \mathbf{x}_n^\top \mathbf{T} \mathbf{x}_n \le \tau}} \|\mathbf{x}_n\|_2^2 = \frac{\tau}{\widetilde{\mu}_{\min}(W)} \ge \frac{\tau}{\mu_{p+1}}. \qquad (\dagger)$$

Combining $(*)$ and $(\dagger)$ gives

$$\min_{\dim(V)=p} \sup_{\mathbf{x}_n \in \mathcal{M}_\tau} \|(\mathbf{I} - P_V)\mathbf{x}_n\|_2^2 = \frac{\tau}{\mu_{p+1}},$$

with the minimum attained at $V = V_p$.

Remark 3 *The constraint $\mathbf{x}_n^\top \mathbf{T} \mathbf{x}_n \le \tau$ is an ellipsoid aligned with the eigenvectors of $\mathbf{T}$. The largest Euclidean norm inside this ellipsoid occurs along the* smallest *eigenvalue direction. If the subspace V leaves a small-μ direction outside, an adversary can place all energy there and produce a large residual. Courant–Fischer formalizes this: every p–dimensional V leaves some direction with Rayleigh quotient $\le \mu_{p+1}$ in $V^\perp$.*

Selecting $V_p = \mathrm{span}\{\mathbf{v}_1, \dots, \mathbf{v}_p\}$ "covers" all directions with the smallest graph energy (smoothest modes). The worst direction you do not *cover is then $\mathbf{v}_{p+1}$ with energy μ_{p+1}, making the worst-case miss exactly τ/μ_{p+1}. Any other choice leaves an even smoother (smaller μ) direction uncovered, increasing the worst-case error.*

The bound quantifies how quickly the worst-case miss decays as p grows: the key driver is μ_{p+1}. For graph Laplacians, $\{\mu_j\}$ increases smoothly, so μ_{p+1} grows and the miss shrinks rapidly; for $\mathbf{L} = \mathbf{I}$, $\mathbf{T} = \mathbf{P}_n$ has a flat spectrum on $\mathrm{Null}(\mathbf{H})$, so μ_{p+1} is constant and the bound does not improve. This explains why graph-limited designs achieve much better null-space coverage than geometry-free design choices (the argument is the Kolmogorov n-width of the ellipsoid $\{\mathbf{x}_n : \mathbf{x}_n^\top \mathbf{T} \mathbf{x}_n \le \tau\}$).

A3. Proof of Proposition 1

Proposition 1 (Per–mode predictability bound) *Let $\mathbf{x} \in \mathbb{R}^n$ be zero–mean Gaussian with covariance $\mathbf{C}$ and precision $\mathbf{Q} = \mathbf{C}^{-1}$. Let $\mathbf{H} \in \mathbb{R}^{m \times n}$ and denote the orthogonal projectors onto $\mathcal{R} := \mathrm{Range}(\mathbf{H}^\top)$ and $\mathcal{N} := \mathrm{Null}(\mathbf{H})$ by $\mathbf{P}_r$ and $\mathbf{P}_n$, respectively. Block $\mathbf{Q}$ and $\mathbf{C}$ with respect to the decomposition $\mathbb{R}^n = \mathcal{R} \oplus \mathcal{N}$:*

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{rr} & \mathbf{Q}_{rn} \\ \mathbf{Q}_{nr} & \mathbf{Q}_{nn} \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} \mathbf{C}_{rr} & \mathbf{C}_{rn} \\ \mathbf{C}_{nr} & \mathbf{C}_{nn} \end{bmatrix}.$$

Assume $\mathbf{Q}_{nn} \succ 0$ on $\mathcal{N}$ and let $\{\mathbf{v}_j\}$ be an orthonormal eigenbasis of $\mathbf{Q}_{nn}$ with $\mathbf{Q}_{nn}\mathbf{v}_j = \mu_j \mathbf{v}_j$ and $\mu_j > 0$. Define the j-th null coefficient $a_j := \mathbf{v}_j^\top \mathbf{x}_n$ where $\mathbf{x}_n := \mathbf{P}_n \mathbf{x}$. Consider measurements from (1) and denote $\mathbf{C}_y := \mathbf{H}\mathbf{C}_{rr}\mathbf{H}^\top + \sigma^2 \mathbf{I}_m$. Then the population R^2 of the optimal linear predictor of a_j from $\mathbf{y}$ satisfies

$$\rho_j^2 := \frac{\mathrm{Cov}(a_j, \mathbf{y})^\top\, \mathbf{C}_y^{-1}\, \mathrm{Cov}(\mathbf{y}, a_j)}{\mathrm{Var}(a_j)} \le \frac{c_j}{c_j + \mu_j},$$

$$c_j := \mathbf{v}_j^\top \big(\mathbf{Q}_{rn}\mathbf{C}_{rr}\mathbf{Q}_{nr}\big)\mathbf{v}_j.$$

Equality holds in the ideal case $\mathbf{H} = \mathbf{I}$ and $\sigma^2 = 0$.

Since $\mathbf{Q} \succ 0$ and $\mathbf{Q}_{nn} \succ 0$, the block inverse of $\mathbf{Q}$ exists and yields the standard formulas (Schur complements)

$$\begin{aligned}\mathbf{C}_{nr} &= -\mathbf{Q}_{nn}^{-1}\mathbf{Q}_{nr}\mathbf{C}_{rr},\\ \mathbf{C}_{nn} &= \mathbf{Q}_{nn}^{-1} + \mathbf{Q}_{nn}^{-1}\mathbf{Q}_{nr}\mathbf{C}_{rr}\mathbf{Q}_{rn}\mathbf{Q}_{nn}^{-1}.\end{aligned} \tag{21}$$

They follow from

$$\mathbf{Q}^{-1} = \begin{bmatrix} \cdot & -(\mathbf{Q}_{rr}-\mathbf{Q}_{rn}\mathbf{Q}_{nn}^{-1}\mathbf{Q}_{nr})^{-1}\mathbf{Q}_{rn}\mathbf{Q}_{nn}^{-1} \\ -\mathbf{Q}_{nn}^{-1}\mathbf{Q}_{nr}(\cdot) & \cdot \end{bmatrix}$$

.

By definition $a_j = \mathbf{v}_j^\top \mathbf{x}_n$ and $\mathbf{y} = \mathbf{H}\mathbf{x}_r + \boldsymbol{\omega}$ with $\boldsymbol{\omega} \perp \mathbf{x}$. Hence

$$\mathrm{Cov}(a_j, \mathbf{y}) = \mathbf{v}_j^\top \mathrm{Cov}(\mathbf{x}_n, \mathbf{H}\mathbf{x}_r) = \mathbf{v}_j^\top \mathbf{C}_{nr}\mathbf{H}^\top. \tag{22}$$

Using (21), this becomes

$$\mathrm{Cov}(a_j, \mathbf{y}) = -\,\mathbf{v}_j^\top \mathbf{Q}_{nn}^{-1}\mathbf{Q}_{nr}\mathbf{C}_{rr}\,\mathbf{H}^\top.$$

Therefore the *numerator* of ρ_j^2 is

$$\begin{aligned}\mathrm{Num} &= \mathrm{Cov}(a_j, \mathbf{y})^\top \mathbf{C}_y^{-1}\,\mathrm{Cov}(a_j, \mathbf{y})\\ &= \mathbf{H}\mathbf{C}_{rr}\mathbf{Q}_{rn}\mathbf{Q}_{nn}^{-1}\mathbf{v}_j \;\cdot\; \mathbf{C}_y^{-1} \;\cdot\; \mathbf{v}_j^\top \mathbf{Q}_{nn}^{-1}\mathbf{Q}_{nr}\mathbf{C}_{rr}\mathbf{H}^\top.\end{aligned} \tag{23}$$

Introduce $\mathbf{A} := \mathbf{H}\mathbf{C}_{rr}^{1/2}$ and note $\mathbf{C}_y = \mathbf{A}\mathbf{A}^\top + \sigma^2\mathbf{I}$. Define also

$$\mathbf{M} := \mathbf{C}_{rr}^{1/2}\mathbf{H}^\top\mathbf{C}_y^{-1}\mathbf{H}\mathbf{C}_{rr}^{1/2} = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \sigma^2\mathbf{I})^{-1}\mathbf{A}.$$

Let $\mathbf{w}_j := \mathbf{C}_{rr}^{1/2}\mathbf{Q}_{rn}\mathbf{Q}_{nn}^{-1}\mathbf{v}_j$. Then (23) rewrites compactly as

$$\mathrm{Num} = \mathbf{w}_j^\top \mathbf{M}\,\mathbf{w}_j. \tag{24}$$

Take an SVD $\mathbf{A} = \mathbf{V}\mathbf{S}\mathbf{V}^\top$, $\mathbf{S} = \mathrm{diag}(s_i) \geq 0$. Then

$$\mathbf{M} = \mathbf{V}\mathbf{S}\left(\mathbf{S}^2+\sigma^2\mathbf{I}\right)^{-1}\mathbf{S}\mathbf{V}^\top = \mathbf{V}\;\mathrm{diag}\Big(\frac{s_i^2}{s_i^2+\sigma^2}\Big)\mathbf{V}^\top \;\preceq\; \mathbf{I},$$

with equality iff $\sigma^2 = 0$ (and then $\mathbf{M} = \mathbf{I}$). Consequently,

$$\mathrm{Num} = \mathbf{w}_j^\top \mathbf{M}\mathbf{w}_j \;\leq\; \mathbf{w}_j^\top \mathbf{w}_j = \mathbf{v}_j^\top \mathbf{Q}_{nn}^{-1}\mathbf{Q}_{nr}\mathbf{C}_{rr}\mathbf{Q}_{rn}\mathbf{Q}_{nn}^{-1}\mathbf{v}_j. \tag{25}$$

Using (21),

$$\begin{aligned}\mathrm{Den} := \mathrm{Var}(a_j) &= \mathbf{v}_j^\top \mathbf{C}_{nn}\mathbf{v}_j\\ &= \mathbf{v}_j^\top \mathbf{Q}_{nn}^{-1}\mathbf{v}_j + \mathbf{v}_j^\top \mathbf{Q}_{nn}^{-1}\mathbf{Q}_{nr}\mathbf{C}_{rr}\mathbf{Q}_{rn}\mathbf{Q}_{nn}^{-1}\mathbf{v}_j.\end{aligned} \tag{26}$$

Since $\mathbf{Q}_{nn}\mathbf{v}_j = \mu_j\mathbf{v}_j$ and $\|\mathbf{v}_j\|_2 = 1$, we have

$$\mathbf{Q}_{nn}^{-1}\mathbf{v}_j = \mu_j^{-1}\mathbf{v}_j, \qquad \mathbf{v}_j^\top \mathbf{Q}_{nn}^{-1}\mathbf{v}_j = \mu_j^{-1}.$$

Define the scalar

$$c_j := \mathbf{v}_j^\top\big(\mathbf{Q}_{rn}\mathbf{C}_{rr}\mathbf{Q}_{nr}\big)\mathbf{v}_j \;\geq 0,$$

and observe that

$$\mathbf{v}_j^\top \mathbf{Q}_{nn}^{-1}\mathbf{Q}_{nr}\mathbf{C}_{rr}\mathbf{Q}_{rn}\mathbf{Q}_{nn}^{-1}\mathbf{v}_j = \mu_j^{-2}\,c_j.$$

Plugging into (26) gives

$$\mathrm{Den} = \mu_j^{-1} + \mu_j^{-2}\,c_j. \tag{27}$$

From (25) and (27),

$$\rho_j^2 = \frac{\mathrm{Num}}{\mathrm{Den}} \;\leq\; \frac{\mu_j^{-2}c_j}{\mu_j^{-1}+\mu_j^{-2}c_j} = \frac{c_j}{c_j+\mu_j}.$$

If $\sigma^2 = 0$ and $\mathbf{H} = \mathbf{I}$, then $\mathbf{C}_y = \mathbf{C}_{rr}$ and hence $\mathbf{M} = \mathbf{I}$, so inequality (25) is an equality and $\rho_j^2 = \frac{c_j}{c_j+\mu_j}$.

A4. Graph structures

A4.1. Additional graphs

We analyze two additional graph topologies.

Random-walk normalized graph ($\mathbf{L}_{\mathrm{rw}}$). Starting from an undirected graph with adjacency $\mathbf{A}$ and degree matrix $\mathbf{D}$, The random-walk normalized Laplacian is

$$\mathbf{L}_{\mathrm{rw}} \;=\; \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}.$$

It can be read as $\mathbf{L}_{\mathrm{rw}} = \mathbf{I} - \mathbf{P}$, where $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ is the one-step Markov transition; hence $\mathbf{L}_{\mathrm{rw}}\mathbf{1} = \mathbf{0}$ and the spectrum reflects how probability mass mixes locally. This normalization corrects for degree imbalance: edges from low-degree nodes are not underweighted, so the quadratic form $\mathbf{x}^\top\mathbf{L}_{\mathrm{rw}}\mathbf{x}$ penalizes variations relative to local node degree. In our pipeline, we never use $\mathbf{L}_{\mathrm{rw}}$ directly; we act on the null component through the null-restricted operator $\mathbf{T} = \mathbf{P}_n\mathbf{L}_{\mathrm{rw}}\mathbf{P}_n$ and compute its smooth eigenmodes to build $\mathbf{S}$.

Symmetric normalized graph ($\mathbf{L}_{\mathrm{sym}}$). The symmetric normalized Laplacian is

$$\mathbf{L}_{\mathrm{sym}} \;=\; \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\,\mathbf{D}^{-1/2},$$

which is symmetric positive semidefinite and yields the energy $\mathbf{x}^\top\mathbf{L}_{\mathrm{sym}}\mathbf{x} = \frac{1}{2}\sum_{i,j} w_{ij}\big(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}}\big)^2$. It equalizes contributions across nodes with different degrees and, on regular grids, is similar to $\mathbf{L}_{\mathrm{rw}}$ (hence they share eigenvalues up to a similarity transform). Within GSNR, we apply the same null restriction $\mathbf{T} = \mathbf{P}_n\mathbf{L}_{\mathrm{sym}}\mathbf{P}_n$ and take its p smallest eigenpairs to span the graph-smooth null subspace used by $\mathbf{S}$. In practice, both normalizations deliver the intended effect, concentrating smooth, predictable variation in $\mathrm{Null}(\mathbf{H})$, with small empirical differences that reflect how degree reweighting shapes the spectrum of $\mathbf{T}$.

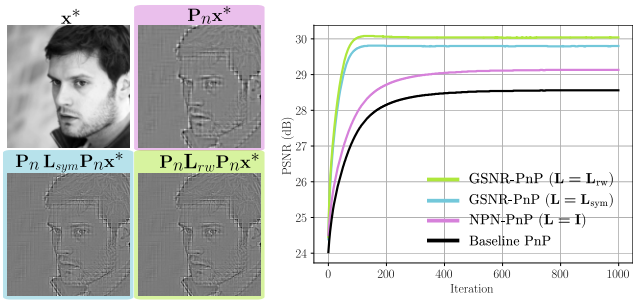

Figure 8. SR with different graph Laplacians. Left: ground truth $\mathbf{x}^*$, its null component $\mathbf{P}_n\mathbf{x}^*$, and graph-smoothed null responses $\mathbf{P}_n\mathbf{L}_{\text{sym}}\mathbf{P}_n\mathbf{x}^*$ and $\mathbf{P}_n\mathbf{L}_{\text{rw}}\mathbf{P}_n\mathbf{x}^*$. Right: PSNR vs. iteration for GSNR-PnP with $\mathbf{L} \in \{\mathbf{L}_{\text{sym}}, \mathbf{L}_{\text{rw}}, \mathbf{I}\}$, and the PnP baseline.

In Figure 8, the left panels visualize the invisible component and its graph–smoothed variants: $\mathbf{P}_n\mathbf{x}^*$ (bottom–left) highlights edges and fine textures that lie in the null space of the sensing operator, while applying the null–restricted Laplacians $\mathbf{P}_n\mathbf{L}_{\text{sym}}\mathbf{P}_n$ and $\mathbf{P}_n\mathbf{L}_{\text{rw}}\mathbf{P}_n$ further emphasizes coherent edge geometry and suppresses isolated, high–frequency speckles. The two normalizations yield very similar structures, with $\mathbf{L}_{\text{rw}}$ marginally enhancing sharp contours, consistent with its degree-normalized reweighting. This visual evidence aligns with the goal of GSNR: to impose structure *only* in the null space and make its content smoother and more predictable.

The PSNR–iteration curves (right) show that graph–limited null–space designs (GSNR–PnP with $\mathbf{L}_{\text{sym}}$ or $\mathbf{L}_{\text{rw}}$) both *converge faster* and *stabilize at higher PSNR* than the geometry–free alternative ($\mathbf{L} = \mathbf{I}$) and the PnP baseline. Among the graph choices, the $\mathbf{L}_{\text{rw}}$ variant exhibits the steepest initial rise and the highest plateau, while $\mathbf{L}_{\text{sym}}$ tracks closely behind; both consistently dominate $\mathbf{L} = \mathbf{I}$. This behavior matches the theory: normalized Laplacians produce a more informative spectrum for the null–restricted operator $\mathbf{T} = \mathbf{P}_n\mathbf{L}\mathbf{P}_n$, yielding higher spectral coverage and stronger statistical coupling, which in turn improves both the fixed point (final PSNR) and the transient (speed of convergence).

A4.2. Learning a Structured Laplacian

The learning objective maximizes the average coverage over a prescribed set $\mathcal{P}$ (typically 10 equispaced values):

$$\max_{\mathbf{L}_\theta \succeq 0} \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} C(p),$$

using $\mathbf{T} = \mathbf{P}_n\mathbf{L}_\theta\mathbf{P}_n$ for the creation of $\mathbf{S}$ as in Sec. 3.2. Each parametrization enforces $\mathbf{L}_\theta \succeq 0$ via a symmetric construction plus a small $\varepsilon\mathbf{I}$ to keep eigenvalues nonnegative.

Dense (low-rank PSD). Learn $\mathbf{\Theta} \in \mathbb{R}^{n\times r}$:

$$\mathbf{L}_\theta = \mathbf{\Theta}\,\mathbf{\Theta}^\top + \varepsilon\,\mathbf{I}.$$

Diagonal. Let $\boldsymbol{\theta} \in \mathbb{R}^n$ be learnable logits and define $\text{softplus}(t) \triangleq \log\big(1 + e^t\big)$. Set

$$\mathbf{d} = \text{softplus}(\boldsymbol{\theta}) + \varepsilon, \qquad \mathbf{L}_\theta = \text{diag}(\mathbf{d}).$$

Circulant (wrap-around convolution). Learn a kernel $\mathbf{\Theta} \in \mathbb{R}^{E\times E}$; let $\mathbf{C}_{\text{circ}}(\mathbf{\Theta}) \in \mathbb{R}^{n\times n}$ denote the circular (wrap-around) convolution operator on vectorized images. Then

$$\mathbf{L}_\theta = \mathbf{C}_{\text{circ}}(\mathbf{\Theta})^\top\,\mathbf{C}_{\text{circ}}(\mathbf{\Theta}) + \varepsilon\,\mathbf{I}.$$

Block-diagonal. Partition n into $b = n/B$ blocks of size B. Learn $\mathbf{\Theta}_i \in \mathbb{R}^{B\times B}$ for $i = 1, \ldots, b$. With

$$\text{blockdiag}(\mathbf{A}_1, \ldots, \mathbf{A}_b) \triangleq \begin{bmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_b \end{bmatrix},$$

we set

$$\mathbf{L}_\theta = \text{blockdiag}\big(\mathbf{\Theta}_1\mathbf{\Theta}_1^\top + \varepsilon\mathbf{I}_B,\ \ldots,\ \mathbf{\Theta}_b\mathbf{\Theta}_b^\top + \varepsilon\mathbf{I}_B\big).$$

Fig. 9 shows the comparison of a) normalized $\mathbf{T}$-eigenvalues and b) $\mathbf{S}$-coverage for the different parametrizations of $\mathbf{L}_\theta$. Using a Laplacian graph such as $\mathbf{L}_{8nn}$ in GSNR yields the best spectral performance and coverage for any value of p. Although NPN has an adequate spectrum, its coverage is very limited, whereas the opposite is true for block-diag.

A5. Settings for construction null-restricted Laplacian in practice

We never materialize $\mathbf{P}_n$ or $\mathbf{T}$ as dense matrices. Instead, $\mathbf{P}_n\mathbf{v} = \mathbf{v} - \mathbf{H}^\top(\mathbf{H}\mathbf{H}^\top)^{-1}\mathbf{H}\mathbf{v}$ is exposed as a callable projector using a factorization of $\mathbf{H}\mathbf{H}^\top$, and $\mathbf{T}$ is wrapped as a SciPy `LinearOperator` that applies $\mathbf{T}\mathbf{x} = \mathbf{P}_n\big(\mathbf{L}\,\mathbf{P}_n\mathbf{x}\big)$ on the fly. We then invoke ARPACK [24] via `eigsh` with spectral method to extract the k smallest–magnitude eigenpairs of $\mathbf{T}$ (the smoothest graph–null modes), respecting the constraint $k \le n$ and defaulting to $k = \min(q, n-1)$ when unspecified. The routine returns eigenvectors as columns $\mathbf{U} \in \mathbb{R}^{n\times k}$ and eigenvalues $\{\mu_j\}$; we set $\mathbf{S}_{\text{full}} = \mathbf{U}^\top$ so that the rows of $\mathbf{S}$ form an orthonormal basis of the selected null–space subspace, and finally truncate to the first p rows for training/inference. The process is performed for different graph Laplacians $\mathbf{L}$, and the structure selection is guided by our spectral criteria, i.e., maximum coverage/predictability over the first p modes. The *demosaicing* case follows the same flow but loads a precomputed sparse $\mathbf{H}$, optionally

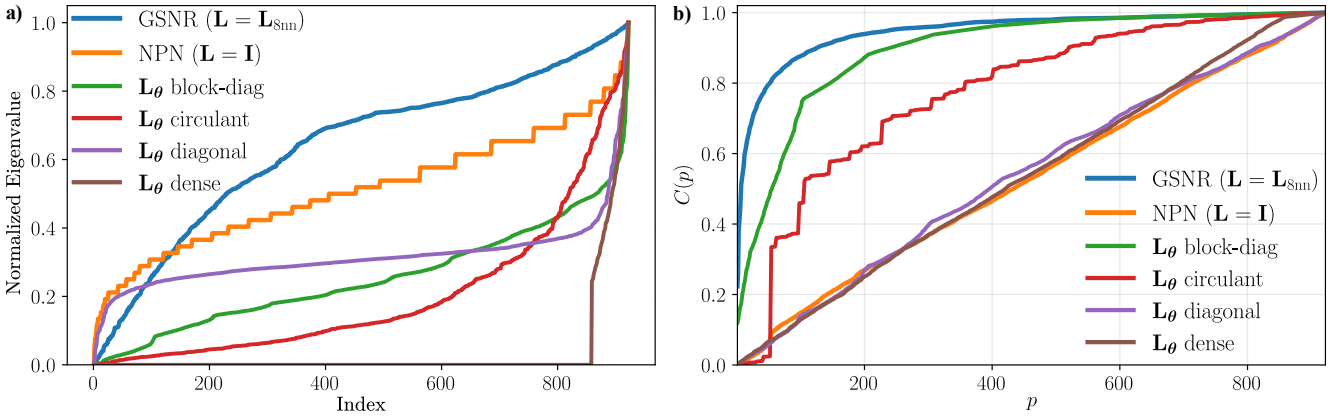

Figure 9. **(a)** Variation of $\mathbf{T}$ normalized eigenvalues for $\mathbf{L}_\theta$ with respect to their index in CS. **(b)** Coverage of $\mathbf{S}$ with $\mathbf{L}_\theta$ in CS.

lifts the Laplacian to multi–channel form with a Kronecker product $\mathbf{L} \leftarrow \mathbf{I}_C \otimes \mathbf{L}$.

For the numerical results of this work, we empirically set the dimension p. However, our framework provides a principled evaluation on how to select p based on the null-space coverage. Let $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_q)$ be the eigenvalues of $\mathrm{Cov}(\mathbf{x}_n)$ in the graph-smooth basis, ordered so that $\lambda_1 \geq \cdots \geq \lambda_q > 0$. Define the cumulative coverage $C(p) = \big(\sum_{i=1}^{p} \lambda_i\big)/\big(\sum_{i=1}^{q} \lambda_i\big)$. We select the effective dimension $p^\star$ as the smallest p that simultaneously achieves a target coverage level and lies on a plateau of the coverage curve, following Algorithm 1.

Algorithm 1 Coverage-based automatic selection of p

Require: Eigenvalues $\lambda_1 \geq \cdots \geq \lambda_q > 0$; coverage target $\kappa \in (0, 1)$ (e.g. $\kappa = 0.95$); slope tolerance $\delta > 0$ (e.g. $\delta = 10^{-3}$); plateau length $L \in \mathbb{N}$ (e.g. $L = 10$).

Compute total variance $S \leftarrow \sum_{i=1}^{q} \lambda_i$.
For $p = 1, \dots, q$, compute coverage $C(p) \leftarrow \big(\sum_{i=1}^{p} \lambda_i\big)/S$.
For $p = 1, \dots, q$, compute incremental gains $\Delta C(p) \leftarrow C(p) - C(p-1)$, with $C(0) \equiv 0$.
for $p = 1, \dots, q$ **do**
  if $C(p) \geq \kappa$ **then**
    Check plateau condition: $\max\{\Delta C(p), \dots, \Delta C(\min(p + L - 1, q))\} \leq \delta$.
    if plateau condition holds **then**
      return $p^\star \leftarrow p$.
    end if
  end if
end for
If no p satisfies the above, set $p^\star \leftarrow q$ (use all modes).

A6. Ablation Studies of the Graph Regularizer

Recall that GSNR is *solver-agnostic*: it augments the data-fidelity objective with the terms $\gamma\|\mathrm{G}^*(\mathbf{y}) - \mathbf{S}\mathbf{x}\|_2^2 + \frac{\gamma_g}{2}\mathbf{x}^\top\mathbf{T}\mathbf{x}$. Consequently, GSNR can be incorporated into

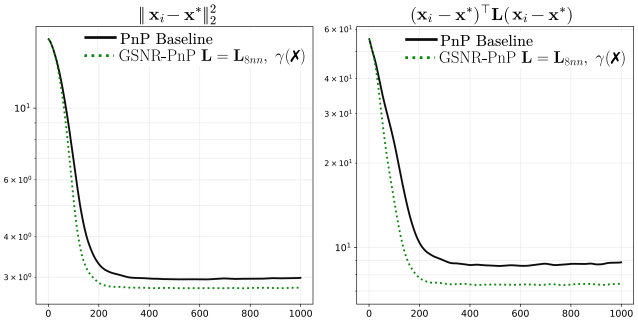

Figure 10. PnP variant with null-only projector regularizer: illustrative convergence.

any iterative solver, e.g., ADMM, HQS, FISTA, PD, or diffusion-based methods (see A8), by including these terms in its $\mathbf{x}$-update. Before showing further experiments, we describe the baseline algorithms and the GSNR versions.

Plug-and-Play. In this case, we used the PGD-PnP method. Algorithm 2 shows the GSNR PnP-PGD modification.

Algorithm 2 GSNR PnP-PGD with null-space and graph regularization

Require: $K, \mathbf{H}, \mathbf{y}, \alpha, \eta, \gamma, \gamma_g, \mathrm{G}^*, \mathbf{S}$

$\mathbf{x}_0 = \mathbf{H}^\top\mathbf{y} + \mathbf{S}^\top\mathrm{G}^*(\mathbf{y})$
for $i = 1, \dots, K$ **do**

$$\mathbf{x}_i \leftarrow \mathbf{x}_{i-1} - \alpha\Big(\mathbf{H}^\top(\mathbf{H}\mathbf{x}_{i-1} - \mathbf{y}) + \gamma\mathbf{S}^\top(\mathbf{S}\mathbf{x}_{i-1} - \mathrm{G}^*(\mathbf{y})) + \gamma_g\mathbf{T}\mathbf{x}_{i-1}\Big)$$

  $\mathbf{x}_i \leftarrow \mathrm{D}_\eta(\mathbf{x}_i)$
end for
return $\mathbf{x}_i$

A6.1. Why graph smooth null-space?

An initial test that shows us the usefulness of the null-space Laplacian is to start from the assumption that the reconstruction error with respect to the ground-truth, $\mathbf{x}_n^{(i)} = \mathbf{x}_i - \mathbf{x}^*$ is the null-space. Fig. 10 shows the convergence

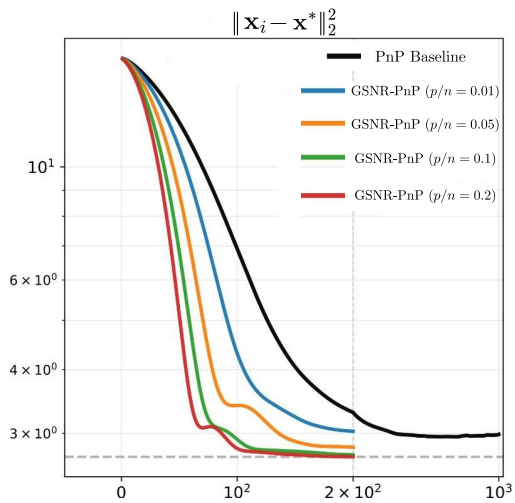

Figure 11. Low-dimensional null-space.

of the reconstruction and how the null-laplacian regularizer behaves. The baseline (black) shows that improving the quality of our reconstruction also reduces the null-laplacian error $(\mathbf{x}_n^{(i)})^\top \mathbf{L}(\mathbf{x}_n^{(i)})$, even though this term is not considered in the PnP cost function. Adding this term (green) would further improve the reconstruction. This demonstrates the usefulness of including a regularizer that promotes a graph-limited null-space.

A6.2. Why low-dimensional null-space projections?

Fig. 11 shows the convergence of the reconstruction when using different values of p. It can be seen that as the value of p increases, the reconstruction error decreases; however, this reaches a limit, since from $p/n = 0.1$ onward, the gain decreases. These results justify the need to use low-dimensional null-space projections, as they enable higher-quality reconstruction without being significantly affected by the decrease in predictability.

A6.3. Cost-function ablation

Figure 12 highlights two effects of the GSNR design: improved conditioning through the null-only graph regularizer and sensitivity to the choice of Laplacian in demosaicing problem. All GSNR-PnP variants eventually reach a similar high PSNR plateau, slightly above the PnP baseline, showing that incorporating the graph-smooth null-space prior does not harm the final reconstruction quality and can modestly improve it. However, the convergence speed differs significantly: when γ_{g} is active, GSNR-PnP with $\mathbf{L}_{4\mathrm{nn}}$ or $\mathbf{L}_{8\mathrm{nn}}$ reaches its peak PSNR in far fewer iterations than both the geometry-free $\mathbf{L} = \mathbf{I}$ case and the baseline.

When the graph regularizer is turned off, the graph-based methods still outperform the baseline but converge more slowly, with trajectories that are closer to the standard PnP. This confirms the theoretical prediction that the null-only

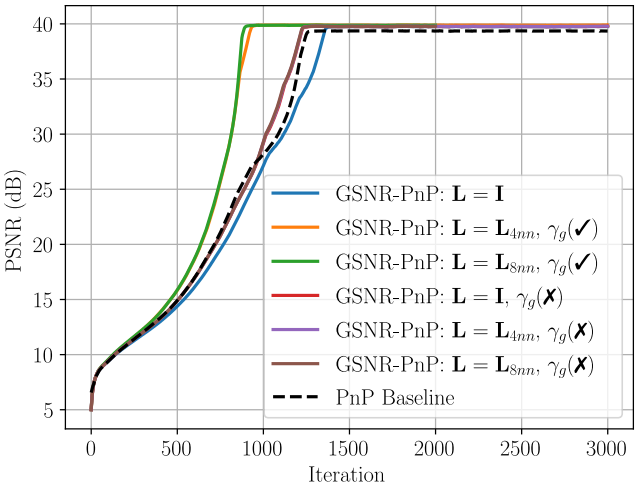

Figure 12. Effect of the null-only graph regularizer on GSNR-PnP convergence for demosaicing. We plot PSNR versus iteration for GSNR-PnP with different Laplacians ($\mathbf{L} = \mathbf{I}, \mathbf{L}_{4\mathrm{nn}}, \mathbf{L}_{8\mathrm{nn}}$) and with the graph-regularization weight γ_{g} either enabled or disabled, along with the standard PnP baseline (dashed).

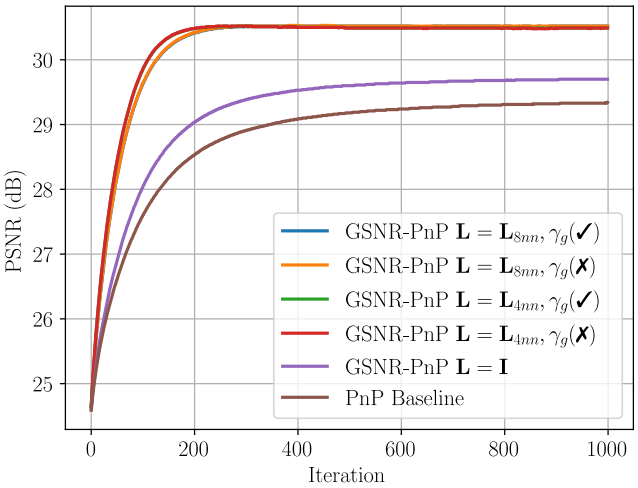

Figure 13. Effect of the null-only graph regularizer on GSNR-PnP convergence for super-resolution. We plot PSNR versus iteration for GSNR-PnP with different Laplacians ($\mathbf{L} = \mathbf{I}, \mathbf{L}_{4\mathrm{nn}}, \mathbf{L}_{8\mathrm{nn}}$) and with the graph-regularization weight γ_{g} either enabled or disabled, along with the standard PnP baseline (dashed).

graph term improves the spectrum of the normal matrix, effectively "lifting" the null directions, while the GSNR basis itself governs the final achievable PSNR. In practice, combining a graph Laplacian with a nonzero γ_{g} yields the best trade-off: fast convergence to high-quality solutions with minimal overhead in the PnP update. Similar analysis and results are shown in Fig. 13, for the image super-resolution problem, where the graph regularizer slightly increases convergence speed.

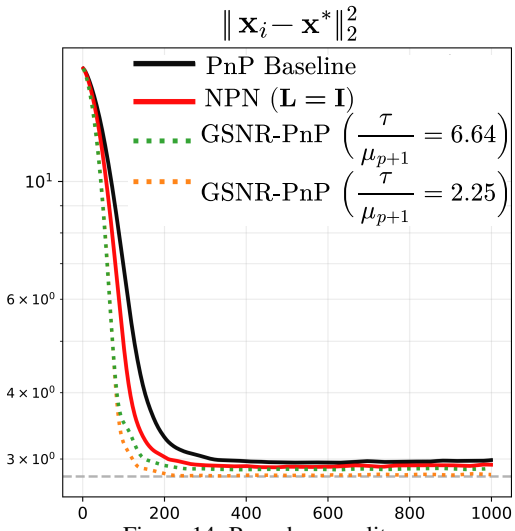

Figure 14. Bound vs. quality.

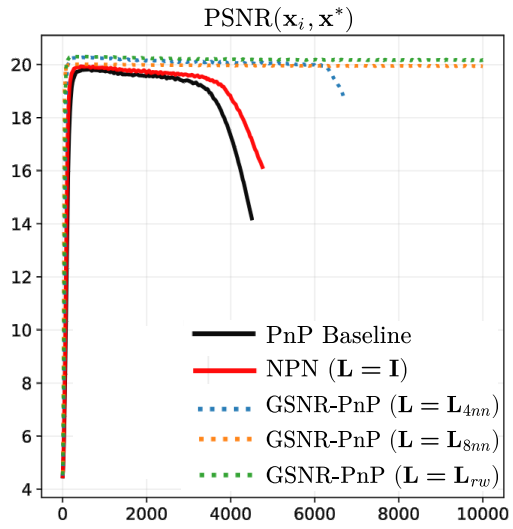

Figure 15. Fixed-point convergence.

A6.4. Minimax optimality bound

To experimentally validate the theory of Theorem 2, two different operators $\mathbf{S}$ were tested with $p = 0.1n$ and $\tau = 1$ for Fig. 14. In the green and yellow cases, there are two extreme values of the bound $\frac{\tau}{\mu_{p+1}}$ that demonstrate the theorem's postulate, since a $\mathbf{S}$ with a lower bound (2.25) results in better reconstruction.

A6.5. Fixed-point convergence

When analyzing an extreme case in Fig. 15, it can be observed that the proposed GSNR method (dotted lines) converges to a fixed point without diverging as iterations progress, achieving even faster convergence than NPN and the baseline.

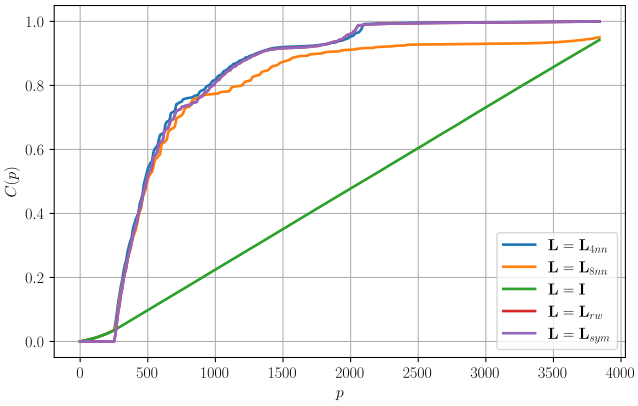

Figure 16. Spectral coverage curves $C(p)$ for super-resolution, comparing different Laplacian choices in the GSNR construction: grid $\mathbf{L}_{4\text{nn}}$, grid $\mathbf{L}_{8\text{nn}}$, random-walk normalized $\mathbf{L}_{\text{rw}}$, symmetric normalized $\mathbf{L}_{\text{sym}}$, and the geometry-free baseline $\mathbf{L} = \mathbf{I}$. Coverage $C(p)$ is the fraction of null-space variance captured by the first p graph-smooth modes.

A7. Coverage curve

Figure 16 shows that graph-based Laplacians concentrate null-space variance into a small number of modes. For $\mathbf{L}_{4\text{nn}}$, $\mathbf{L}_{\text{rw}}$, and $\mathbf{L}_{\text{sym}}$, the coverage rises steeply and reaches almost full variance with only a fraction of the null-space dimension, whereas the identity Laplacian $\mathbf{L} = \mathbf{I}$ exhibits an almost perfectly linear curve $C(p) \approx p/q$, meaning that coverage grows only proportionally to the dimension, and no "early" compression occurs. The $\mathbf{L}_{8\text{nn}}$ graph still offers a substantial advantage over $\mathbf{L} = \mathbf{I}$. But its curve saturates below the others, indicating slightly less concentrated variance. Overall, These results confirm the theoretical prediction that graph-smooth null modes provide much better coverage than geometry-free bases: a relatively small p already captures most null-space energy for the normalized and grid Laplacians, while the identity requires many more modes to achieve

A8. GSNR inclusion in Diffusion-based solvers

A8.1. DPS [8].

We denote K is the number of reverse diffusion steps, and $i \in \{0, \dots, K-1\}$ is the reverse-time index; $\mathbf{x}_i \in \mathbb{R}^n$ is the current latent state and $\mathbf{x}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the Gaussian start; $\hat{\mathbf{s}} = \mathbf{s}_\theta(\mathbf{x}_i, i)$ is the score/noise estimate produced by the network with parameters θ; $\hat{\mathbf{x}}_0$ is the network's prediction of the clean sample at step i; $\alpha_i \in (0, 1]$ is the per-step retention factor, $\beta_i = 1 - \alpha_i$ is the noise increment, and $\bar{\alpha}_i = \prod_{j=1}^{i} \alpha_j$ is the cumulative product (with $\bar{\alpha}_0 = 1$); $\zeta_i > 0$ is the data-consistency step size and $\tilde{\sigma}_i \geq 0$ is the sampling noise scale at step i; $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is i.i.d. Gaussian noise; $\mathbf{x}'_{i-1}$ denotes the pre–data-consistency iterate before applying the gradient correction. In the GSNR version, we further introduce a learned null-space predictor

Algorithm 3 GSNR–DPS Sampling with null-space and graph regularization

Require: K, $\mathbf{H}$, $\mathbf{y}$, $\{\zeta_i\}_{i=1}^K$, $\{\tilde{\sigma}_i\}_{i=1}^K$, γ, γ_{g}, $\mathbf{S}$, G^*, $\mathbf{P}_n$, $\mathbf{L}$

$\mathbf{x}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = K-1, \ldots, 0$ **do**
  $\hat{\mathbf{s}} \leftarrow \mathbf{s}_\theta(\mathbf{x}_i, i)$
  $\hat{\mathbf{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\mathbf{x}_i + (1-\bar{\alpha}_i)\hat{\mathbf{s}})$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\mathbf{x}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\mathbf{x}}_0 + \tilde{\sigma}_i \mathbf{z}$
  $\mathbf{x}_{i-1} \leftarrow \mathbf{x}'_{i-1} - \zeta_i \nabla_{\mathbf{x}_i}\left[\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_0\|_2^2 + \gamma\|\mathrm{G}^*(\mathbf{y}) - \mathbf{S}\hat{\mathbf{x}}_0\|_2^2 + \gamma_{\mathrm{g}}\|\mathbf{P}_n\mathbf{L}\hat{\mathbf{x}}_0\|_2^2\right]$
end for
return $\hat{\mathbf{x}}_0$

$\mathrm{G}(\mathbf{y}) \approx \mathbf{S}\mathbf{x}^*$ and a weight $\gamma > 0$ for the graph-smooth null-space penalty $\|\mathrm{G}(\mathbf{y}) - \mathbf{S}\hat{\mathbf{x}}_0\|_2^2$. Algorithm 3 shows the integration of GSNR in DPS.

A8.2. DiffPIR [55].

$\sigma_n > 0$ denotes the standard deviation of the measurement noise, and $\eta > 0$ is the data–proximal penalty that trades off data fidelity and the denoiser prior inside the subproblem; $\rho_i \triangleq \eta\,\sigma_n^2/\tilde{\sigma}_i^2$ is the iteration-dependent weight used in the proximal objective at step i; $\tilde{\mathbf{x}}_0^{(i)}$ is the score-model denoised prediction of the clean sample at step i (before enforcing data consistency); $\hat{\mathbf{x}}_0^{(i)}$ is the solution of the data-proximal subproblem at step i; $\hat{\boldsymbol{\epsilon}} = \big(1-\alpha_i\big)^{-1/2}\big(\mathbf{x}_i - \sqrt{\bar{\alpha}_i}\,\hat{\mathbf{x}}_0^{(i)}\big)$ is the effective noise estimate implied by $(\mathbf{x}_i, \hat{\mathbf{x}}_0^{(i)})$; $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the fresh Gaussian noise injected at step i; and $\zeta \in [0,1]$ mixes deterministic and stochastic updates ($\zeta = 0$ fully deterministic, $\zeta = 1$ fully stochastic). In the GSNR variant, we again use a null-space predictor $\mathrm{G}(\mathbf{y})$ and a weight $\gamma > 0$ to bias the proximal subproblem towards graph-smooth null-space coefficients. In Algorithm 4, we show the GSNR modification of DiffPIR.

In addition to the DPS / DiffPIR variables, GSNR uses: $\mathbf{S} \in \mathbb{R}^{p\times n}$, the graph-smooth null-space projector; $\mathrm{G}^*(\mathbf{y}) \approx \mathbf{S}\mathbf{x}^*$, a learned predictor of the target null coefficients; $\gamma > 0$, the weight of the null-space matching term; $\gamma_{\mathrm{g}} > 0$, the weight of the null-only graph regularizer; and $\mathbf{P}_n$ and $\mathbf{L}$, the null projector and graph Laplacian, respectively. The GSNR prior is

$$\gamma\big\|\mathrm{G}^*(\mathbf{y}) - \mathbf{S}\hat{\mathbf{x}}_0\big\|_2^2 + \gamma_{\mathrm{g}}\big\|\mathbf{P}_n\mathbf{L}\hat{\mathbf{x}}_0\big\|_2^2,$$

for DPS (acting on the current prediction $\hat{\mathbf{x}}_0$), and the analogous expression with $\mathbf{x}$ in the DiffPIR proximal subproblem.

Algorithm 4 GSNR–DiffPIR Sampling null-space and graph regularization

Require: K, $\mathbf{H}$, $\mathbf{y}$, σ_n, $\{\tilde{\sigma}_i\}_{i=1}^K$, ζ, η, γ, γ_{g}, $\mathbf{S}$, G^*, $\mathbf{P}_n$, $\mathbf{L}$

Precompute $\rho_i \leftarrow \eta\,\sigma_n^2/\tilde{\sigma}_i^2$ for $i = 1, \ldots, K$
$\mathbf{x}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = K, \ldots, 1$ **do**
  $\hat{\mathbf{s}} \leftarrow \mathbf{s}_\theta(\mathbf{x}_i, i)$
  $\tilde{\mathbf{x}}_0^{(i)} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\mathbf{x}_i + (1-\bar{\alpha}_i)\hat{\mathbf{s}})$
  $\hat{\mathbf{x}}_0^{(i)} \leftarrow \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \rho_i\|\mathbf{x} - \tilde{\mathbf{x}}_0^{(i)}\|_2^2 + \gamma\|\mathrm{G}^*(\mathbf{y}) - \mathbf{S}\mathbf{x}\|_2^2 + \gamma_{\mathrm{g}}\|\mathbf{P}_n\mathbf{L}\mathbf{x}\|_2^2$
  $\hat{\boldsymbol{\epsilon}} \leftarrow \frac{1}{\sqrt{1-\alpha_i}}\Big(\mathbf{x}_i - \sqrt{\bar{\alpha}_i}\,\hat{\mathbf{x}}_0^{(i)}\Big)$
  $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_{i-1} \leftarrow \sqrt{\bar{\alpha}_{i-1}}\,\hat{\mathbf{x}}_0^{(i)} + \sqrt{1-\bar{\alpha}_{i-1}}\big(\sqrt{1-\zeta}\,\hat{\boldsymbol{\epsilon}} + \sqrt{\zeta}\,\boldsymbol{\epsilon}_i\big)$
end for
return $\hat{\mathbf{x}}_0^{(1)}$

Algorithm 5 GSNR–MPGD Sampling null-space and graph regularization

Require: K, $\mathbf{H}$, $\mathbf{y}$, $\{\zeta_i\}_{i=1}^K$, $\{\tilde{\sigma}_i\}_{i=1}^K$, γ, γ_{g}, $\mathbf{S}$, G^*, $\mathbf{P}_n$, $\mathbf{L}$

$\mathbf{z}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = K-1, \ldots, 0$ **do**
  $\epsilon_i \sim \mathcal{N}(0, \mathbf{I})$
  $\mathbf{z}_{0|i} = \frac{1}{\sqrt{\bar{\alpha}_i}}(\mathbf{z}_i - \sqrt{1-\bar{\alpha}_i}\epsilon_\theta(\mathbf{z}_i, i)$
  $\mathbf{z}_{0|i} = \mathbf{z}_{0|i} - \zeta_i\Big(\nabla_{\mathbf{z}_{0|i}}\|\mathbf{H}\mathrm{D}(\mathbf{z}_{0|i}) - \mathbf{y}\| + \gamma\big\|\mathrm{G}^*(\mathbf{y}) - \mathbf{S}\mathrm{D}(\mathbf{z}_{0|i})\big\|_2^2 + \gamma_{\mathrm{g}}\big\|\mathbf{P}_n\mathbf{L}\mathrm{D}(\mathbf{z}_{0|i})\big\|_2^2\Big)$
  $\mathbf{z}_{i-1} = \sqrt{\bar{\alpha}_{i-1}}\mathbf{z}_{0|i} + \sqrt{1-\bar{\alpha}_{i-1}-\sigma_i^2}\epsilon_\theta(\mathbf{z}_i, i) + \sigma_i\epsilon_i$
end for
return $\hat{\mathbf{x}} = \mathrm{D}(\mathbf{z}_0)$

A8.3. MPGD [15].

We evaluate graph-null-space-regularized manifold projected gradient descent (MPGD) [15] for super-resolution. This DM is performed on the latent space [36]. We used the pre-trained latent diffusion model from [2] with the CelebA-HQ model. In Algorithm 5, we show the GSNR modification of MPGD. For the experiments, we used the $\mathbf{L} = \mathbf{L}_{8nn}$ variant with $p = 0.1n$. In Fig. 17, we present the visual outcomes of incorporating GSNR into MPGD. This integration yields up to 0.78 dB SR improvement, indicating that GSNR enhances even competitive end-to-end diffusion-based solvers.

A9. CS results

Fig. 18 shows an ablation of (8) for the convergence of the reconstruction in PnP-FISTA for CS. PnP Baseline indicates that $\gamma = \gamma_g = 0$. For the GSNR, with only the graph-regularizer (green), $\gamma = 0$ is used. The state-of-the-art, NPN baseline (red) uses the matrix $\mathbf{S}$ from [17] and $\gamma_g = 0$, obtains greater acceleration. GSNR without graph-regularization (blue) uses the proposed operator

[2] github.com/CompVis/latent-diffusion

Figure 17. Results using latent-space diffusion models for SR.

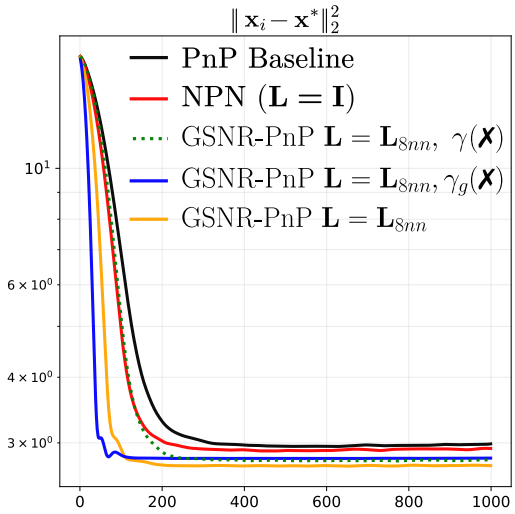

Figure 18. Convergence comparison with PnP, NPN, and GSNR for CS.

$\mathbf{S}$ from (6) and the learned network G with $\gamma_g = 0$. Finally, GSNR with both regularizers, in yellow, utilizes the entire equation (8). From this figure, we can conclude the contribution of each term: if $\frac{\gamma_g}{2}(\mathbf{P}_n\mathbf{x})^\top \mathbf{L}(\mathbf{P}_n\mathbf{x})$ is used, better convergence and reconstruction will be achieved. If $\gamma \|\mathrm{G}(\mathbf{y}) - \mathbf{S}\mathbf{x}\|_2^2$ is used, convergence is further accelerated and reconstruction is improved. Using both terms guarantees good convergence and the best possible reconstruction quality (yellow).

A10. Demosaicking results

In Table 3, the parameter settings are shown for demosaicing experiments. As shown in Table 4, GSNR consistently improves over the baselines. With Lip-DnCNN, GSNR with either $\mathbf{L}_{4\mathrm{nn}}$ or $\mathbf{L}_{8\mathrm{nn}}$ yields a clear PSNR gain over both PGD-PnP and NPN, showing that graph-limited null-space information brings a systematic boost. With DRUNet, all methods achieve similar absolute PSNR, but GSNR still matches or slightly surpasses NPN across graph topologies.

Table 3. Experimental settings for GSNR PGD-PnP in demosaicing.

Parameter	Value
γ_{g}	0.01
γ	0.1
η	0.01
α	$\frac{1.0}{\|\mathbf{H}\|}$
K (Max iterations)	3000

Table 4. Final PSNR (dB) comparison across graph variants and topologies for demosaicing with $n = 64^2$ and $p = 0.5 \cdot 3 \cdot n$.

Method	γ_{g}	Lip-DnCNN [40]	DRUNet [52]
Baseline	–	39.35	27.91
NPN [17]	✘	39.77	30.12
GSNR w. $\mathbf{L}_{4nn}$	✘	39.79	30.14
GSNR w. $\mathbf{L}_{4nn}$	✓	39.89	30.13
GSNR w. $\mathbf{L}_{8nn}$	✘	39.77	30.27
GSNR w. $\mathbf{L}_{8nn}$	✓	39.88	30.27

Overall, the improvements are modest yet consistent, underscoring that the main advantage comes from the GSNR null-space representation rather than the specific choice of backbone denoiser or additional graph penalty.

A11. SR results

SAR details for SR: from which we extract 3289 patches of size 128×128 from six satellite scenes for training and 1162 patches from two additional satellite scenes for evaluation.

Figure 19 demonstrates that the proposed graph-smooth null-space representation extends beyond natural images to other imaging domains. The top-row reconstructions show that GSNR-PnP with a grid Laplacian $\mathbf{L}_{4\mathrm{nn}}$ produces sharper, less noisy structures than both the baseline PnP solver and the geometry-free $\mathbf{L} = \mathbf{I}$ variant, even though the scene exhibits a speckled texture rather than a smooth photographic content. The corresponding PSNR curves in the bottom panel confirm this behavior quantitatively: GSNR-PnP with $\mathbf{L}_{4\mathrm{nn}}$ converges faster and stabilizes at the highest PSNR, while GSNR-PnP with $\mathbf{L} = \mathbf{I}$ still improves over the baseline but remains clearly below the graph-based design. These results indicate that encoding graph-smooth structure in the null-space is beneficial not only for face or natural-image SR, but also for more challenging sensing models such as SAR-like imaging, supporting the broader applicability of GSNR across imaging modalities.

Figure 20 illustrates the effect of the graph-smooth null-space design on super-resolution performance and convergence. The PSNR–iteration plots (left) show that, for all three denoisers, GSNR-PnP with either $\mathbf{L}_{4\mathrm{nn}}$ or $\mathbf{L}_{8\mathrm{nn}}$ converges to a higher PSNR than both the baseline PnP solver

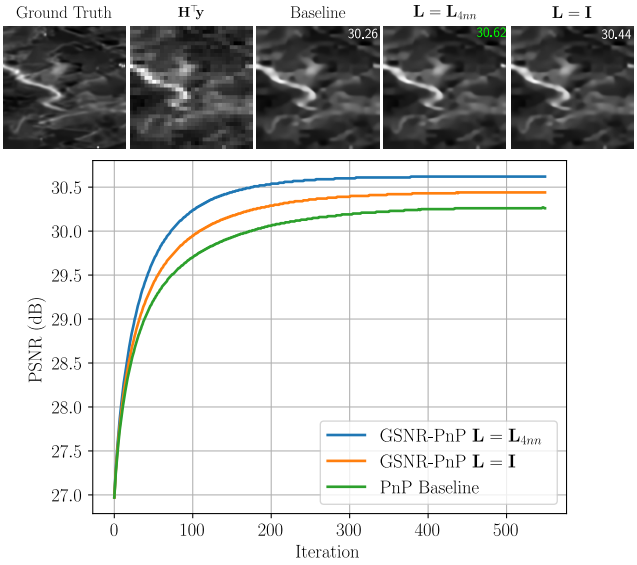

Figure 19. Evaluation of GSNR-PnP on a non-optical imaging example (SAR-like data). Top row: ground-truth image $\mathbf{x}^*$, back-projection $\mathbf{H}^\top\mathbf{y}$, baseline PnP reconstruction, and GSNR-PnP with $\mathbf{L}_{4\text{nn}}$ and $\mathbf{L} = \mathbf{I}$, with PSNR values overlaid. Bottom: PSNR as a function of iteration for GSNR-PnP with $\mathbf{L}_{4\text{nn}}$, GSNR-PnP with $\mathbf{L} = \mathbf{I}$, and the PnP baseline.

and the geometry-free $\mathbf{L} = \mathbf{I}$ variant. With DnCNN and DRUNET, the graph-based curves not only reach a higher plateau but also exhibit a steeper initial rise, indicating a faster approach to a good solution. Even with the simpler wavelet denoiser, the graph-limited versions match or slightly improve upon the baseline while maintaining stable dynamics. The image grid on the right confirms these trends visually: reconstructions obtained with $\mathbf{L}_{4\text{nn}}$ and $\mathbf{L}_{8\text{nn}}$ display sharper facial contours and more coherent high-frequency details than both the baseline and $\mathbf{L} = \mathbf{I}$, highlighting the benefit of injecting graph-smooth structure specifically into the null space.

Figure 21 shows the evolution of PSNR for super-resolution on 20 images from the CelebA dataset when DIP is run with and without the proposed graph-smooth null-space representation. The DIP baseline (red) improves rapidly at first but then saturates at a lower PSNR and exhibits noticeable instabilities, including pronounced dips during the late iterations. In contrast, all GSNR-DIP variants converge to a higher PSNR plateau and have much smoother trajectories. Among them, the graph-based choices $\mathbf{L}_{4\text{nn}}$ and $\mathbf{L}_{8\text{nn}}$ (blue/green) provides the most stable and accurate reconstructions, consistently outperforming both the baseline and the geometry-free $\mathbf{L} = \mathbf{I}$ (orange). This indicates that enforcing graph-smooth structure specifically in the null space not only improves the final reconstruction quality but also regularizes the DIP optimization itself, mitigating the overfitting and oscillations typically observed in vanilla DIP.

Table 5. Experimental settings for GSNR PnP-PGD in deblurring.

Parameter	Lip-DnCNN	Wavelet
γ_g	0.1	0.1
γ	0.1	0.1
η	0.0001	0.001
α	$\frac{1.5}{\|\|\mathbf{H}\|\|}$	$\frac{1.5}{\|\|\mathbf{H}\|\|}$
K (Max iterations)	800	800

Table 6. GSNR's computational cost for SR: offline and online (PnP, one image, 1000 iters).

Metric / Resolution	128^2	256^2	512^2
Offline EVD computation (s)	102	1315	22310
Offline RAM (GiB)	0.68	10.88	174.09
PnP runtime (Wavelet) (s)	1.09	1.42	1.96

A12. Deblurring results

In the deblurring setting, the sensing matrix $\mathbf{H}$ does not reduce dimensionality, meaning that $n - m = 0$. Consequently, the full set of n eigenvectors of $\mathbf{T}$ was considered when selecting the p smoothest directions.

Experimental settings. Table 5 reports the GSNR PnP-PGD parameters defined to perform image reconstruction.

Visual results. Fig. 22 presents deblurring examples on the CelebA and Places365 datasets for baseline PGD-PnP, GSNR with $\mathbf{L} = \mathbf{I}$ (NPN), and GSNR with $\mathbf{L} = \mathbf{L}_{8nn}$, using either Lip-DnCNN or Wavelet denoisers. The qualitative behavior is consistent across both denoisers. The baseline PGD-PnP produces reconstructions with noticeable residual blur and poor recovery of high-frequency structures, such as eyes and eyebrows in faces or the sharp contours of tree tops in natural scenes.

Using GSNR with $\mathbf{L} = \mathbf{I}$ improves the reconstruction of high-frequency components, producing visually sharper results. However, because the identity Laplacian imposes no geometric constraints, the recovered details are not necessarily aligned with the true image structures. This often leads to overly sharp but inaccurate features that do not reflect the desired image geometry. In contrast, GSNR with graph-based Laplacians, such as $\mathbf{L}_{8nn}$, enforces geometric consistency through graph-smoothness. As a result, the restored high-frequency structures are both sharp and coherent with the true image content, yielding the most faithful reconstructions among the tested methods. This advantage is reflected not only visually but also quantitatively, with GSNR achieving the highest PSNR values. A noteworthy observation is that GSNR enables reconstructions using a simple Wavelet denoiser to surpass the performance of reconstructions obtained with a more advanced neural-network denoiser like DnCNN. This highlights GSNR's capacity to adapt to and significantly strengthen any recon-

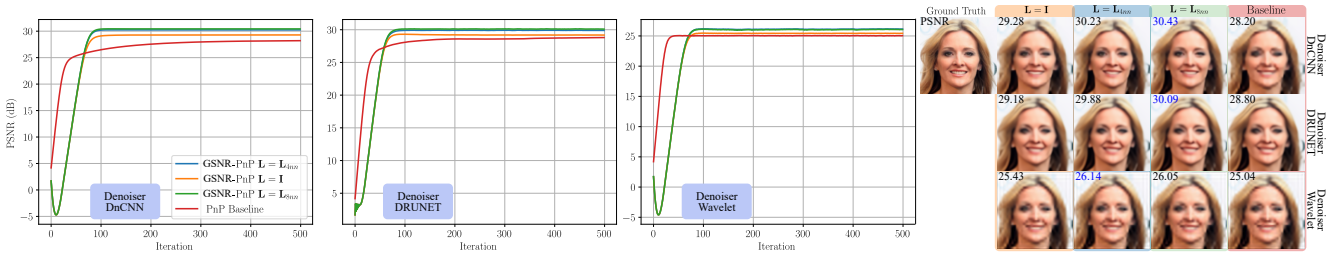

Figure 20. Super-resolution results with GSNR-PnP for different graph Laplacians and denoisers. Left: PSNR versus iteration for GSNR-PnP with $\mathbf{L} = \mathbf{I}$, $\mathbf{L}_{4\text{nn}}$, $\mathbf{L}_{8\text{nn}}$, and the PnP, using (from left to right) DnCNN, DRUNET, and a wavelet denoiser. Right: corresponding reconstructions for a representative face image, with the ground truth at the top-left and PSNR values overlaid on each result.

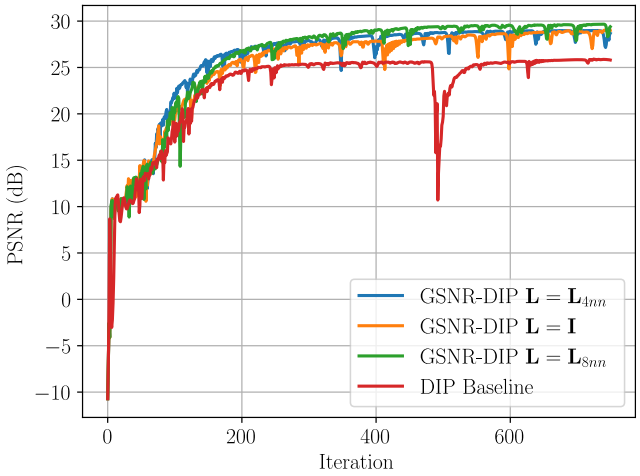

Figure 21. Super-resolution on CelebA (20 images) with Deep Image Prior (DIP). We plot PSNR versus iteration for the standard DIP baseline (red) and the proposed GSNR-DIP variants using different Laplacians: $\mathbf{L} = \mathbf{I}$, $\mathbf{L}_{4\text{nn}}$, and $\mathbf{L}_{8\text{nn}}$.

Table 7. Final PSNR on DIV2K for deblurring (G trained on Places365).

	Lip-DnCNN	Wavelet
Baseline (PnP-PGD)	31.23	30.39
NPN [17]	33.05	32.90
GSNR ($\mathbf{L}_{8\text{nn}}$)	**33.69**	**33.65**

struction method, offering robust improvements across a broad range of inverse problem solvers.

A13. Scalability and computational cost

We use 128^2 images to enable controlled comparisons. GSNR introduces only a low *online* overhead at inference time, since the subspace computation is performed *offline* once per $(\mathbf{H}, \mathbf{L}, n, p)$ and reused. We only compute the p smoothest eigenvectors (selected by the coverage plateau in Algorithm 1), so memory scales as $\mathcal{O}(pn)$. We never materialize $\mathbf{P}_n$ or $\mathbf{T}$: we wrap $\mathbf{Tx} = \mathbf{P}_n\mathbf{L}\mathbf{P}_n\mathbf{x}$ as an implicit LinearOperator and compute the first p eigenpairs via ARPACK `eigsh` (A5), exploiting sparse $\mathbf{L}$ for fast matrix operations. Table 6 reports scaling to 512^2; the online cost remains low, whereas the offline EVD increases with resolution. For $n^2 \geq 1024^2$, EVD computation becomes costly with ARPACK. Future work will focus on efficient approximate EVD computation.

A14. Dataset generalization and neural network ablation

To assess whether GSNR regularization transfers across image distributions, we performed a cross-dataset experiment for deblurring setting. We trained the neural network G on Places365 and then evaluated GSNR on DIV2K [1]. The results in Table 7 show that the gains provided by GSNR remain consistent under this distribution shift, indicating that the learned null-space predictor generalizes to diverse natural images.

We further examine the robustness of GSNR with respect to the choice of neural network architecture for G. In Figure 23, we replace the default U-Net with three recent architectures: DFPIR [47], EVSSM [22], and AdaIR [10]. As expected, stronger predictors yield more accurate null-space estimates, as reflected by the higher PSNR values for the null-space mapping (see Fig. 23 legend). Importantly, GSNR improves PnP reconstruction quality and accelerates convergence for all tested backbones, suggesting its benefits arise from the proposed GSNR regularization rather than a specific architecture.

A15. Inexact forward operator

In many inverse problems, the forward operator available to the reconstruction algorithm is only an approximation of the true sensing physics. In practice, the nominal sensing matrix $\mathbf{H}$ can deviate from the actual measurement operator due to calibration errors, hardware tolerances, or other unmodeled effects. Such mismatches are particularly detrimental because they propagate into the algorithmic components derived from $\mathbf{H}$, including null-space projectors and any structure imposed through them, and can therefore undermine both accuracy and convergence. To quantify this effect in a concrete setting, Fig. 24 evaluates deblurring under operator mismatch. Measurements are generated as $\mathbf{y} = (\mathbf{H} + \mathbf{H}_\xi)\mathbf{x}^* + \boldsymbol{\omega}$, with $\mathbf{H}_\xi \sim \mathcal{N}(0, 0.005^2)$ while re-

Figure 22. Deblurring visual results comparing PGD-PnP, GSNR with $\mathbf{L} = \mathbf{I}$ (NPN), and GSNR with $\mathbf{L} = \mathbf{L}_{8nn}$ across two datasets and two denoisers. Best results for each denoiser are in **bold**. Here, $p = 0.8$.

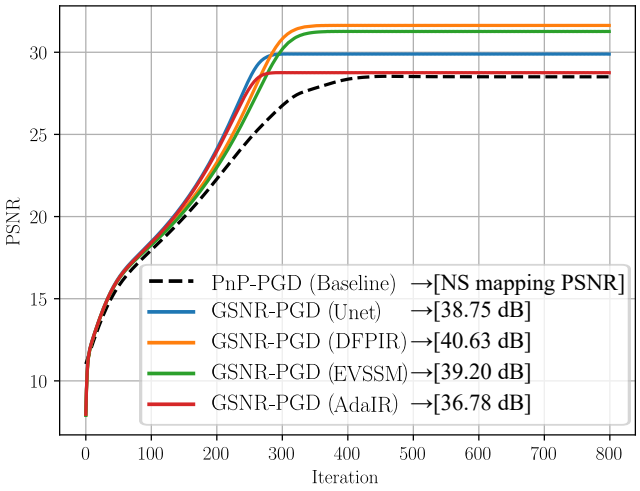

Figure 23. PSNR with G ablation in Places for the deblurring task.

covery still assumes the nominal $\mathbf{H}$. This mismatch yields imperfect estimates of $\mathbf{P}_n$ and $\mathbf{S}$, which in turn degrades the null-space representation used by GSNR. Despite these compounded imperfections, GSNR remains effective, improving performance by approximately 1 dB and converging in fewer iterations.

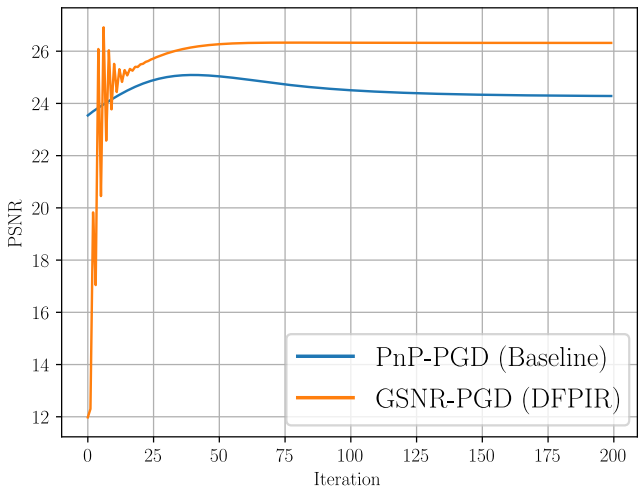

Figure 24. PSNR in deblurring with inexact forward operator.