DIVERGENCE-FREE NEURAL NETWORKS WITH APPLICATION TO IMAGE DENOISING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a resource-efficient neural network architecture with zero divergence by design, adapted for high-dimensional problems. Our method is directly applicable to image denoising, for which divergence-free estimators are particularly well-suited for self-supervised learning, in accordance with Stein's unbiased risk estimation theory. Comparisons of our parameterization on popular denoising datasets demonstrate that it retains sufficient expressivity to remain competitive with other divergence-based approaches, while outperforming its counterparts when the noise level is known.

1 INTRODUCTION

The divergence is a scalar quantity that measures the rate at which a vector field "flows out" of an infinitesimal region of space. Formally, for a weakly differentiable function $f : \mathbb{R}^n \to \mathbb{R}^n$, the divergence at point $\boldsymbol{y} \in \mathbb{R}^n$ is defined as the trace of the Jacobian matrix $\boldsymbol{J}_f(\boldsymbol{y})$:

$$\operatorname{div} f(\boldsymbol{y}) \triangleq \operatorname{tr}(\boldsymbol{J}_f(\boldsymbol{y})) = \sum_{i=1}^{n} \frac{\partial f_i}{\partial y_i}(\boldsymbol{y})\,. \tag{1}$$

In the special case where the divergence is zero everywhere, the vector field is said to be divergence-free or solenoidal, indicating an incompressible flow. One of the most famous example is without doubt the magnetic field, which, according to Maxwell's equations, has zero divergence (Maxwell, 1873). Learning divergence-free vector fields is of particular interest at the interface of physics and machine learning (Richter-Powell et al., 2022; Raissi et al., 2017a), as such fields naturally emerge in systems governed by fundamental conservation laws. Parameterizations for learning often exploit the fact that, in $\mathbb{R}^3$, the curl of any vector field is divergence-free (Morita, 2001), or, more generally, draw on its multidimensional extension via differential forms (Cartan, 1899; Richter-Powell et al., 2022). As long as the target functions remain low-dimensional, training can be performed efficiently with the help of an automatic differentiation engine (Paszke et al., 2019) that powers the computation of partial derivatives. However, scaling challenges arise quickly as the dimensionality increases (Richter-Powell et al., 2022).

In this paper, we establish a representer theorem for divergence-free vector fields, based on structured combinations of conservative fields. Building on this result, and incorporating sparsity constraints, we show how this representation can inform neural network parameterizations that remain resource-efficient, thereby ensuring computational tractability in high dimension. With application to image denoising, for which divergence-free estimators are particularly well-suited for self-supervised learning, in accordance with Stein's unbiased risk estimation theory Stein (1981), we propose a methodology to construct low-overhead network architectures that have zero divergence by design and which are adapted to image processing tasks. We demonstrate their competitiveness in comparison to other divergence-based approaches (Batson & Royer, 2019; Tachella et al., 2025a; Soltanayev & Chun, 2018) for the removal of Gaussian noise without clean data.

In summary, the contributions of our work are as follows:

1. The establishment of a representer theorem for divergence-free fields, on which we build to construct neural network architectures that have zero divergence by design.

2. A theoretical framework for analyzing self-supervised image denoising methods grounded in the principle of constant divergence.
3. The demonstration of the competitiveness of our approach in comparison with other divergence-based approaches, particularly when the noise level is known.

2 Related Work

Divergence-free networks are particularly studied within physics-informed machine learning and related scientific modeling tasks, which integrate physical laws into the training of neural networks to solve partial differential equations. Notably, enforcing incompressibility constraints is often important—especially in fluid dynamics, where velocity fields are required to be divergence-free.

A common approach employs soft constraints by adding penalty terms to the loss function that encourage the predicted fields to be divergence-free (Raissi et al., 2017b; Mao et al., 2020; Jin et al., 2021). Although such penalty-based methods are straightforward to implement, they do not guarantee strict satisfaction of the incompressibility condition, and residual divergence can remain in some cases, particularly for complex or high-dimensional problems.

To overcome these limitations, recent works have explored hard constraints that enforce divergence-free properties by construction through network architecture or parameterization. For example, in Raissi et al. (2017a), a stream function formulation is used in 2D to represent the velocity field as derivatives of a scalar network output, which is analytically divergence-free. Extending this idea to the multidimensional case, Richter-Powell et al. (2022) designed networks that directly encode conservation laws—including divergence-free constraints—thereby allowing modeling of flow fields and advected quantities without explicit divergence penalties. Nonetheless, scaling these models proves challenging due to their heavy reliance on automatic differentiation. For example, the vector-field parameterization proposed by Richter-Powell et al. (2022) requires computing a Jacobian matrix, which becomes intractable as the dimension grows.

3 Divergence-Based Approaches for Self-Supervised Denoising

We focus on denoising problems under the assumption of additive white Gaussian noise (AWGN):

$$\boldsymbol{y} = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}\,, \tag{2}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the noisy observation, $\boldsymbol{x} \in \mathbb{R}^n$ is the underlying noise-free signal, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$ models the random nature of noise and $\sigma > 0$ is the noise level. Provided that a sufficiently large dataset composed of pairs consisting of a clean signal and its noisy counterpart $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$ is available, problem (2) is traditionally tackled in a supervised manner by solving:

$$\arg\min_{f}\ \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2\,, \tag{3}$$

that is, by finding the minimum mean square error (MMSE) estimator, which we denote f^{MMSE}. Interestingly, f^{MMSE} has a closed-form expression which is given by Tweedie's formula (Efron, 2011) which reads $f^{\mathrm{MMSE}}(\boldsymbol{y}) = \boldsymbol{y} + \sigma^2 \nabla \log p_{\boldsymbol{y}}(\boldsymbol{y})$. In this latter expression, the optimal estimator f_{MMSE} depends solely on the score of the distribution of the noisy data $\nabla \log p_{\boldsymbol{y}}(\boldsymbol{y})$. Accordingly, this formulation indicates that Gaussian denoising may be effectively addressed even in the absence of ground-truth data $\boldsymbol{x}$ for training.

Among all the methods proposed in the literature for tackling self-supervised denoising, divergence-based approaches hold a prominent place. They are all grounded in Stein's Unbiased Risk Estimator (SURE) theory (Stein, 1981) which establishes a remarkable identity involving the divergence operator:

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 = \mathbb{E}_{\boldsymbol{y}} \left[-n\sigma^2 + \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 + 2\sigma^2 \operatorname{div} f(\boldsymbol{y}) \right]\,, \tag{4}$$

provided that f belongs to $\mathcal{L}^1$, the space of weakly differentiable functions, and under the assumption that $\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x}}|f_i(\boldsymbol{y})|$ is bounded. This result is particularly powerful, as it reveals that the mean square error can be reformulated to depend solely on noisy observations, as long as the divergence can be computed. Consequently, equation (4) can, in effect, be interpreted as a self-supervised loss for Gaussian denoising. Many traditional image denoisers—whose divergence admits a closed-form

expression—are in fact rooted in this identity (Blu & Luisier, 2007; Van De Ville & Kocher, 2009; Wang & Morel, 2013), even if this connection is not made explicit in some cases (Dabov et al., 2007; Lebrun et al., 2013), as shown by Herbreteau & Kervrann (2025). However, when the estimator f is considerably more complex, such as a deep neural network, its divergence is generally intractable to compute. In what follows, we describe two distinct approaches proposed in the literature to use (4) anyway as a self-supervised loss for training networks, propose a third way and then study their shared properties.

Remark In addition to the divergence-based approaches studied in this paper, we also mention, for completeness, the approaches that directly utilize the score function $\nabla \log p_{\boldsymbol{y}}(\boldsymbol{y})$ in Tweedie's formula, as proposed in Kim & Ye (2021); Xie et al. (2023), all of which depend on the estimation technique introduced by Lim et al. (2020). Moreover, Noise2Noise-like (Lehtinen et al., 2018) data augmentation techniques were also proposed (Pang et al., 2021; Huang et al., 2021; Wang et al., 2022; Mansour & Heckel, 2023) as an alternative to SURE.

3.1 Blind-spot estimators

A radical way to bypass the computation of the divergence is to impose that each component function f_i does not depend on y_i. Under this constraint, f becomes trivially divergence-free by construction since $\forall \boldsymbol{y} \in \mathbb{R}^n, \frac{\partial f_i}{\partial y_i}(\boldsymbol{y}) = 0$. This idea dates back to Efron (2004) and lies at the core of the **Noise2Self** approach (Batson & Royer, 2019) and its variants (Krull et al., 2019; Laine et al., 2019), in which a so-called "blind-spot" network architecture is employed. From a broader perspective, this constraint can be generalized by restricting f to the space

$$\mathcal{S}^c_{\mathrm{BS}} = \{f \in \mathcal{L}^1(\mathbb{R}^n, \mathbb{R}^n) \,:\, \forall \boldsymbol{y} \in \mathbb{R}^n, \frac{\partial f_i}{\partial y_i}(\boldsymbol{y}) = c\}\,, \tag{5}$$

where $c \in \mathbb{R}$ is an arbitrary constant, fixed in advance.

An important byproduct of this approach is that the divergence term in (4) becomes constant and thus irrelevant to the optimization objective, regardless of the noise level σ. Consequently, in addition to not requiring clean targets $\boldsymbol{x}$, blind-spot estimators also dispense with prior knowledge of the noise level σ, thereby avoiding the need for its ad hoc estimation (Chen et al., 2015; Pyatykh et al., 2013; Foi et al., 2008). Finally, the blind-spot approach simply amounts to minimizing the data consistency term:

$$\arg \min_{f \in \mathcal{S}^c_{\mathrm{BS}}} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2\,. \tag{6}$$

In the case $c = 0$, the solution of (6) is given by $f_i^{\mathrm{BS}}(\boldsymbol{y}) = \mathbb{E}\{y_i|\boldsymbol{y}_{-i}\} = \mathbb{E}\{x_i|\boldsymbol{y}_{-i}\}$, where $\boldsymbol{y}_{-i}$ refers to the vector obtained by excluding the ith entry.

The strength of blind-spot approaches lies actually in their versatility: they can handle a wide range of noise types, specifically those that are zero-mean and spatially independent, of which (2) is a prime example, without precisely knowing the noise distribution. However, this flexibility comes at a significant performance cost. A blind-spot architecture is indeed inherently less expressive than a classic one, especially since y_i is usually highly informative about x_i, and tends to introduce checkerboard artifacts (Höck et al., 2022).

3.2 Monte Carlo approximation methods

Alongside the blind-spot approach, an alternative method involves employing a Monte Carlo approximation of the divergence, grounded in the following result (Ramani et al., 2008):

$$\operatorname{div} f(\boldsymbol{y}) = \lim_{\tau \to 0} \mathbb{E}_{\boldsymbol{h} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})} \left[\boldsymbol{h}^\top \frac{f(\boldsymbol{y} + \tau \boldsymbol{h}) - f(\boldsymbol{y})}{\tau} \right], \tag{7}$$

provided that f admits a well-defined second-order Taylor expansion (if not, this is still valid in the weak sense provided that f is tempered, which is the case for networks with piecewise differentiable activation functions as shown by Soltanayev & Chun (2018)). In practice, a single realization $\boldsymbol{h}$ from the standard normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is used for approximating the divergence and τ is chosen

as a small constant. In total, only two evaluations of the function f are necessary to estimate its divergence with this method.

In a deep learning setting, Soltanayev & Chun (2018); Chen et al. (2022) leveraged this Monte Carlo approximation in combination to the SURE loss (4) to train neural networks on datasets composed only of noisy observations $\boldsymbol{y}$, leading to the **MC-SURE** approach. While they achieved performance close to that of the MMSE estimator, a slight gap remains, partly due to approximation errors in the divergence term.

UNSURE To overcome the limitation of requiring knowledge of the noise level σ, Tachella et al. (2025a) proposed a softened version of the constraint (5), imposing only that the estimator has *zero expected divergence*, that is, $\mathbb{E}_{\boldsymbol{y}} \operatorname{div} f(\boldsymbol{y}) = 0$. This relaxation has the advantage to produce the same simplification effect on the optimization objective (4) as with blind-spot estimators, while allowing more expressivity. Extending it to the constant case, this alternative constraint forces f to belong to the space

$$\mathcal{S}^c_{\mathrm{CED}} = \{f \in \mathcal{L}^1(\mathbb{R}^n, \mathbb{R}^n) \,:\, \mathbb{E}_{\boldsymbol{y}} \operatorname{div} f(\boldsymbol{y}) = nc\}\,. \tag{8}$$

Note that we do have $\mathcal{S}^c_{\mathrm{BS}} \subset \mathcal{S}^c_{\mathrm{CED}}$. Similarly to blind-spot approaches, training consists in solving:

$$\arg\min_{f \in \mathcal{S}^c_{\mathrm{CED}}} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2\,, \tag{9}$$

for which a closed-form solution was established in the case $c = 0$, namely $f^{\mathrm{CED}}(\boldsymbol{y}) = \boldsymbol{y} + \hat{\eta} \nabla \log p_{\boldsymbol{y}}(\boldsymbol{y})$, with $\hat{\eta} = (\mathbb{E}_{\boldsymbol{y}} \frac{1}{n} \|\nabla \log p_{\boldsymbol{y}}(\boldsymbol{y})\|_2^2)^{-1}$. An important difference with the blind-spot approach lies in how the optimization is carried out in practice. Indeed, unlike blind-spot approaches—where the constraint is enforced directly through the design of f—the UNSURE approach seeks a saddle point of the Lagrangian by formulating the problem as a min–max optimization which is solved by alternating gradient-descent-ascent (Arrow et al., 1958; Platt & Barr, 1987). However, such an optimization method comes with several caveats. First, constraint satisfaction is not guaranteed in practice; only the penalty term associated with violations is minimized. Second, the outcome is highly sensitive to the choice of learning-rate pair for gradient-descent-ascent, which controls the trade-off between the objective and the constraint, and an inappropriate choice can lead to instabilities or oscillatory dynamics during training (Platt & Barr, 1987; Gallego-Posada et al., 2022). Finally, in this setting, the divergence term is estimated via a Monte Carlo approximation based on (7) using a limited number of samples, which can further degrade the accuracy of the optimization.

3.3 Proposed alternative: divergence-constant estimators

In this work, we propose to study the set of weakly differentiable vector fields on $\mathbb{R}^n$ with constant (normalized) divergence $c \in \mathbb{R}$, denoted by

$$\mathcal{S}^c_{\mathrm{DC}} = \{f \in \mathcal{L}^1(\mathbb{R}^n, \mathbb{R}^n) \,:\, \forall \boldsymbol{y} \in \mathbb{R}^n, \operatorname{div} f(\boldsymbol{y}) = nc\}\,, \tag{10}$$

de facto introducing an intermediate constraint set lying between the strict blind-spot constraint set and the much looser expected divergence constraint one: $\mathcal{S}^c_{\mathrm{BS}} \subset \mathcal{S}^c_{\mathrm{DC}} \subset \mathcal{S}^c_{\mathrm{CED}}$. We emphasize that all inclusions are strict, with in particular the possibility for $f \in \mathcal{S}^0_{\mathrm{DC}}$ to have each of its component function f_i to depend on y_i, which is excluded for a function in $\mathcal{S}^0_{\mathrm{BS}}$ (see Appendix C). We postpone the description of the way we construct in practice such divergence-constant estimators to the next section. Let us simply note that, similar to existing alternatives, divergence-constant mappings are of particular interest in self-supervised denoising in view of (4) since training consists in solving:

$$\arg\min_{f \in \mathcal{S}^c_{\mathrm{DC}}} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2\,. \tag{11}$$

3.4 Properties shared by all constraint sets

For conciseness, $\mathcal{S}^c$ denotes one of the following sets: $\mathcal{S}^c_{\mathrm{BS}}, \mathcal{S}^c_{\mathrm{CED}}$ or $\mathcal{S}^c_{\mathrm{DC}}$. This paragraph should be read by selecting one of these three sets consistently, without mixing them. As a preliminarily observation, we notice that the constraint set $\mathcal{S}^c$ admits an affine space structure, based at $c\,\mathrm{id}$, where id refers to the identity map on $\mathbb{R}^n$. This statement is formalized in the following lemma (all the proofs of this paper are given in Appendix B).

Lemma 1. *$\mathcal{S}^0$ is a linear space and $\mathcal{S}^c$ is an affine space with $\mathcal{S}^c = c\,\mathrm{id} + \mathcal{S}^0$.*

A direct consequence (see Proposition 1) is that the optimal denoiser within each class can be written as an affine combination of the identity function and the minimizer in $\mathcal{S}^0$ of the data consistency term. Thus, it is sufficient to restrict the search to estimators in $\mathcal{S}^0$, since any optimal denoiser in $\mathcal{S}^c$ can be recovered straightforwardly from an optimal denoiser in $\mathcal{S}^0$.

Proposition 1. *In the AWGN setting (see (2)),*

$$\arg\min_{f \in \mathcal{S}^c} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 = c\,\mathrm{id} + (1-c) \arg\min_{f \in \mathcal{S}^0} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 \,.$$

An immediate question that arises at this point is: *How to choose the constant c to achieve the best denoising?* Proposition 2 provides a theoretical characterization of the optimal constant c^*, provided that the noise level σ is known.

Proposition 2 (Optimal constant)**.** *In the AWGN setting (see (2)),*

$$c^* = \arg\min_{c \in \mathbb{R}} \min_{f \in \mathcal{S}^c} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 = 1 - \frac{n\sigma^2}{\min_{f \in \mathcal{S}^0} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2} \in [0,1] \,.$$

Interestingly, the optimal constant c^* lies in $[0,1]$. As a consequence, the affine combination in Prop. 1 is in fact a convex combination in the optimal case. Naturally, c^* depends on the knowledge of the noise level σ, which may be unknown in some settings. This explains why the arbitrary choice $c = 0$ is made in practice (Batson & Royer, 2019; Tachella et al., 2025a).

4 Design of Divergence-Free Neural Networks

We now present our proposed methodology for constructing divergence-free network architectures.

4.1 Representing divergence-free vector fields

Lemma 2 offers a straightforward method for generating divergence-free vector fields and highlights the key role played by skew-symmetric matrices in ensuring zero divergence.

Lemma 2 (A simple divergence-free vector field)**.** *Let $\psi : \mathbb{R}^n \to \mathbb{R}$ be a smooth scalar field and let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ be a skew-symmetric matrix, i.e. $\boldsymbol{A}^\top = -\boldsymbol{A}$. The vector field $\boldsymbol{A}\nabla\psi$ is divergence-free.*

Nevertheless, this construction does not capture all divergence-free vector fields, expect for the case $n \leq 2$. In fact, fully representing such fields typically requires combining multiple expressions of this form, as formalized in the following representer theorem.

Theorem 1 (A universal approximation of divergence-free fields)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a smooth divergence-free vector field and let $\{\boldsymbol{A}_1, \dots, \boldsymbol{A}_K\} \in \mathbb{R}^{n \times n}$ be a basis of the space of real skew-symmetric $n \times n$ matrices. There exist smooth scalar fields $\psi_1, \dots, \psi_K : \mathbb{R}^n \to \mathbb{R}$ such that the vector field $\tilde{f} : \mathbb{R}^n \to \mathbb{R}^n$ defined as*

$$\tilde{f} = \sum_{k=1}^{K} \boldsymbol{A}_k \nabla \psi_k$$

is divergence-free and can approximate f "arbitrarily well".

The proof is an extension of the work of Richter-Powell et al. (2022) (for which the reader is referred to for more details on the precise meaning of "arbitrarily well") and builds on the classical Hodge decomposition theorem (Morita, 2001; Berger, 2003). Note that the space of real skew-symmetric $n \times n$ matrices is of dimension $K = \binom{n}{2}$, hence the number of scalar fields required to approximate a divergence-free field scales quadratically with the dimension n.

Application for $n = 3$ Consider the following basis of real skew-symmetric 3×3 matrices:

$$\boldsymbol{A}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}, \boldsymbol{A}_2 = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{ and } \boldsymbol{A}_3 = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

and let f be a smooth divergence-free vector field. According to Theorem 1, there exist $\psi_1, \psi_2, \psi_3 : \mathbb{R}^3 \to \mathbb{R}$, such that

$$\tilde{f} = \sum_{k=1}^{3} \boldsymbol{A}_k \nabla \psi_k = \begin{pmatrix} \frac{\partial \psi_3}{\partial y_2} - \frac{\partial \psi_2}{\partial y_3} \\ \frac{\partial \psi_1}{\partial y_3} - \frac{\partial \psi_3}{\partial y_1} \\ \frac{\partial \psi_2}{\partial y_1} - \frac{\partial \psi_1}{\partial y_2} \end{pmatrix}$$

is diververgence-free and can approximate f "arbitrarily well". In other words, there exists a vector field $\psi = (\psi_1, \psi_2, \psi_3)$ such that its curl approximates f "arbitrarily well". This is a well-known result in the literature (Morita, 2001).

4.2 Proposed architecture

Our goal is to construct a parameterized function f, under the form of a neural network, that is divergence-free by design and whose architecture is tailored for image processing tasks, in particular denoising. To this end, we build on the representer Theorem 1, which suggests defining f as a structured combination of conservative fields $\nabla \psi_k$. However, as previously noted, the number of terms in this representation, namely K in Theorem 1, is on the order of n^2, which becomes prohibitive as soon as we work with images. This scalability issue was previously highlighted by Richter-Powell et al. (2022). To keep computations tractable, we propose to constraint f to be represented using a sparse combination of conservative fields, which we assume retains sufficient representational fidelity. This deliberate simplification ultimately consists in substituting K with $K' \ll K$ (typically $K' = 8$). More precisely, we build f under the form

$$f = \sum_{k=1}^{K'} \boldsymbol{A}_k \nabla \psi_k \,, \tag{12}$$

where $\psi_1, \ldots, \psi_{K'} : \mathbb{R}^n \to \mathbb{R}$ are parameterized via a single shared neural network and $\{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_{K'}\} \in \mathbb{R}^{n \times n}$ are (sparse) skew-symmetric matrices, also parameterized. We now detail the construction of both types of parameterization.

Design of the skew-symmetric matrices For the sake of computational efficiency, the skew-symmetric matrices $\boldsymbol{A}_k$ in (12) are chosen to be sparse matrices with shared parameters as follows:

$$\boldsymbol{A}_k = \boldsymbol{P}_k^\top \frac{\boldsymbol{\Theta} - \boldsymbol{\Theta}^\top}{2} \boldsymbol{P}_k \,, \tag{13}$$

where $\boldsymbol{\Theta}$ is a shared parameterized repeated-block diagonal matrix and where each $\boldsymbol{P}_k \in \mathbb{R}^{n \times n}$ is a different and fixed permutation matrix (typically a rotation or shift matrix). Note that the matrices $\boldsymbol{A}_k$ are guaranteed to be skew-symmetric by design thanks to the following equality of sets, valid for any permutation matrix $\boldsymbol{P}_k$: $\{\boldsymbol{A} \in \mathbb{R}^{n \times n} \,:\, \boldsymbol{A}^\top = -\boldsymbol{A}\} = \{\boldsymbol{P}_k^\top \frac{\boldsymbol{A} - \boldsymbol{A}^\top}{2} \boldsymbol{P}_k \,:\, \boldsymbol{A} \in \mathbb{R}^{n \times n}\}$.

Design of the scalar fields The idea of designing neural networks to represent exact conservative fields, *i.e.* of the form $\nabla \psi$, has already been explored in works targeting energy based models or plug-and-play methods (Salimans & Ho, 2021; Hurault et al., 2022a). They all point out the critical choice of the architecture for the scalar potential function ψ in order to achieve good performance in practice. In particular, as experimentally observed, modeling ψ as a standard feedforward network, such as the ones used for classification, severely degrades performance. Instead, it is recommended to incorporate an architecture tailored to the target application directly into the design of ψ. This is why, we propose to consider parameterized scalar fields of the form

$$\psi_{\boldsymbol{\theta}, \boldsymbol{B}_k} : \boldsymbol{y} \in \mathbb{R}^n \mapsto \frac{1}{2} \left(\|\boldsymbol{B}_k \boldsymbol{y}\|_2^2 - \|\boldsymbol{B}_k \boldsymbol{y} - D_{\boldsymbol{\theta}}(\boldsymbol{y})\|_2^2 \right) \,, \tag{14}$$

where $\boldsymbol{B}_k \in \mathbb{R}^{n \times n}$ and $D_{\boldsymbol{\theta}} : \mathbb{R}^n \to \mathbb{R}^n$ is a neural network specific to image processing, typically a U-Net (Ronneberger et al., 2015). Please note that the neural network parameters $\boldsymbol{\theta}$ are shared for

all scalar fields. The specific form of the scalar fields in (14) is strongly inspired by Hurault et al. (2022a), with the addition of the $\boldsymbol{B}_k$ matrices introduced in our formulation. It is justified by the fact that

$$\nabla \psi_{\boldsymbol{\theta},\boldsymbol{B}_k}(\boldsymbol{y}) = \boldsymbol{B}_k^\top D_{\boldsymbol{\theta}}(\boldsymbol{y}) + \mathbf{J}_{D_{\boldsymbol{\theta}}}(\boldsymbol{y})^\top (\boldsymbol{B}_k \boldsymbol{y} - D_{\boldsymbol{\theta}}(\boldsymbol{y}))\,, \tag{15}$$

for which the first term is known to be effective for learning denoising functions. Note that the inclusion of the matrix $\boldsymbol{B}_k$ in (14) has the effect of introducing the term $\boldsymbol{B}_k^\top D_{\boldsymbol{\theta}}(\boldsymbol{y})$ instead of $D_{\boldsymbol{\theta}}(\boldsymbol{y})$, with the hope that this (transposed) matrix could counterbalance the potentially negative effect of multiplication by a skew-symmetric matrix $\boldsymbol{A}_k$ afterwards. In practice, expression (15) is computed by differentiating (14) with respect to the input $\boldsymbol{y}$ using an automatic differentiation engine (Paszke et al., 2019), which avoids computing the full Jacobian.

Finally, the matrices $\boldsymbol{B}_k$ in (14) are parameterized analogously to (13) via a shared repeated-block diagonal matrix $\boldsymbol{\Theta}' \in \mathbb{R}^{n \times n}$, in accordance with

$$\boldsymbol{B}_k = \boldsymbol{P}_k^\top \boldsymbol{\Theta}' \boldsymbol{P}_k\,. \tag{16}$$

Please note that the fixed permutation matrices $\boldsymbol{P}_k$ are the same as in (13). Ultimately, the learnable parameters for the proposed parametrization of (12) are $\{\boldsymbol{\theta}, \boldsymbol{\Theta}, \boldsymbol{\Theta}'\}$ and their number is only slightly greater than that of $D_{\boldsymbol{\theta}}$ since $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}'$ are sparse, which supports the practicality of our proposed parameterization.

5 Experimental Results

We demonstrate the effectiveness of our proposed methodology to construct divergence-free networks, termed DivFree, in the case of self-supervised image denoising under the assumption of Gaussian noise and compare its competitiveness with related state-of-the-art divergence-based approaches, namely MC-SURE (Soltanayev & Chun, 2018), Noise2Self (Batson & Royer, 2019) and UNSURE (Tachella et al., 2025a). We trained all models ourselves, with a separate model for each noise level σ. For divergence-free estimators, either everywhere or in expectation, we also evaluated the performance of their divergence-constant counterparts, marked with symbol †, based on Propositions 1 and 2. Performance of DivFree and other methods are assessed in terms of PSNR values.

5.1 Implementation details

Common backbone architecture For a fair comparison, we adopt a variant of the DRUNet architecture (Zhang et al., 2022) as the shared backbone across all approaches. In the original formulation, each scale is composed of four residual blocks of the form "3×3 conv $\rightarrow$ ReLU $\rightarrow$ 3×3 conv". To reduce computational cost, we limit this to two residual blocks per scale. Moreover, as in Hurault et al. (2022b), we replace ReLU by Softplus activations with sharpness parameter $\beta = 100$, which acts as a smooth surrogate for ReLU, easing training for convervative field networks (Hurault et al., 2022a). Note that only "blind" models were considered in this work, so the noise level map was removed from the original architecture.

Datasets All models were trained using the same large-scale dataset proposed in Zhang et al. (2022), which contains a total of 8,694 images. This includes 400 images from the Berkeley Segmentation Dataset (BSD400) (Martin et al., 2001), 4,744 images from the Waterloo Exploration Database (Ma et al., 2017), 900 images from DIV2K (Agustsson & Timofte, 2017), and 2,750 images from Flickr2K (Lim et al., 2017). The training set is augmented through random horizontal and vertical flips as well as random rotations of 90°. For validation, we use the BSD32 dataset (Martin et al., 2001), consisting of 32 images, to monitor training progress and select the best-performing model. Finally, the evaluation is carried out on two test sets, Set12 and BSD68 (Martin et al., 2001), which are completely separate from both the training and validation data.

Training details All models are trained for 600,000 iterations, where each training iteration involves a gradient-based pass on a batch of patches of size 128×128 that are randomly cropped from training images (except for DivFree where patches are taken of size 64×64 in order to accelerate training). We use a batch size of 16 and employ the Adam optimizer (Kingma & Ba, 2015) with an initial learning rate of 10^{-4} as in Zhang et al. (2022), which is halved every 150,000 iterations.

Table 1: The PSNR (dB) results of deep learning-based methods applied to popular grayscale datasets corrupted by synthetic white Gaussian noise with $\sigma = 15$, 25 and 50. Best in each category is in bold.

Dataset		Set12	BSD68
Noise level σ		15 / 25 / 50	15 / 25 / 50
supervised	DRUNet *light*	**33.24 / 30.92 / 27.84**	**31.91 / 29.46 / 26.55**
self-supervised **known** σ	MC-SURE	32.13 / 29.97 / **27.27**	**31.20** / 28.86 / **26.22**
	Noise2Self†	32.07 / 30.05 / 27.25	30.80 / 28.66 / 26.09
	UNSURE†	31.80 / 29.49 / 27.14	30.58 / 28.39 / 26.01
	DivFree† (ours)	**32.46 / 30.28 / 27.27**	**31.20 / 28.91** / 26.13
self-supervised **unknown** σ	Noise2Self	31.15 / 29.55 / 27.02	29.29 / 27.83 / 25.73
	UNSURE	**31.88 / 29.84 / 27.15**	**30.90 / 28.72 / 26.08**
	DivFree (ours)	31.65 / 29.81 / 27.05	29.87 / 28.14 / 25.78

Approaches that rely on a Monte Carlo approximation (7) of the divergence involve an additional hyperparameter τ. In our experiments, we adopted the default choice $\tau = 10^{-2}$, which is recommended for vectors with entries in $[0, 1]$ (Tachella et al., 2025b). For the UNSURE loss (Tachella et al., 2025a), we followed the default settings of the DeepInverse library (Tachella et al., 2025b); in particular, the momentum parameter for the gradient ascent on the noise level was fixed at 0.9.

Implementation choices for DivFree Our proposed parameterization (12) of divergence-free estimators requires several additional hyperparameters that must be specified, including the number of terms K' in the sum, the size $\kappa \times \kappa$ of the blocks in the two repeated-block diagonal matrices $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}'$, and the selection of the permutation matrices $\boldsymbol{P}_k$. First of all, in order to drastically reduce the computational burden, we set $K' = 8$. The block size κ is chosen as 16, resulting in a total of 512 additional learnable parameters, which is negligible compared to the 17,007,744 parameters of the network backbone. Finally, the first four permutation matrices $\boldsymbol{P}_k$ were selected to perform circular horizontal shifts of the input image by 0 to 3 pixels, while the remaining four are obtained by composing these shifts with a 90° rotation.

Our implementation is written in Python using the PyTorch framework (Paszke et al., 2019) and with additional support from the DeepInverse library (Tachella et al., 2025b). Training was conducted on a Tesla V100 GPU.

5.2 Results for self-supervised image denoising

Table 1 reports a quantitative comparison of state-of-the-art divergence-based methods for image denoising trained without ground truth data. The results are organized into two categories: methods that require the noise level σ during training, such as MC-SURE (Soltanayev & Chun, 2018), and those that are agnostic to it, including DivFree. Importantly, Propositions 1 and 2 show that the latter category can be converted into noise-level–aware denoisers without additional training. Specifically, for a divergence-free estimator f, either everywhere or in expectation, the quantity $\min_{f \in \mathcal{S}^0} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2$ in Proposition 2 can be approximated using a single realization $\|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2$, where $\boldsymbol{y}$ denotes the noisy input image (averaging this term over more image realizations did not lead to further improvements).

As expected, UNSURE (Tachella et al., 2025a) achieves the best performance when the noise level σ is unknown, followed by DivFree and then Noise2Self (Batson & Royer, 2019). This ordering is consistent with the fact that $\mathcal{S}^0_{\mathrm{BS}} \subset \mathcal{S}^0_{\mathrm{DC}} \subset \mathcal{S}^0_{\mathrm{CED}}$: the fewer constraints imposed on the search space of the estimator, the more expressive it becomes. The situation changes, however, when the noise level σ is assumed known. While DivFree and Noise2Self naturally benefit from this additional information—showing PSNR gains in line with theoretical expectations (see Subsection 3.4)—UNSURE exhibits degraded performance. We attribute this counterintuitive outcome to the fact that, unlike DivFree or Noise2Self where zero divergence everywhere is enforced by design, UNSURE does not strictly enforce it to be *exactly* zero in expectation. Instead, this property is only encour-

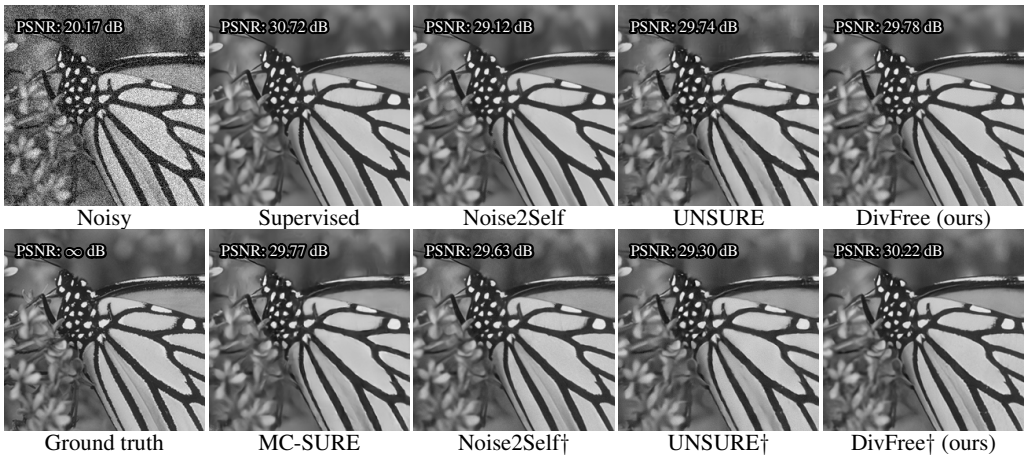

Figure 1: Image denoising results for $\sigma = 25$ on Monarch image (Set12). Best viewed by zooming.

aged through the loss function. This distinction between hard and soft constraints appears to play a decisive role in our setting.

Another noteworthy observation concerns the poor performance of MC-SURE. In principle, according to (4), a denoiser trained with the SURE loss should achieve performance comparable to its supervised counterpart. The observed underperformance can largely be attributed to stability issues during training. As illustrated in Figure 3 in Appendix, the training curves of MC-SURE and UNSURE exhibit pronounced fluctuations, in contrast to the much smoother trajectories of the other methods, including ours. This instability arises from the Monte Carlo approximations (7) employed to estimate the divergence during training. Because these estimates are obtained via random sampling, their variance propagates into the optimization, leading to noisy gradient updates. As a result, the PSNR values on the validation set oscillate significantly instead of following a stable, monotonic improvement, ultimately preventing the models from reaching their full potential.

Finally, our proposed method emerges as the most effective divergence-based approach for Gaussian noise removal when the noise level σ is known, outperforming MC-SURE in the majority of cases. This advantage is further corroborated by the qualitative results in Figure 1, with additional examples provided in the Appendix.

6 Conclusion

We presented an original approach for constraining neural networks to be divergence-free by design. Our proposed parameterization is grounded in a representer theorem for divergence-free vector fields, which characterizes them as structured combinations of conservative fields. Leveraging this theoretical foundation and incorporating sparsity constraints, we derived parameterizations for neural networks that are both resource-efficient and scalable to high-dimensional settings. The practical relevance of our approach is illustrated in the context of self-supervised image denoising, where we demonstrated that that these models achieve competitive performance compared to existing divergence-based methods, especially when the noise level is known. Beyond denoising, our results suggest that our divergence-free parameterization may hold promise for a wider range of high-dimensional learning tasks, in particular in physics-informed machine learning, opening new avenues for future research.

References

Eirikur Agustsson and Radu Timofte. NTIRE 2017 Challenge on single image super-resolution: Dataset and study. In Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1122–1131, 2017.

Kenneth Joseph Arrow, Leonid Hurwicz, Hirofumi Uzawa, Hollis Burnley Chenery, Selmer Johnson, and Samuel Karlin. Studies in linear and non-linear programming, volume 2. Stanford University Press Stanford, 1958.

Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In International Conference on Machine Learning (ICML), volume 97, pp. 524–533, 2019.

Marcel Berger. A panoramic view of Riemannian geometry. Springer, 2003.

D.P. Bertsekas. Nonlinear Programming. Athena Scientific optimization and computation series. Athena Scientific, 1995. ISBN 9781886529144.

Thierry Blu and Florian Luisier. The SURE-LET approach to image denoising. IEEE Transactions on Image Processing, 16(11):2778–2786, 2007.

Elie Cartan. On certain differential expressions and the Pfaff problem. In Annales Scientifiques de l'École Normale Supérieure, volume 3, pp. 239–332, 1899.

Dongdong Chen, Julián Tachella, and Mike E Davies. Robust equivariant imaging: a fully unsupervised framework for learning to image from noisy and partial measurements. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5647–5656, 2022.

Guangyong Chen, Fengyuan Zhu, and Pheng Ann Heng. An efficient statistical method for image noise level estimation. In Proceedings of the IEEE international conference on computer vision, pp. 477–485, 2015.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions on Image Processing, 16 (8):2080–2095, 2007.

Bradley Efron. The estimation of prediction error: covariance penalties and cross-validation. Journal of the American Statistical Association, 99(467):619–632, 2004.

Bradley Efron. Tweedie's formula and selection bias. Journal of the American Statistical Association, 106(496):1602–1614, 2011.

Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. IEEE Transactions on Image Processing, 17(10):1737–1754, 2008.

Jose Gallego-Posada, Juan Ramirez, Akram Erraqabi, Yoshua Bengio, and Simon Lacoste-Julien. Controlled sparsity via constrained optimization or: How I learned to stop tuning penalties and love constraints. Advances in Neural Information Processing Systems (NeurIPS), 35:1253–1266, 2022.

Sébastien Herbreteau and Charles Kervrann. A unified framework of nonlocal parametric methods for image denoising. SIAM Journal on Imaging Sciences, 18(1):89–119, 2025.

Eva Höck, Tim-Oliver Buchholz, Anselm Brachmann, Florian Jug, and Alexander Freytag. N2v2-fixing noise2void checkerboard artifacts with modified sampling strategies and a tweaked network architecture. In European Conference on Computer Vision (ECCV), pp. 503–518. Springer, 2022.

Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2Neighbor: Self-supervised denoising from single noisy images. In Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14781–14790, 2021.

Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Gradient step denoiser for convergent plug-and-play. In International Conference on Learning Representations (ICLR), 2022a.

Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization. In International Conference on Machine Learning (ICML), pp. 9483–9505. PMLR, 2022b.

Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations. Journal of Computational Physics, 426:109951, 2021.

Kwanyoung Kim and Jong Chul Ye. Noise2Score: Tweedie's approach to self-supervised image denoising without clean images. In Advances in Neural Information Processing Systems (NeurIPS), volume 34, pp. 864–874, 2021.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR), 2015.

Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void - Learning denoising from single noisy images. In Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2124–2132, 2019.

Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In Advances in Neural Information Processing Systems (NeurIPS), volume 32, 2019.

M. Lebrun, A. Buades, and J. M. Morel. A nonlocal Bayesian image denoising algorithm. SIAM Journal on Imaging Sciences, 6(3):1665–1688, 2013.

Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In International Conference on Machine Learning (ICML), volume 80, pp. 2965–2974, 2018.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1132–1140, 2017.

Jae Hyun Lim, Aaron Courville, Christopher Pal, and Chin-Wei Huang. AR-DAE: Towards unbiased neural entropy gradient estimation. In International Conference on Machine Learning (ICML), volume 119, pp. 6061–6071, 2020.

Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. IEEE Transactions on Image Processing, 26(2):1004–1016, 2017.

Youssef Mansour and Reinhard Heckel. Zero-Shot Noise2Noise: Efficient image denoising without any data. In Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14018–14027, 2023.

Zhiping Mao, Ameya D Jagtap, and George Em Karniadakis. Physics-informed neural networks for high-speed flows. Computer Methods in Applied Mechanics and Engineering, 360:112789, 2020.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In International Conference on Computer Vision (ICCV), volume 2, pp. 416–423 vol.2, 2001.

James Clerk Maxwell. A treatise on electricity and magnetism, volume 1. Clarendon press, 1873.

Shigeyuki Morita. Geometry of differential forms, volume 201. American Mathematical Society, 2001.

Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-Recorrupted: Unsupervised deep learning for image denoising. In Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2043–2052, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems (NeurIPS), volume 32, 2019.

John Platt and Alan Barr. Constrained differential optimization. In Neural information processing systems, 1987.

Stanislav Pyatykh, Jürgen Hesser, and Lei Zheng. Image noise level estimation by principal component analysis. IEEE Transactions on Image Processing, 22(2):687–699, 2013.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part II): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561, 2017a.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part I): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561, 2017b.

Sathish Ramani, Thierry Blu, and Michael Unser. Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms. IEEE Transactions on Image Processing, 17(9):1540–1554, 2008.

Jack Richter-Powell, Yaron Lipman, and Ricky TQ Chen. Neural conservation laws: A divergence-free perspective. Advances in Neural Information Processing Systems (NeurIPS), 35:38075–38088, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer Assisted Intervention (MICCAI), pp. 234–241, 2015.

Tim Salimans and Jonathan Ho. Should ebms model the energy or the score? In Energy based models workshop-ICLR, 2021.

Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. In Advances in Neural Information Processing Systems (NeurIPS), volume 31, 2018.

Charles M. Stein. Estimation of the mean of a multivariate normal distribution. The Annals of Statistics, 9(6):1135–1151, 1981.

Julián Tachella, Mike Davies, and Laurent Jacques. Unsure: Self-supervised learning with Unknown Noise level and Stein's Unbiased Risk Estimate. In International Conference on Learning Representation (ICLR), 2025a.

Julián Tachella, Matthieu Terris, Samuel Hurault, Andrew Wang, Dongdong Chen, Minh-Hai Nguyen, Maxime Song, Thomas Davies, Leo Davy, Jonathan Dong, Paul Escande, Johannes Hertrich, Zhiyuan Hu, Tobías I. Liaudat, Nils Laurent, Brett Levac, Mathurin Massias, Thomas Moreau, Thibaut Modrzyk, Brayan Monroy, Sebastian Neumayer, Jérémy Scanvic, Florian Sarron, Victor Sechaud, Georg Schramm, Romain Vo, and Pierre Weiss. Deepinverse: A python package for solving imaging inverse problems with deep learning. arXiv preprint arXiv:2505.20160, 2025b.

Dimitri Van De Ville and Michel Kocher. SURE-based Non-Local Means. IEEE Signal Processing Letters, 16(11):973–976, 2009.

Yi-Qing Wang and Jean-Michel Morel. SURE guided Gaussian mixture image denoising. SIAM Journal on Imaging Sciences, 6(2):999–1034, 2013.

Zejin Wang, Jiazheng Liu, Guoqing Li, and Hua Han. Blind2Unblind: Self-supervised image denoising with visible blind spots. In Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2027–2036, 2022.

Yutong Xie, Mingze Yuan, Bin Dong, and Quanzheng Li. Unsupervised image denoising with score function. Advances in Neural Information Processing Systems (NeurIPS), 36:69752–69763, 2023.

Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-Play image restoration with deep denoiser prior. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(10):6360–6376, 2022.

A LIMITATIONS

We would like to mention that the proposed parameterization for enforcing zero divergence entails higher computational cost during both training and inference compared to its supervised and divergence-based self-supervised counterparts (Tachella et al., 2025a; Soltanayev & Chun, 2018; Batson & Royer, 2019). In particular, our method requires one feedforward pass through the backbone network along with K' gradient computations at inference ($K' = 8$ in our implementation). Since each backpropagation has a cost comparable to a forward pass (Hurault et al., 2022a), the overall inference cost amounts to roughly $K' + 1$ times that of a single feedforward evaluation, which may limit its applicability in time-sensitive settings. Moreover, due to scalability constraints, we deliberately restricted the number of terms in the sum from Theorem 1. While this sparsity constraint may not fully capture the underlying optimal solution, we demonstrated that it still yields strong performance in image denoising. Finally, our application in image denoising targets only additive white Gaussian noise, and its effectiveness under alternative noise models remains unexplored. Extending our parameterization to handle other types of noise, such as Poisson–Gaussian corruption, presents a promising direction for future research.

B PROOFS

Proof of Lemma 1. $\mathcal{S}^0$ is a linear space due to the linearity of the partial derivative operator and the linearity of expectation. Let $f \in c\,\mathrm{id}_n + \mathcal{S}^0$. For all $\boldsymbol{y} \in \mathbb{R}^n$, $\frac{\partial [c\,\mathrm{id}_n]_i}{\partial y_i}(\boldsymbol{y}) = c$ and so $\operatorname{div}(c\,\mathrm{id}_n)(\boldsymbol{y}) = nc$. Therefore, $f \in \mathcal{S}^c$. Reciprocally, let $f \in \mathcal{S}^c$. Then, $f = c\,\mathrm{id}_n + (f - c\,\mathrm{id}_n) \in c\,\mathrm{id}_n + \mathcal{S}^0$ by linearity of the partial derivative operator and the linearity of expectation. □

Proof of Proposition 1. According to (4) (Stein, 1981),

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 = \mathbb{E}_{\boldsymbol{y}} \left[-n\sigma^2 + \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 + 2\sigma^2 \operatorname{div}(f)(\boldsymbol{y}) \right] .$$

Hence,

$$\arg\min_{f \in \mathcal{S}^c} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 = \arg\min_{f \in \mathcal{S}^c} \mathbb{E}_{\boldsymbol{y}} \left[-n\sigma^2 + \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 + 2\sigma^2 nc \right] = \arg\min_{f \in \mathcal{S}^c} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 .$$

According to Lemma 1, $\mathcal{S}^c = c\,\mathrm{id} + \mathcal{S}^0$. In particular, if $c \neq 1$, $\mathcal{S}^c = c\,\mathrm{id} + (1-c)\mathcal{S}^0$ and we have

$$\begin{aligned}
\arg\min_{f \in \mathcal{S}^c} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 &= \arg\min_{f \in c\,\mathrm{id} + (1-c)\mathcal{S}^0} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 \\
&= c\,\mathrm{id} + (1-c) \arg\min_{f \in \mathcal{S}^0} \mathbb{E}_{\boldsymbol{y}} \|c\,\mathrm{id}(\boldsymbol{y}) + (1-c) f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 \\
&= c\,\mathrm{id} + (1-c) \arg\min_{f \in \mathcal{S}^0} (1-c)^2 \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 \\
&= c\,\mathrm{id} + (1-c) \arg\min_{f \in \mathcal{S}^0} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 .
\end{aligned}$$

For $c = 1$, we have also trivially

$$\arg\min_{f \in \mathcal{S}^c} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 = \mathrm{id} = c\,\mathrm{id} + (1-c) \arg\min_{f \in \mathcal{S}^0} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 .$$

□

Proof of Proposition 2. According to (4) (Stein, 1981),

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 = \mathbb{E}_{\boldsymbol{y}} \left[-n\sigma^2 + \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 + 2\sigma^2 \operatorname{div}(f)(\boldsymbol{y}) \right] .$$

Hence,

$$\min_{f \in \mathcal{S}^c} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 = -n\sigma^2 + 2\sigma^2 nc + \min_{f \in \mathcal{S}^c} \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2$$

According to Lemma 1, $\mathcal{S}^c = c\,\mathrm{id} + \mathcal{S}^0$. In particular, if $c \neq 1$, $\mathcal{S}^c = c\,\mathrm{id} + (1-c)\mathcal{S}^0$ and we have

$$\begin{aligned}\min_{f\in\mathcal{S}^c}\mathbb{E}_{\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{y}\|_2^2 &= \min_{f\in c\,\mathrm{id}+(1-c)\mathcal{S}^0}\mathbb{E}_{\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{y}\|_2^2\\ &= \min_{f\in\mathcal{S}^0}\mathbb{E}_{\boldsymbol{y}}\|c\boldsymbol{y}+(1-c)f(\boldsymbol{y})-\boldsymbol{y}\|_2^2\\ &= (1-c)^2\min_{f\in\mathcal{S}^0}\mathbb{E}_{\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{y}\|_2^2\,.\end{aligned}$$

For $c=1$, by considering the identity function $\mathrm{id}\in\mathcal{S}^1$, we also have

$$\min_{f\in\mathcal{S}^c}\mathbb{E}_{\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{y}\|_2^2 = 0 = (1-c)^2\min_{f\in\mathcal{S}^0}\mathbb{E}_{\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{y}\|_2^2\,.$$

Finally, for all $c\in\mathbb{R}$,

$$\min_{f\in\mathcal{S}^c}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{x}\|_2^2 = -n\sigma^2+2\sigma^2nc+(1-c)^2\min_{f\in\mathcal{S}^0}\mathbb{E}_{\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{y}\|_2^2\,,$$

As a consequence,

$$\arg\min_{c\in\mathbb{R}}\min_{f\in\mathcal{S}^c}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{x}\|_2^2 = 1-\frac{n\sigma^2}{\min_{f\in\mathcal{S}^0}\mathbb{E}_{\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{y}\|_2^2}\,.$$

But this latter quantity lies in $[0,1]$ since, in particular for $c=0$,

$$\min_{f\in\mathcal{S}^0}\mathbb{E}_{\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{y}\|_2^2 = n\sigma^2+\min_{f\in\mathcal{S}^0}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{x}\|_2^2 > 0\,.$$

and so,

$$\frac{n\sigma^2}{\min_{f\in\mathcal{S}^0}\mathbb{E}_{\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{y}\|_2^2} = \frac{n\sigma^2}{n\sigma^2+\min_{f\in\mathcal{S}^0}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\|f(\boldsymbol{y})-\boldsymbol{x}\|_2^2}\in[0,1]\,.$$

□

Proof of Lemma 2. Let $f:\boldsymbol{x}\mapsto\boldsymbol{A}\nabla\psi(\boldsymbol{x})$. We want to compute $\operatorname{div}f(\boldsymbol{x})=\operatorname{tr}(\nabla f(\boldsymbol{x}))$ for all $\boldsymbol{x}\in\mathbb{R}^n$. We have $f=\varphi_2\circ\varphi_1$, with for all $\boldsymbol{x}\in\mathbb{R}^n$,

$\varphi_1(\boldsymbol{x})=\nabla\psi(\boldsymbol{x})$	$\nabla\varphi_1(\boldsymbol{x})=\boldsymbol{H}_\psi(\boldsymbol{x})$
$\varphi_2(\boldsymbol{x})=\boldsymbol{A}\boldsymbol{x}$	$\nabla\varphi_2(\boldsymbol{x})=\boldsymbol{A}^\top$

where $\boldsymbol{H}_\psi(\boldsymbol{x})$ denotes the Hessian matrix of ψ evaluated at $\boldsymbol{x}$. According to the chain rule (Bertsekas, 1995),

$$\nabla f(\boldsymbol{x})=\nabla\varphi_1(\boldsymbol{x})\nabla\varphi_2(\varphi_1(\boldsymbol{x}))=\boldsymbol{H}_\psi(\boldsymbol{x})\boldsymbol{A}^\top\,,$$

hence $\operatorname{div}f(\boldsymbol{x})=\operatorname{tr}(\boldsymbol{H}_\psi(\boldsymbol{x})\boldsymbol{A}^\top)=-\operatorname{tr}(\boldsymbol{A}\boldsymbol{H}_\psi(\boldsymbol{x}))=0$. Indeed, the trace of the product of a skew-symmetric and a symmetric matrix is zero:

$$\operatorname{tr}(\boldsymbol{A}\boldsymbol{B})=\operatorname{tr}((\boldsymbol{A}\boldsymbol{B})^\top)=\operatorname{tr}(\boldsymbol{B}^\top\boldsymbol{A}^\top)=-\operatorname{tr}(\boldsymbol{B}\boldsymbol{A})=-\operatorname{tr}(\boldsymbol{A}\boldsymbol{B}).$$

where $\boldsymbol{B}\in\mathbb{R}^{n\times n}$ denotes a symmetric matrix.

□

Proof of Theorem 1. First of all, from Lemma 2, we have that each $\boldsymbol{A}_k\nabla\psi_k$ is divergence-free. Since the set of divergence-free functions is a linear space from Lemma 1, $\tilde{f}=\sum_{k=1}^K\boldsymbol{A}_k\nabla\psi_k$ is divergence-free.

Moreover, we know from Richter-Powell et al. (2022) that there exists $F:\mathbb{R}^n\to\mathbb{R}^{n\times n}$ a skew-symmetric matrix field[1] such that the function $\tilde{f}:\mathbb{R}^n\to\mathbb{R}^n$ defined component-wise as

$$\forall i\in\{1,\dots n\},\quad \tilde{f}_i=\sum_{j=1}^n\frac{\partial F_{i,j}}{\partial x_j}$$

[1] *i.e.* a map $F:\mathbb{R}^n\to\mathbb{R}^{n\times n}$ such that $\forall\boldsymbol{x}\in\mathbb{R}^n$, $F(\boldsymbol{x})^\top=-F(\boldsymbol{x})$.

can approximate f "arbitrarily well". By denoting $\boldsymbol{E}^{(i,j)}$ the standard basis matrix of $\mathbb{R}^{n\times n}$, having a 1 in the (i,j)-th entry and zeros elsewhere, we can rewrite it as

$$\begin{aligned}
\tilde{f} = \sum_{i,j} \boldsymbol{E}^{(i,j)} \nabla F_{i,j} &= \sum_{i<j} \boldsymbol{E}^{(i,j)} \nabla F_{i,j} + \sum_{j<i} \boldsymbol{E}^{(i,j)} \nabla F_{i,j} + \sum_{i} \boldsymbol{E}^{(i,i)} \nabla F_{i,i} \\
&= \sum_{i<j} \boldsymbol{E}^{(i,j)} \nabla F_{i,j} + \sum_{i<j} \boldsymbol{E}_{j,i} \nabla F_{j,i} \\
&= \sum_{i<j} (\boldsymbol{E}^{(i,j)} - \boldsymbol{E}^{(j,i)}) \nabla F_{i,j} \\
&= \sum_{i<j} (\boldsymbol{E}^{(i,j)} - \boldsymbol{E}^{(i,j)\top}) \nabla F_{i,j} \\
&= \sum_{k=1}^{K} (\boldsymbol{E}^{\varphi(k)} - \boldsymbol{E}^{\varphi(k)\top}) \nabla F_{\varphi(k)} \\
&= \sum_{k=1}^{K} \boldsymbol{B}_k \nabla F_{\varphi(k)} \,,
\end{aligned}$$

where φ is a bijection from $\{1,\dots,\binom{n}{2}\}$ to $\{(i,j)\in\{1,\dots,n\}^2 | i<j\}$ and $\boldsymbol{B}_k \triangleq \boldsymbol{E}^{\varphi(k)} - \boldsymbol{E}^{\varphi(k)\top}$.

We can notice that $(\boldsymbol{B}_1,\dots,\boldsymbol{B}_K) \in \mathbb{R}^{n\times n}$ is nothing else than the canonical basis of the space of real skew-symmetric $n\times n$ matrices. Therefore, $\forall 1 \le k \le K$, $\exists \lambda_1^{(k)},\dots,\lambda_K^{(k)}$,

$$\boldsymbol{B}_k = \sum_{i=1}^{K} \lambda_i^{(k)} \boldsymbol{A}_i \,.$$

Hence,

$$\tilde{f} = \sum_{k=1}^{K} \left(\sum_{i=1}^{K} \lambda_i^{(k)} \boldsymbol{A}_i \right) \nabla F_{\varphi(k)} = \sum_{i=1}^{K} \boldsymbol{A}_i \left(\sum_{k=1}^{K} \lambda_i^{(k)} \nabla F_{\varphi(k)} \right) = \sum_{i=1}^{K} \boldsymbol{A}_i \nabla \left(\sum_{k=1}^{K} \lambda_i^{(k)} F_{\varphi(k)} \right) \,,$$

We conclude by setting $\psi_i = \sum_{k=1}^{K} \lambda_i^{(k)} F_{\varphi(k)}$. □

C Divergence-free vs. Blind-spot

Divergence-free estimators provide greater expressiveness than their blind-spot counterparts (Batson & Royer, 2019), as they are not restricted by architectural masking strategies. Blind-spot networks, by construction, enforce their receptive field to exclude the center pixel y_i when estimating x_i, since the component function f_i cannot depend on y_i by definition. In contrast, divergence-free networks can exploit the full image context without masking, including the central pixel y_i, which is typically highly informative about x_i. This difference in expressiveness is illustrated in the toy image-denoising example of Figure 2, where Noise2Self removes all isolated white data points, whereas a divergence-free estimator can preserve them more accurately, leading to improved denoising performance.

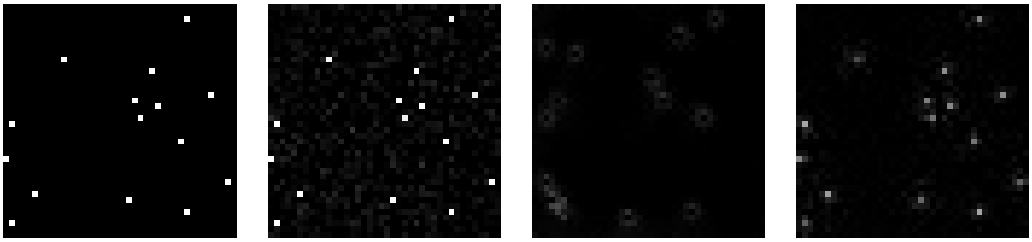

Figure 2: Divergence-free estimators are more expressive than their blind-spot counterparts.

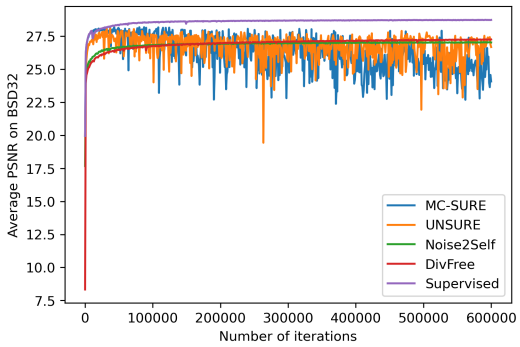

Figure 3: Stability issues during training for Monte Carlo approximations methods ($\sigma = 25$).

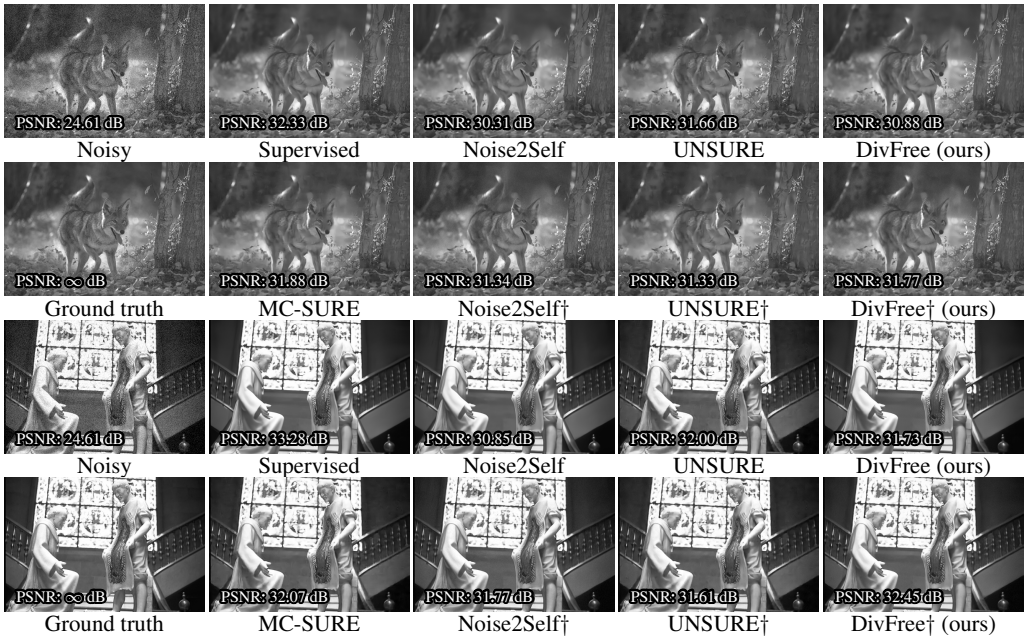

Figure 4: Image denoising results for $\sigma = 15$. Best viewed by zooming.

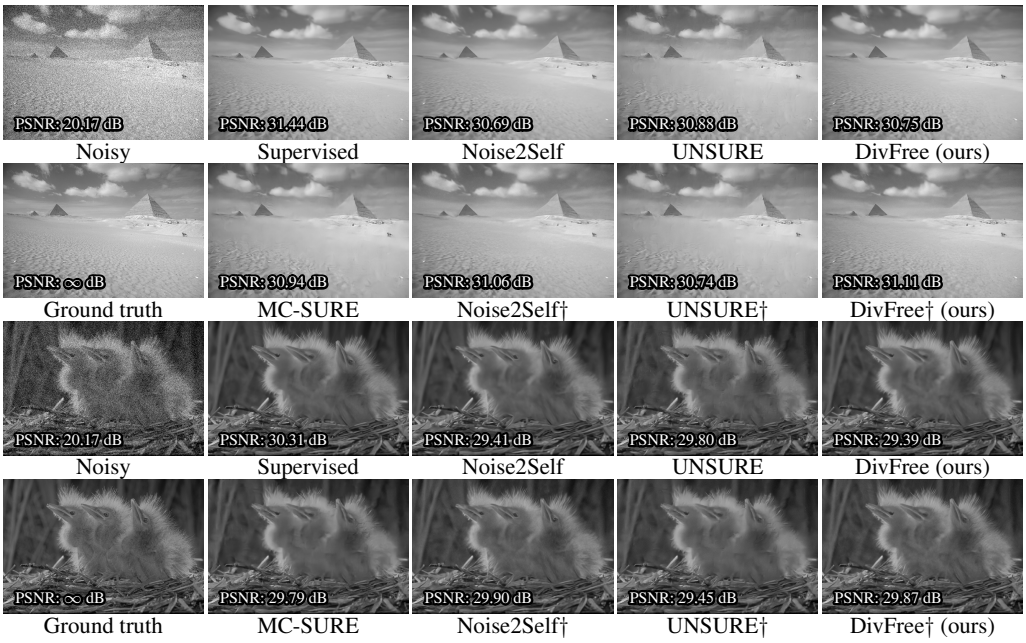

Figure 5: Image denoising results for $\sigma = 25$. Best viewed by zooming.

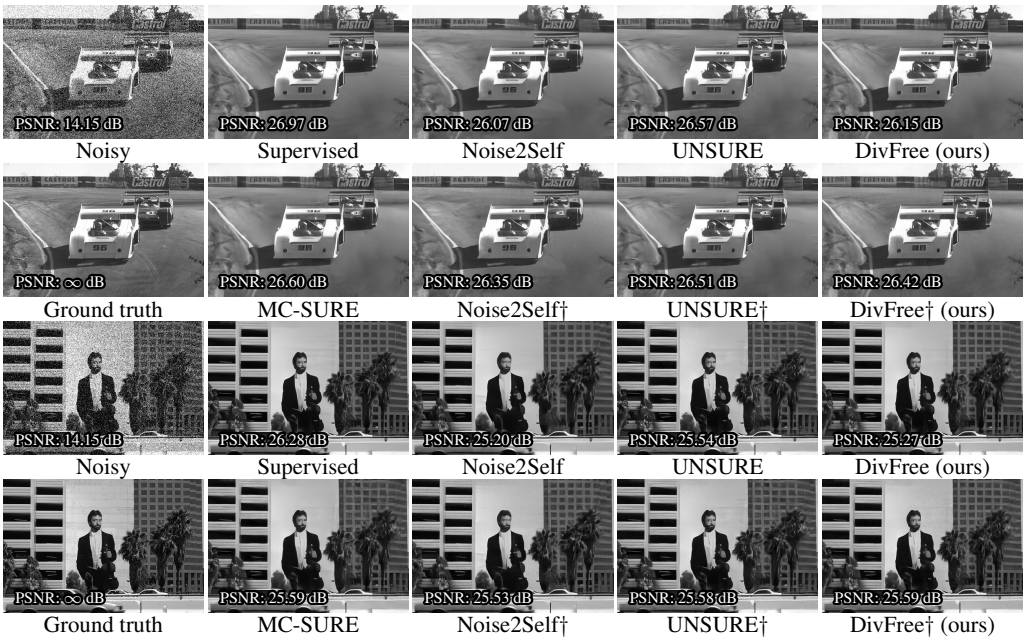

Figure 6: Image denoising results for $\sigma = 50$. Best viewed by zooming.