COMPACT REPRESENTATIONS OF DIVERGENCE-FREE FIELDS FOR SELF-SUPERVISED IMAGE DENOISING

Anonymous authors
Paper under double-blind review

ABSTRACT

We show that divergence-free vector fields admit parsimonious representations, thus extending the classical curl representation beyond three dimensions. Building on this foundation, we propose a novel computationally efficient network architecture that enforces zero divergence by design. Experiments on image denoising—a setting where divergence-free fields are amenable to learning without ground-truth data—demonstrate that our approach sets a new benchmark for constraint-based methods under Gaussian noise, as well as Poisson and real-world noise with minor architectural modifications, bridging the performance gap with leading denoisers.

1 INTRODUCTION

Many physical phenomena are governed by conservation laws that naturally give rise to vector fields $f : \mathbb{R}^n \to \mathbb{R}^n$ with zero divergence, *i.e.*, verifying the following partial differential equation:

$$\forall \boldsymbol{y} \in \mathbb{R}^n, \quad \operatorname{div} f(\boldsymbol{y}) \triangleq \operatorname{tr}(\boldsymbol{J}_f(\boldsymbol{y})) = \sum_{i=1}^{n} \partial_i f_i(\boldsymbol{y}) = 0\,, \tag{E$_0$}$$

where $\boldsymbol{J}_f(\boldsymbol{y})$ denotes the Jacobian matrix of f at $\boldsymbol{y}$. Examples include incompressible fluid flows (Batchelor, 2000) and magnetic fields, the latter satisfying Gauss's law for magnetism (Griffiths, 2023). Motivated by these applications, divergence-free vector fields have become an important modeling tool in machine learning (Richter-Powell et al., 2022; Raissi et al., 2017a), where embedding physical constraints into neural networks can improve both accuracy and interpretability.

A common strategy for constructing such models exploits analytical characterizations of divergence-free fields. In three dimensions, any curl field is divergence-free (Batchelor, 2000). However, how to generalize this representation beyond three dimensions remained an open question in the literature (Barbarosie, 2011). Although constructions can be obtained through differential forms (Cartan, 1899; Richter-Powell et al., 2022), in practice, these parameterizations require evaluating many partial derivatives via automatic differentiation (Paszke et al., 2019). As the dimension grows, this computation becomes increasingly expensive, limiting their applicability (Richter-Powell et al., 2022).

In this paper, we establish that divergence-free vector fields admit compact representations involving only a limited number of partial derivatives, thereby truly generalizing the curl representation and alleviating the computational challenges in high dimensions. Building on these theoretical results, we derive a neural network parameterization that is computationally efficient, scalable to high dimensions, and divergence-free by construction. Unlike existing implementations of divergence-free networks, which are either difficult to scale (Richter-Powell et al., 2022) or deliberately constrained in their expressivity (Herbreteau & Meunier, 2026; Jarraya et al., 2024; Benaïchouche et al., 2022), our approach preserves high expressive power while introducing only a marginal computational overhead compared to more restrictive blind-spot architectures (Batson & Royer, 2019; Krull et al., 2019). We demonstrate the effectiveness of our approach on self-supervised image denoising under Gaussian noise, a setting where divergence-free estimators naturally arise through Stein's unbiased risk estimation (Stein, 1981). Furthermore, we naturally extend the framework to Poisson noise, broadening its applicability to real-world imaging scenarios. The proposed method outperforms existing constraint-based denoising approaches (Herbreteau & Meunier, 2026; Batson & Royer, 2019; Tachella et al., 2025a) and brings their performance substantially closer to that of leading unsupervised methods (Huang et al., 2021; Kim & Ye, 2021). In summary, the contributions of our work are as follows:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1. The establishment of two novel representation theorems for divergence-free vector fields, with compact parameterizations suitable for learning in high-dimensional settings.
2. The introduction of a simple and computationally efficient neural network architecture, guaranteeing zero divergence by design, with high expressive power.
3. The demonstration of the effectiveness of the proposed approach for self-supervised image denoising under Gaussian, Poisson, and real-world noise.

2 Related work

Learning divergence-free fields Soft-constrained methods promote divergence-free solutions by introducing regularization terms into the training objective (Raissi et al., 2017b; Mao et al., 2020; Jin et al., 2021). While computationally convenient, these approaches only enforce the constraint approximately and may leave non-zero divergence. To achieve exact constraint satisfaction, recent works have focused on architectural parameterizations that impose zero divergence by construction. In two dimensions, Raissi et al. (2017a) leverage a stream function representation (Batchelor, 2000) expressing the velocity field as derivatives of a scalar neural network output. In arbitrary dimensions, the representation of Richter-Powell et al. (2022) relies on anti-symmetric matrix fields. Equivalently, this can be written in the form of a sum of conservative fields acted upon by anti-symmetric operators (Herbreteau & Meunier, 2026). However, the number of terms required in this representation scales quadratically with the dimension. Practical implementations therefore typically retain only one term (Jarraya et al., 2024; Benaïchouche et al., 2022) or a small number of terms (Herbreteau & Meunier, 2026), reducing computational complexity at the potential expense of expressive capacity.

Self-supervised image denoising Noise2Noise (Lehtinen et al., 2018) established that training on independent pairs of noisy images is theoretically equivalent to learning with clean targets. Subsequent work removed the requirement for paired noisy measurements by generating surrogate noisy pairs from a single observation through subsampling or synthetic noise injection (Huang et al., 2021; Wang et al., 2022; Pang et al., 2021; Moran et al., 2020; Mansour & Heckel, 2023; Zou et al., 2023; Monroy et al., 2025). In parallel, constraint-based methods (Tachella et al., 2025a; Herbreteau & Meunier, 2026), including blind-spot approaches (Batson & Royer, 2019; Krull et al., 2019; Laine et al., 2019; Jang et al., 2023), learn with noisy images as targets by preventing the identity mapping through constraints most often implicitly encoded in the network architecture. Orthogonally, score-based approaches (Kim & Ye, 2021; Kim et al., 2022; Xie et al., 2023) have leveraged Tweedie's formula (Efron, 2011) for self-supervised denoising. Finally, dataset-free learning has been explored in Ulyanov et al. (2018); Quan et al. (2020); Herbreteau & Unser (2025). We refer the reader to Tachella & Davies (2026) for a comprehensive survey of self-supervised image denoising.

3 Theory

Existing representation theorems for divergence-free vector fields (Richter-Powell et al., 2022; Herbreteau & Meunier, 2026) are not parsimonious. Indeed, each component function f_i is essentially expressed as the sum of numerous partial derivatives. For example, the matrix-field representation proposed by Richter-Powell et al. (2022) writes each f_i as the divergence of a suitably defined vector field, which involves a sum of n partial derivatives. This lack of parsimony poses some serious issues of scalability in high dimensions, as the expressions typically either rely on computing a full Jacobian matrix (Richter-Powell et al., 2022), or $O(n^2)$ gradients (Herbreteau & Meunier, 2026), which is prohibitive in practice. In what follows, we show how divergence-free fields can be equivalently represented in a minimal manner, thereby enabling a truly resource-efficient evaluation.

3.1 Divergence-free fields have $n-1$ degrees of freedom

We aim to construct a smooth vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ satisfying (E_0). A key observation is that one may prescribe $n-1$ components of f freely. To make this explicit, we choose any $i \in \{1, \dots, n\}$ and fix the components f_j for $j \neq i$. Without loss of generality, we express them as

$$f_j = \partial_i g_j \,, \tag{1}$$

where $g_j : \mathbb{R}^n \to \mathbb{R}$ are scalar fields (g_j is nothing else than a primitive of f_j with regard to x_i). Since f is divergence-free, we have necessarily that f_i is such that

$$\partial_i f_i = -\sum_{j \neq i} \partial_j f_j = -\sum_{j \neq i} \partial_j \partial_i g_j = \partial_i \left(-\sum_{j \neq i} \partial_j g_j \right) , \tag{2}$$

using the symmetry of second derivatives (Rudin, 1976). Integrating with respect to x_i, we obtain that f_i must take the following form, up to an additive term independent of x_i, which we omit as it can be absorbed into the g_j's:

$$f_i = -\sum_{j \neq i} \partial_j g_j . \tag{3}$$

This construction shows that the divergence-free condition removes exactly one degree of freedom: only f_i is constrained via (3) while the remaining components can be chosen freely. The following theorem generalizes this result (all the proofs of this paper are given in Appendix E). In particular, the above construction is recovered by taking $\boldsymbol{v} = \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is the i-th canonical basis vector.

Theorem 1. *Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ and $\boldsymbol{v} \in \mathbb{R}^n \backslash \{\mathbf{0}\}$, with $n \geq 2$. f is divergence-free if and only if there exists $g \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ such that $\forall \boldsymbol{x} \in \mathbb{R}^n$,*

$$f(\boldsymbol{x}) = \boldsymbol{J}_g(\boldsymbol{x})\boldsymbol{v} - \operatorname{div} g(\boldsymbol{x})\boldsymbol{v} . \tag{4}$$

We may draw a parallel with the subspace of zero-sum vectors in $\mathbb{R}^n$: $\mathcal{Z} = \{\boldsymbol{z} \in \mathbb{R}^n \mid \langle \mathbf{1}, \boldsymbol{z} \rangle = 0\}$, where $\mathbf{1}$ denotes the all-ones vector of $\mathbb{R}^n$. Given any $\boldsymbol{x} \in \mathbb{R}^n$ and any $\boldsymbol{u} \in \mathbb{R}^n \backslash \mathcal{Z}$, the vector $\langle \mathbf{1}, \boldsymbol{u} \rangle \, \boldsymbol{x} - \langle \mathbf{1}, \boldsymbol{x} \rangle \, \boldsymbol{u}$ belongs to $\mathcal{Z}$. In particular, when $\boldsymbol{u} = \mathbf{1}/n$, this construction yields the orthogonal projection of $\boldsymbol{x}$ onto $\mathcal{Z}$: $\arg\min_{\boldsymbol{z} \in \mathcal{Z}} \|\boldsymbol{z} - \boldsymbol{x}\|_2^2 = \boldsymbol{x} - \langle \mathbf{1}, \boldsymbol{x} \rangle \, \mathbf{1}/n$. More generally, this characterization can be summarized as $\mathcal{Z} = \operatorname{Im}(\langle \mathbf{1}, \boldsymbol{u} \rangle \boldsymbol{I} - \boldsymbol{u}\mathbf{1}^\top)$. Theorem 1 provides a direct analogue of this characterization for divergence-free vector fields. Indeed, introducing the differential operator $\mathcal{D}_{\boldsymbol{v}}(g) = \boldsymbol{J}_g \boldsymbol{v} - (\operatorname{div} g)\boldsymbol{v}$ with $\boldsymbol{v} \neq \mathbf{0}$, we obtain $\{f \mid \operatorname{div} f = 0\} = \operatorname{Im}(\mathcal{D}_{\boldsymbol{v}})$. Choosing, for instance, $\boldsymbol{v} = \boldsymbol{e}_i$ in Theorem 1 yields a sparse representation of divergence-free vector fields, in which all but one component are parameterized by a single partial derivative. However, this representation is intrinsically asymmetric, which may limit its suitability for learning applications.

3.2 A COMPACT AND WELL-BALANCED REPRESENTATION OF DIVERGENCE-FREE FIELDS

Returning to our analogy with zero-sum vectors, we observe that there exists a construction of $\mathcal{Z}$ that remains sparse while distributing the roles of the coordinates more evenly. Let π be a cyclic permutation of $\{1, \ldots, n\}$, and let $\boldsymbol{P}_\pi$ denote its associated permutation matrix[1]. Then, $\mathcal{Z} = \operatorname{Im}(\boldsymbol{I} - \boldsymbol{P}_\pi)$. The following theorem extends this characterization to divergence-free fields.

Theorem 2. *Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ and π be a cyclic permutation of $\{1, \ldots, n\}$, with $n \geq 2$. f is divergence-free if and only if there exists $g \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ such that $\forall \boldsymbol{x} \in \mathbb{R}^n$,*

$$f(\boldsymbol{x}) = \operatorname{diag}(\boldsymbol{J}_g(\boldsymbol{x})\boldsymbol{P}_\pi) - \boldsymbol{P}_\pi \operatorname{diag}(\boldsymbol{J}_g(\boldsymbol{x})) , \tag{5}$$

where $\boldsymbol{P}_\pi$ denotes the permutation matrix associated to π. In other words, $\forall i \in \{1, \ldots, n\}$,

$$f_i = \partial_{\pi(i)} g_i - \partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)} . \tag{6}$$

In contrast to Theorem 1, the characterization (6) exhibits a more balanced structure, while remaining parsimonious. In particular, each component f_i depends exactly on two partial derivatives, which makes this formulation particularly suitable for learning purposes.

Remarks As a direct consequence of Theorem 2, we obtain the following two results. 1) In the case where $n = 3$ with $\pi(i) = i + 1$ (with modulo), f is divergence-free if and only if there exists $g \in C^\infty(\mathbb{R}^3, \mathbb{R}^3)$ such that $f = (\partial_2 g_1 - \partial_3 g_3, \partial_3 g_2 - \partial_1 g_1, \partial_1 g_3 - \partial_2 g_2)^\top = \nabla \times (g_2, g_3, g_1)^\top$. In other words, there exists a vector field such that f is its curl. This is a well-known result in the literature (Morita, 2001). 2) We recover that any blind-spot function b (Batson & Royer, 2019) is divergence-free. Define, for all i, g_i to be a primitive of b_i with regard to $x_{\pi(i)}$ which does not depend on x_i (always exists since b_i does not depend on x_i by definition). We have $\partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)} = 0$ and so $b_i = \partial_{\pi(i)} g_i - \partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)}$, ensuring that b is indeed divergence-free. Recovering this result using the representations proposed by Herbreteau & Meunier (2026); Richter-Powell et al. (2022) is less straightforward.

[1] With the convention, $\boldsymbol{P}_\pi = (p_{ij})$ where $p_{ij} = 1$ when $i = \pi(j)$ and $p_{ij} = 0$ otherwise.

4 DIVERGENCE-FREE ARCHITECTURE

Our goal is to construct a parameterized function $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$, implemented as a neural network, that is divergence-free by design, resource-efficient, and tailored to image denoising tasks. To this end, we build on Theorem 2. The problem therefore reduces to finding a parameterization of g that is both universal in the approximation-theoretic sense (Cybenko, 1989; Hornik et al., 1989) and compatible with (5) computationally. Unfortunately, for a general function $g \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ (e.g., a standard deep neural architecture such as a U-Net (Ronneberger et al., 2015)), there is no straightforward way to efficiently compute the quantities appearing in (5). Indeed, although automatic differentiation engines (Paszke et al., 2019) readily provide access to partial derivatives, they do not offer an efficient mechanism for extracting only the relevant diagonal entries of the Jacobian. In practice, this typically requires forming the full Jacobian matrix, thereby negating the computational advantages of the proposed representation.

As an alternative to full Jacobian evaluation, we observe that the quantities in (5) can be recovered simultaneously from a single gradient computation. Define

$$\Psi(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{n} g_i \left(\boldsymbol{x} + (\alpha_i - x_i)\boldsymbol{e}_i + (\beta_i - x_{\pi(i)})\boldsymbol{e}_{\pi(i)} \right), \tag{7}$$

then we have

$$\nabla_{(\boldsymbol{\alpha},\boldsymbol{\beta})} \Psi(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})\big|_{(\boldsymbol{\alpha},\boldsymbol{\beta})=(\boldsymbol{x},\boldsymbol{P}_\pi^{-1}\boldsymbol{x})} = \left(\operatorname{diag}(\boldsymbol{J}_g(\boldsymbol{x})), \operatorname{diag}(\boldsymbol{J}_g(\boldsymbol{x})\boldsymbol{P}_\pi)\right). \tag{8}$$

While this construction avoids explicit computation of the full Jacobian, it comes nevertheless at a computational cost: evaluating Ψ requires n separate evaluations of g (one per entry where α_i and β_i are substituted), which becomes prohibitive in high dimensions. Since backpropagation has a cost comparable to a forward pass (Hurault et al., 2022), this approach is not necessarily more efficient than a direct Jacobian evaluation. However, (8) remains useful under additional structure on g.

Assumption 1. $\exists m \in \mathbb{N}, \exists h \in C^\infty(\mathbb{R}^{m+2}, \mathbb{R}), \forall i \in \{1, \dots, n\}, \exists b_{1,i}, \dots, b_{m,i} \in C^\infty(\mathbb{R}^n, \mathbb{R})$,

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \quad g_i(\boldsymbol{x}) = h(x_i, x_{\pi(i)}, b_{1,i}(\boldsymbol{x}), \dots, b_{m,i}(\boldsymbol{x})), \tag{9}$$

where each $b_{j,i}$ does not depend on x_i or $x_{\pi(i)}$ (we say that $b_j : \mathbb{R}^n \to \mathbb{R}^n$ is π-blind-spot).

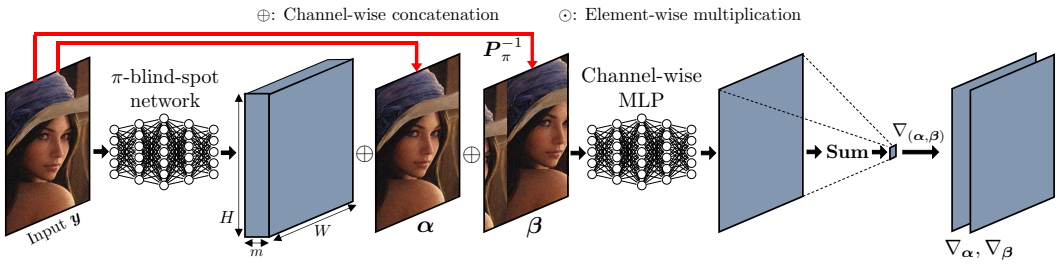
Under Assumption 1, the expression (7) can be rewritten as

$$\Psi(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \langle \mathbf{1},\ h\left([\boldsymbol{\alpha} \quad \boldsymbol{\beta} \quad b_1(\boldsymbol{x}) \quad \dots \quad b_m(\boldsymbol{x})]\right) \rangle, \tag{10}$$

where h is applied row-wise and $\mathbf{1}$ denotes the all-ones vector in $\mathbb{R}^n$. Thus, provided that $m \ll n$ and that h and the m π-blind-spot functions $b_1, \dots, b_m : \mathbb{R}^n \to \mathbb{R}^n$ can be evaluated efficiently, computing (10) remains tractable and the desired quantities in (5) can be obtained via a single additional gradient evaluation using (8).

Assumption 1 enforces that each component g_i depends on a pair of variables of interest, namely x_i and $x_{\pi(i)}$, while all remaining high-dimensional information is compressed into m auxiliary features: $b_{1,i}(\boldsymbol{x}), \dots, b_{m,i}(\boldsymbol{x})$. Although particularly appealing from a computational standpoint, one may wonder whether it is realistic in practice. In fact, a large class of denoising models exhibit a similar structure, up to boundary effects. For instance, in the special case where each $b_{j,i}$ extracts a single variable, Assumption 1 encompasses translation-invariant local functions, for which the i-th component depends only on the local neighborhood of x_i. This latter structural property is actually shared by a wide range of classical patch-based denoisers (Dabov et al., 2007; Buades et al., 2005; Yu & Sapiro, 2011), as well as by numerous denoising neural networks (Burger et al., 2012; Zhang et al., 2017; Herbreteau & Kervrann, 2022). Moreover, with adequate choice for h, Assumption 1 can also be seen as a generalization of the polynomial form proposed by Cha & Moon (2019; 2018). Finally, when $h(x_1, \dots, x_{m+2}) = x_2 x_3$, we get $f = b_1$ from (6), ensuring that our parameterization includes the blind-spot networks (Batson & Royer, 2019; Krull et al., 2019) with extra independence on $x_{\pi(i)}$ of each component.

We now detail the construction of both types of parameterizations, namely functions b_j and h. An overview of the proposed parametrization is shown in Figure 1.



Output: • $\nabla_{\boldsymbol{\beta}} - \boldsymbol{P}_\pi \nabla_{\boldsymbol{\alpha}}$ to satisfy (E_0),
• $(\boldsymbol{P}_\pi^{-1}\boldsymbol{y}) \odot \nabla_{\boldsymbol{\beta}} - \boldsymbol{P}_\pi(\boldsymbol{y} \odot \nabla_{\boldsymbol{\alpha}})$ to satisfy (E_k) for all $k \geq 1$, when the MLP has the form (16).

Figure 1: **Proposed architecture for learning estimators satisfying ($\mathbf{E}_0$) or ($\mathbf{E}_k$) for all $k \geq 1$.**

Design of the b_j: The idea of constructing estimators in which the i-th component is independent of the i-th input variable dates back to Efron (2004). Extending this idea to the case where the i-th component is also independent of the $\pi(i)$-th input variable comes at no additional cost in practice. Indeed, many existing blind-spot neural network implementations already ensure that the i-th output pixel is independent of a subset of the entries of $\boldsymbol{x}$, including x_i. For example, the implementation of Batson & Royer (2019) enforces independence from all pixels whose horizontal and vertical offsets from the i-th pixel are multiples of an integer s (see Figure 3 in Appendix B). We then parameterize the b_j functions using the framework proposed by the latter and define the permutation π as $\pi(i) = i + s$, where the addition is taken modulo the horizontal dimension. Please note that, although this permutation is not cyclic, the resulting vector field f in (5) can nevertheless be shown to remain strictly divergence-free (the cyclic property is only required for the forward implication; see the proof in Appendix E). Finally, for the sake of computational efficiency, we propose to compute the b_j functions all at once via a single π-blind-spot neural network that outputs m different channels, for which the j-th output channel is interpreted as $b_j(\boldsymbol{x})$.

Design of h: Equation (10) places a strong emphasis on the computational efficiency of function h, as it must be evaluated n times in parallel. We propose to implement it as a shallow network and opt for the most straightforward choice, namely a multi-layer perceptron (MLP) (Rosenblatt, 1958).

5 Application to Self-Supervised Denoising

5.1 Gaussian noise

We begin by considering Gaussian denoising problems governed by the model

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}, \sigma^2 \boldsymbol{I})\,, \tag{11}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ denotes the noisy observation, $\boldsymbol{x} \in \mathbb{R}^n$ the underlying clean signal, assumed to be drawn from a distribution $p_{\boldsymbol{x}}$, while the noise level $\sigma > 0$ is itself sampled from a distribution p_σ.

Remarkably, Stein's Unbiased Risk Estimate (SURE) (Stein, 1981) makes it possible to train a denoiser $f : \mathbb{R}^n \to \mathbb{R}^n$ without access to ground-truth signals through the following identity, provided that σ is known for each training sample,

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

where the constant is independent of f. Therefore, when the divergence term can be approximated (Ramani et al., 2008), (SURE) can serve as a self-supervised loss, with almost no performance gap with supervised learning in practice (Soltanayev & Chun, 2018; Metzler et al., 2018).

In this work, we instead focus on the more realistic setting where the noise level σ is unknown, which is substantially more challenging. Among the approaches proposed to address this setting, constraint-based methods have emerged as one of the most successful paradigms (Tachella et al., 2025a; Batson & Royer, 2019; Krull et al., 2019). These methods enforce a structural constraint on the estimator f so that the σ-dependent term in (SURE) vanishes, reducing the learning objective to

the minimization of the measurement consistency $\mathbb{E}_{\boldsymbol{y}}\|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2$. In particular, Herbreteau & Meunier (2026) showed that divergence-free estimators achieve the best trade-off between expressivity and robustness among constraint-based methods under this setting.

5.2 Poisson noise

In many applications, noise arises from the quantum nature of light, as sensors indirectly count discrete photons over a given time interval (Hasinoff, 2014). This leads to a Poisson noise model, which is commonly formulated as (Le Montagner et al., 2014)

$$\boldsymbol{y} \sim \gamma \mathcal{P}(\boldsymbol{x}/\gamma)\,, \tag{12}$$

where $\gamma > 0$ is the noise level drawn from a distribution p_γ. An analogue to SURE, often referred to as PURE (see Appendix C), is given by (Luisier et al., 2010; Le Montagner et al., 2014)

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 = \mathbb{E}_{\boldsymbol{y},\gamma}\left[\|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 - 2\sum_{k=1}^{+\infty}\gamma^k \frac{(-1)^k}{k!}\sum_{i=1}^{n} y_i \partial_i^k f_i(\boldsymbol{y})\right] + \text{const}\,. \tag{PURE}$$

We propose to extend the CENSURE approach (Herbreteau & Meunier, 2026) to the Poisson noise setting by minimizing (PURE) over smooth functions $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ subject to the constraints

$$\forall \boldsymbol{y} \in (0, +\infty)^n, \quad \sum_{i=1}^{n} y_i \partial_i^k f_i(\boldsymbol{y}) = 0\,, \tag{E$_k$}$$

for all $k \geq 1$, thereby eliminating the γ-dependent terms[2] from the objective (PURE).

We first focus on solving (E$_1$). In fact, (E$_1$) can be interpreted as a multidimensional Cauchy–Euler differential equation (Zill, 1997). It is thus not surprising that solving (E$_1$) reduces to solving (E$_0$) through a change of variables involving the logarithm, as established in the following proposition.

Proposition 1. *$f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ satisfies (E$_1$) if and only if there exists $\phi \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ satisfying (E$_0$) such that $\forall \boldsymbol{x} \in (0, \infty)^n$,*

$$f(\boldsymbol{x}) = (\phi \circ \log)(\boldsymbol{x})\,, \tag{13}$$

where log *applies element-wise.*

Corollary 1. *Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ and π be a cyclic permutation of $\{1, \dots, n\}$, with $n \geq 2$. f satisfies (E$_1$) if and only if there exists $g \in C^\infty((0, +\infty)^n, \mathbb{R}^n)$ such that $\forall \boldsymbol{x} \in (0, \infty)^n$,*

$$f(\boldsymbol{x}) = (\boldsymbol{P}_\pi^{-1}\boldsymbol{x}) \odot \operatorname{diag}(\boldsymbol{J}_g(\boldsymbol{x})\boldsymbol{P}_\pi) - \boldsymbol{P}_\pi(\boldsymbol{x} \odot \operatorname{diag}(\boldsymbol{J}_g(\boldsymbol{x})))\,, \tag{14}$$

where $\boldsymbol{P}_\pi$ and $\odot$ denote the permutation matrix associated to π and the Hadamard product, respectively. In other words, $\forall i \in \{1, \dots, n\}, \forall \boldsymbol{x} \in (0, \infty)^n$,

$$f_i(\boldsymbol{x}) = x_{\pi(i)}\partial_{\pi(i)} g_i(\boldsymbol{x}) - x_{\pi^{-1}(i)}\partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)}(\boldsymbol{x})\,. \tag{15}$$

Solving (E$_k$) simultaneously for all $k \geq 1$ is much more challenging, and it remains unclear whether the solutions admit a simple characterization. Nevertheless, observing that any function f satisfying (E$_k$) must also satisfy (E$_1$), we can identify a subset of the solutions, as established in the following proposition.

Proposition 2. *$f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ satisfies (E$_k$) for all $k \geq 1$ if it is of the form (14) with the additional constraint on $g \in C^\infty((0, +\infty)^n, \mathbb{R}^n)$ that $\forall i \in \{1, \dots, n\}, \exists h_1, h_2, h_3 \in C^\infty((0, \infty)^{n-1}, \mathbb{R}), \forall \boldsymbol{x} \in (0, \infty)^n$,*

$$g_i(\boldsymbol{x}) = h_1(x_i + x_{\pi(i)}, \boldsymbol{x}_{-i,-\pi(i)}) + h_2(x_i, \boldsymbol{x}_{-i,-\pi(i)}) + h_3(x_{\pi(i)}, \boldsymbol{x}_{-i,-\pi(i)})\,, \tag{16}$$

where $\boldsymbol{x}_{-i,-\pi(i)}$ denotes the vector $\boldsymbol{x}$ with the i-th and $\pi(i)$-th components omitted.

The close similarity between the representations (5) and (14) naturally motivates a neural network parameterization closely related to that proposed for divergence-free functions. More specifically, we modify only the output layer of the architecture introduced in Section 4, as shown in Figure 1. Note that no specific constraint on the channel-wise MLP is required for the whole network to satisfy (E$_1$). However, satisfying (E$_k$) simultaneously for all $k \geq 1$ requires the MLP to admit a representation of the form (16). This can be achieved, for example, by defining three subnetworks to approximate h_1, h_2, and h_3, respectively, and summing their outputs.

[2]Since f is smooth, the mapping $\boldsymbol{y} \mapsto \sum_{i=1}^n y_i \partial_i^k f_i(\boldsymbol{y})$ is continuous on $\mathbb{R}^n$. As $(0, +\infty)^n$ is dense in $[0, +\infty)^n$, it follows that (E$_k$) automatically extends to the closed orthant $[0, +\infty)^n$, where the data $\boldsymbol{y}$ lie.

5.3 Self-Calibrated Signal-Reinjection for Improved Denoising

Regardless of the noise type, learning is performed by minimizing the measurement consistency under constraint

$$f_0 = \arg\min_f \mathbb{E}_{\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 \quad \text{s.t.} \quad f \in \mathcal{S}, \tag{17}$$

where $\mathcal{S}$ denotes the appropriate constraint set, defined by the satisfaction of either (E_0) or (E_k) for all $k \geq 1$, depending on the noise model. However, enforcing the constraint may exclude certain estimators that, although not admissible during training, can provide improved denoising performance at inference time. This is the case for estimators of the form

$$f_c = c\,\mathrm{id} + (1-c) f_0, \tag{18}$$

where $c \in \mathbb{R}\backslash\{0\}$ and id denotes the identity function. For instance, under the Gaussian noise model, (18) corresponds to divergence-constant estimators[3], which have been shown to outperform their divergence-free counterparts (Herbreteau & Meunier, 2026). However, the theoretically optimal value c^* in (18) for denoising depends on the noise level, which is assumed to be unknown. In what follows, we propose a signal-reinjection strategy for estimating the optimal c^* at inference time. Our approach is applicable regardless of the noise type and does not rely on the knowledge of the noise level.

Following the intuition of Kim et al. (2022), we expect two nearby noisy observations, $\boldsymbol{y}$ and $\boldsymbol{y}'$, to yield similar denoised estimates. In particular, for the optimal interpolation parameter c^*, we expect $f_{c^*}(\boldsymbol{y}) \approx f_{c^*}(\boldsymbol{y}')$. Motivated by this observation, we propose estimating c^* by minimizing the gap between the corresponding denoised outputs:

$$\hat{c} = \arg\min_{c \in \mathbb{R}} \|f_c(\boldsymbol{y}) - f_c(\boldsymbol{y}')\|_1. \tag{19}$$

This optimization problem admits the closed-form solution

$$\hat{c} = \mathrm{wmed}\left(\frac{f_0(\boldsymbol{y}') - f_0(\boldsymbol{y})}{f_0(\boldsymbol{y}') - f_0(\boldsymbol{y}) - (\boldsymbol{y}' - \boldsymbol{y})}, \frac{|f_0(\boldsymbol{y}') - f_0(\boldsymbol{y}) - (\boldsymbol{y}' - \boldsymbol{y})|}{\mathbf{1}^\top |f_0(\boldsymbol{y}') - f_0(\boldsymbol{y}) - (\boldsymbol{y}' - \boldsymbol{y})|} \right), \tag{20}$$

where wmed denotes the weighted median (Edgeworth, 1888), and the division is performed element-wise. Notably, computing $\hat{\boldsymbol{x}} = f_{\hat{c}}(\boldsymbol{y})$ requires only two inference passes, namely $f_0(\boldsymbol{y})$ and $f_0(\boldsymbol{y}')$, where f_0 is learned by solving (17) and parameterized by the proposed architecture (see Fig. 1). In practice, from a single noisy observation $\boldsymbol{y}$, we generate an artificially nearby observation $\boldsymbol{y}' \sim \mathcal{N}(\boldsymbol{y}, \tau^2 \boldsymbol{I})$ with τ a small constant (typically $\tau = 10^{-5}$).

6 Experimental Results

We evaluate the effectiveness of the proposed methodology for constructing neural networks that satisfy the partial differential equations (E_0) or (E_k) for all $k \geq 1$ simultaneously, in the context of self-supervised image denoising, where ground-truth data are not available during training. We consider three distinct noise regimes: Gaussian, Poisson, and real-world noise. We assess the competitiveness of our approach against several representative self-supervised methods. These include the constraint-based approaches UNSURE (Tachella et al., 2025a), CENSURE (Herbreteau & Meunier, 2026) and Noise2Self (Batson & Royer, 2019), the latter being a representative blind-spot network. We further consider two state-of-the-art approaches, Neighbor2Neighbor (Huang et al., 2021) and the noise-level-agnostic extension of Noise2Score (Kim et al., 2022). To ensure a fair and consistent comparison, all methods are implemented using the same backbone architecture and trained from scratch by us, with the aid of the DeepInverse library (Tachella et al., 2025b). As additional baselines, we consider supervised learning (minimization of the mean squared error), and BM3D (Dabov et al., 2007), a classical denoising method provided with the oracle noise level and an appropriate variance-stabilizing transformation (VST) (Makitalo & Foi, 2012) when necessary. Performance is assessed in terms of peak signal-to-noise ratio (PSNR). Further details on the datasets, training protocols, and implementation choices are provided in Appendix D. The source code is available in the Supplementary Material.

[3]Indeed, $f_c = \arg\min_{f \in \mathcal{S}_c} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2$ with $\mathcal{S}_c = \{f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n) \,|\, \forall \boldsymbol{y} \in \mathbb{R}^n, \mathrm{div}\, f(\boldsymbol{y}) = nc\}$.

Table 1: Average PSNR (dB) results of different self-supervised denoising methods on grayscale and color datasets. The best and second best results are highlighted in red and blue, respectively.

	Noise level	Dataset	Baselines		Constraint-based methods					Others	
			SL	BM3D	B-Spot	UNS	CENS	**Ours**	**Ours†**	Nb2Nb	N2Sc
Gaussian	15 / 255	Set12	33.13	32.37	30.98	25.33	31.62	31.87	32.40	**32.48**	**32.73**
	25 / 255		30.84	29.97	29.38	20.56	29.80	29.95	**30.28**	**30.40**	30.27
	50 / 255		27.76	26.72	26.82	14.59	27.05	27.13	**27.33**	**27.40**	26.49
	15 / 255	BSD68	31.84	31.08	29.20	25.23	29.73	30.03	30.82	**31.30**	**31.47**
	25 / 255		29.41	28.57	27.74	20.56	28.13	28.28	28.73	**29.08**	**28.98**
	50 / 255		26.49	25.60	25.63	14.70	25.81	25.89	**26.11**	**26.32**	25.58
Poisson	3 / 255	Set12	32.18	31.33	30.41	24.20	27.53	30.93	**31.32**	**31.69**	27.61
	6 / 255		30.62	29.70	29.25	20.72	24.91	29.66	**29.92**	**30.26**	21.06
	9 / 255		29.69	28.75	28.51	18.81	23.17	28.84	**29.08**	**29.34**	18.18
	3 / 255	BSD68	31.02	30.26	28.75	24.87	27.51	29.21	**29.90**	**30.58**	27.80
	6 / 255		29.38	28.56	27.74	21.46	25.33	28.07	**28.55**	**29.12**	22.15
	9 / 255		28.47	27.62	27.11	19.55	23.82	27.37	**27.78**	**28.23**	19.19
Real-world	raw	SIDD Validation	51.86	48.92	50.53	37.77	39.56	50.64	**50.80**	**51.47**	39.15
	raw	SIDD Benchmark	51.23	48.60	49.97	37.32	38.99	50.04	**50.20**	**50.85**	39.05
	raw	DND Benchmark	48.08	47.15	47.34	40.18	37.86	47.41	**47.70**	**47.99**	43.78
	raw	Confocal Mice	38.69	37.93	**37.87**	29.45	30.93	37.31	**37.44**	37.13	29.38
	raw	Confocal BPAE-G	34.43	34.07	**33.69**	28.68	30.08	33.44	33.62	**34.36**	28.61
	raw	Two-Photon Mice	34.46	33.81	**33.67**	25.03	28.94	33.32	33.44	**33.72**	24.94
	raw	Two-Photon BPAE-G	32.04	31.59	31.46	25.13	28.15	31.38	**31.77**	**31.08**	25.05

Abbreviations: SL: supervised learning, B-Spot: Blind-spot (here, Noise2Self), UNS: UNSURE, CENS: CENSURE, Nb2Nb: Neighbor2Neighbor, N2Sc: Noise2Score.

6.1 Synthetic Noise

We first evaluate our approach under artificially generated noise, following the Gaussian and Poisson noise models defined in (11) and (12), respectively, with varying noise levels. For each noise regime, we train a single model for handling all noise levels within a broad range. Importantly, the noise levels are assumed to be unknown and their values are never provided to the model, either through the loss functions or at inference time. Table 1 reports the resulting PSNR values on standard benchmark datasets, while representative qualitative comparisons are presented in Figure 2. For Gaussian noise, our approach consistently outperforms the competing constraint-based methods, including CENSURE. This result provides empirical evidence that our parameterization of divergence-free fields, derived from Theorem 2, offers a more effective representation than the parameterization adopted by CENSURE, which relies on a less compact construction (Richter-Powell et al., 2022). Notably, our approach also achieves this improvement at a substantially lower computational cost (see Appendix B). The advantage of our method persists under Poisson noise, where the performance of CENSURE deteriorates markedly. This degradation can be attributed to a mismatch between the constraint in (E_0) and the unbiased loss estimate in (PURE), which ultimately leads to overfitting of the noise, as illustrated in Figure 2b. Finally, incorporating the signal-reinjection strategy introduced in Section 5.3 and indicated with the symbol †, yields a substantial further improvement in performance. This enhancement significantly narrows the gap with Neighbor2Neighbor, which achieves the best overall performance among the self-supervised methods tested in our experiments.

6.2 Real-World Noise

We further evaluate our approach on several real-world denoising benchmarks, including SIDD (Abdelhamed et al., 2018) and DND (Plotz & Roth, 2017) for raw-RGB image denoising, as well as FMDD (Zhang et al., 2019) for fluorescence microscopy. In all experiments, we employ the parameterization originally developed for the Poisson noise model, namely, the one satisfying (E_k) for all $k \geq 1$. This allows us to investigate to what extent a parameterization motivated by the Poisson noise model can generalize to real-world acquisition noise. For raw-RGB images, this assumption appears to be reasonably effective. In particular, our approach slightly improves upon the

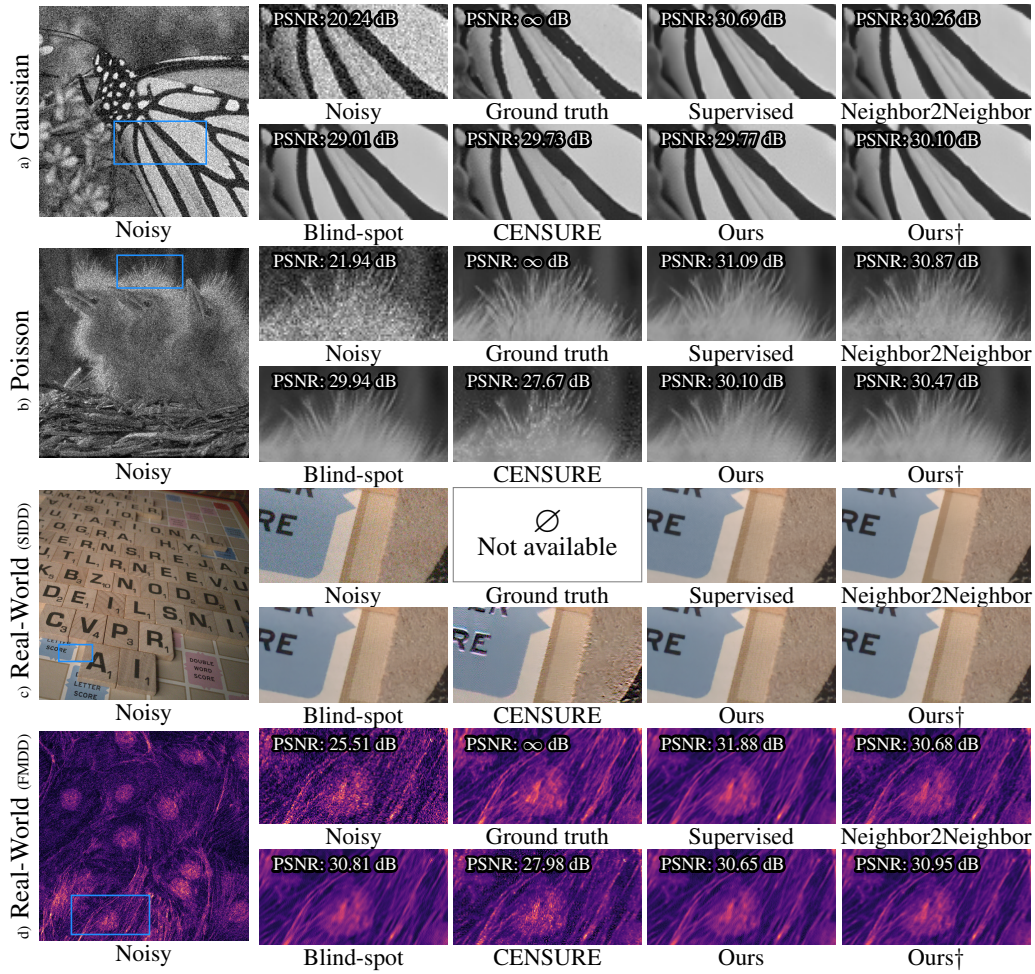


Figure 2: **Qualitative image denoising results.** Best viewed by zooming and in color.

performance of the blind-spot methods and, when combined with the signal-reinjection technique, achieves the second-best overall performance (see Table 1). The results are, however, more nuanced for fluorescence microscopy, where the additional flexibility offered by our approach does not consistently translate into improved performance. This suggests that real-world noise may not always be adequately described by a Poisson model. Additional sources of variability introduced during image acquisition, such as under- or over-exposure (Foi et al., 2008), may indeed lead to deviations from the Poisson assumptions, thereby limiting the benefits of our parameterization in certain settings. Nevertheless, our approach remains robust and can occasionally outperform Neighbor2Neighbor, especially in the low-noise regime (see Figure 2c), where Neighbor2Neighbor tends to oversmooth fine details, whereas our approach better preserves the underlying image structure and texture.

7 Conclusion

We introduced a novel neural network architecture, constrained to have zero divergence by design. Our parameterization is grounded in a compact representation theorem that generalizes the classical curl representation to arbitrary dimensions, enabling both expressive and computationally efficient estimators. With application to self-supervised image denoising, we further extended the CENSURE framework from Gaussian to Poisson noise, providing a natural pathway toward real-world imaging scenarios, and introduced an original signal-reinjection technique that boosts performance at inference time. Extensive experiments demonstrate that our approach outperforms existing constraint-based methods while narrowing the gap with state-of-the-art self-supervised denoisers.

AI USE STATEMENT

In this work, we used generative AI tools for aiding and polishing writing. We have not used generative AI tools for any other tasks requiring disclosure. We have reviewed all AI-assisted text and verified that it accurately reflects the work and our intended meaning. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1692–1700, 2018.

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

Cristian Barbarosie. Representation of divergence-free vector fields. Quarterly of applied mathematics, 69(2):309–316, 2011.

George Keith Batchelor. An introduction to fluid dynamics. Cambridge University Press, 2000.

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

Simon Benaïchouche, Guillaume Morel, François Rousseau, and Ronan Fablet. Divergence-free continuous normalizing flows for uncertainty quantification. HAL preprint hal-03813499, 2022.

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

A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation, 4(2):490–530, 2005.

Harold C. Burger, Christian J. Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with BM3D? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2392–2399, 2012.

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

Sungmin Cha and Taesup Moon. Neural adaptive image denoiser. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2981–2985. IEEE, 2018.

Sungmin Cha and Taesup Moon. Fully convolutional pixel adaptive image denoiser. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4160–4169, 2019.

George Cybenko. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems, 2(4):303–314, 1989.

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.

Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In Advances in Neural Information Processing Systems (NeurIPS), volume 13, 2000.

Francis Ysidro Edgeworth. XXII. On a new method of reducing observations relating to several quantities. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 25(154):184–191, 1888.

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.

David J Griffiths. Introduction to electrodynamics. Cambridge University Press, 2023.

Samuel W Hasinoff. Photon, Poisson noise. Computer Vision, A Reference Guide, 4(16):1, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415, 2016.

Sébastien Herbreteau and Etienne Meunier. Divergence-free neural networks with application to image denoising. In International Conference on Learning Representations (ICLR), pp. 145778–145802, 2026.

Sébastien Herbreteau and Michael Unser. Self-calibrated variance-stabilizing transformations for real-world image denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10496–10506, 2025.

Sébastien Herbreteau and Charles Kervrann. DCT2net: an interpretable shallow CNN for image denoising. IEEE Transactions on Image Processing, 31:4292–4305, 2022.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural Networks, 2(5):359–366, 1989.

Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2Neighbor: Self-supervised denoising from single noisy images. In Proceedings of the IEEE/CVF 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), 2022.

Hyemi Jang, Junsung Park, Dahuin Jung, Jaihyun Lew, Ho Bae, and Sungroh Yoon. PUCA: Patch-unshuffle and channel attention for enhanced self-supervised image denoising. Advances in Neural Information Processing Systems (NeurIPS), 36:19217–19229, 2023.

Zakaria Jarraya, Lucas Drumetz, Simon Benaïchouche, Douraied Ben Salem, and François Rousseau. On divergence-free neural ODE for classification. In International Conference on Pattern Recognition, pp. 254–265. Springer, 2024.

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.

Kwanyoung Kim, Taesung Kwon, and Jong Chul Ye. Noise distribution adaptive self-supervised image denoising using tweedie distribution and score matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2008–2016, 2022.

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 Proceedings of the IEEE/CVF 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.

Yoann Le Montagner, Elsa D Angelini, and Jean-Christophe Olivo-Marin. An unbiased risk estimator for image denoising in the presence of mixed Poisson–Gaussian noise. IEEE Transactions on Image Processing, 23(3):1255–1268, 2014.

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 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1132–1140, 2017.

Florian Luisier, Thierry Blu, and Michael Unser. Image denoising in mixed Poisson–Gaussian noise. IEEE Transactions on Image Processing, 20(3):696–708, 2010.

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.

Markku Makitalo and Alessandro Foi. Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise. IEEE Transactions on Image Processing, 22(1):91–103, 2012.

Youssef Mansour and Reinhard Heckel. Zero-Shot Noise2Noise: Efficient image denoising without any data. In Proceedings of the IEEE/CVF 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 Proceedings of the IEEE International Conference on Computer Vision (ICCV), volume 2, pp. 416–423 vol.2, 2001.

Christopher A Metzler, Ali Mousavi, Reinhard Heckel, and Richard G Baraniuk. Unsupervised learning with Stein's Unbiased Risk Estimator. arXiv preprint arXiv:1805.10531, 2018.

Sreyas Mohan, Zahra Kadkhodaie, Eero P Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. International Conference on Learning Representations (ICLR), 2020.

Brayan Monroy, Jorge Bacca, and Julián Tachella. Generalized recorrupted-to-recorrupted: Self-supervised learning beyond gaussian noise. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 28155–28164, 2025.

Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2Noise: Learning to denoise from unpaired noisy data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12064–12072, 2020.

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 Proceedings of the IEEE/CVF 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.

Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1586–1595, 2017.

Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1890–1898, 2020.

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. In Advances in Neural Information Processing Systems (NeurIPS), volume 35, pp. 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.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review, 65(6):386, 1958.

Walter Rudin. Principles of Mathematical Analysis. International series in pure and applied mathematics. McGraw-Hill, 1976. ISBN 9780070856134.

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 and Mike Davies. Self-supervised learning from noisy and incomplete data. Foundations and Trends® in Signal Processing, 20(2):85–184, 2026.

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 Representations (ICLR), pp. 84908–84928, 2025a.

Julián Tachella, Matthieu Terris, Samuel Hurault, Andrew Wang, Leo Davy, Jérémy Scanvic, Victor Sechaud, Romain Vo, Thomas Moreau, Thomas Davies, Dongdong Chen, Nils Laurent, Brayan Monroy, Jonathan Dong, Zhiyuan Hu, Minh-Hai Nguyen, Florian Sarron, Pierre Weiss, Paul Escande, Mathurin Massias, Thibaut Modrzyk, Brett Levac, Tobías I. Liaudat, Maxime Song, Johannes Hertrich, Sebastian Neumayer, and Georg Schramm. Deepinverse: A Python package for solving imaging inverse problems with deep learning. Journal of Open Source Software, 10 (115):8923, 2025b.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9446–9454, 2018.

Zejin Wang, Jiazheng Liu, Guoqing Li, and Hua Han. Blind2Unblind: Self-supervised image denoising with visible blind spots. In Proceedings of the IEEE/CVF 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. In Advances in Neural Information Processing Systems (NeurIPS), volume 36, pp. 69752–69763, 2023.

Guoshen Yu and Guillermo Sapiro. DCT image denoising: a simple and effective image denoising algorithm. Image Processing On Line, 1:292–296, 2011.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5728–5739, 2022.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Transactions on Image Processing, 26 (7):3142–3155, 2017.

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.

Yide Zhang, Yinhao Zhu, Evan Nichols, Qingfei Wang, Siyuan Zhang, Cody Smith, and Scott Howard. A Poisson-Gaussian denoising dataset with real fluorescence microscopy images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11710–11718, 2019.

D.G. Zill. A First Course in Differential Equations with Modeling Applications. Mathematics Series. Brooks/Cole, 1997. ISBN 9780534955748.

Yunhao Zou, Chenggang Yan, and Ying Fu. Iterative denoiser and noise estimator for self-supervised image denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 13265–13274, 2023.

A LIMITATIONS

Our self-calibrated signal-reinjection procedure requires an additional inference pass, effectively doubling the inference time compared to a single-pass configuration. This additional computational cost should be considered when applying the method in computationally constrained or real-time settings.

The training and inference costs of our approach are closely tied to the choice of π-blind-spot architecture. In this work, we employ Noise2Self (Batson & Royer, 2019) due to its simplicity and low training cost (at the expense of a higher inference time). Alternative π-blind-spot implementations, such as PUCA (Jang et al., 2023), may offer improved inference efficiency and could therefore reduce the computational overhead of our approach.

Finally, our parameterization is constrained by Assumption 1, which may limit its expressiveness. While this assumption is useful for the proposed formulation, relaxing it or developing a more expressive parameterization could potentially improve the flexibility and performance of the method.

B ADDITIONAL RESULTS

Table 2: Execution time (in seconds) comparison on a batch of size $4 \times 1 \times 128 \times 128$ for different approaches with shared backbone architecture (GPU: Tesla V100).

Model	Training ↓	Inference ↓	# parameters
Supervised	0.026	0.008	17, 007, 744
Neighbor2Neighbor	0.020	0.008	17, 007, 744
Noise2Score	0.026	0.016	17, 007, 744
UNSURE	0.054	0.008	17, 007, 744
Noise2Self (blind-spot)	0.026	0.072	17, 007, 744
CENSURE	1.323	0.073	17, 008, 256
Ours	0.031	0.076	17, 078, 496
Ours†	0.031	0.152	17, 078, 496

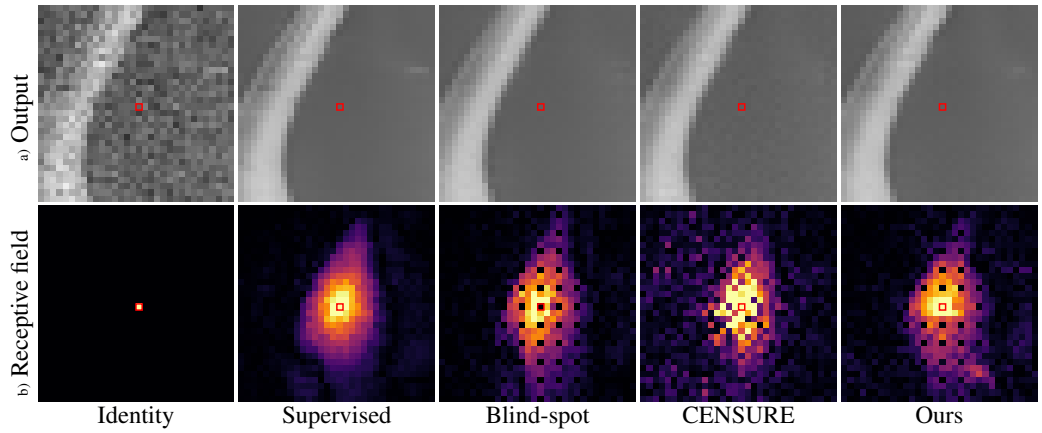


Figure 3: **Visualization of the receptive fields induced by different denoising methods.** We quantify the influence of each input pixel on the central output pixel using input-output gradients, with brighter colors indicating stronger influence. Unlike blind-spot methods, our approach can directly leverage the central pixel, which often provides highly informative signal for denoising, resulting in a more expressive estimator. Best viewed by zooming and in color.

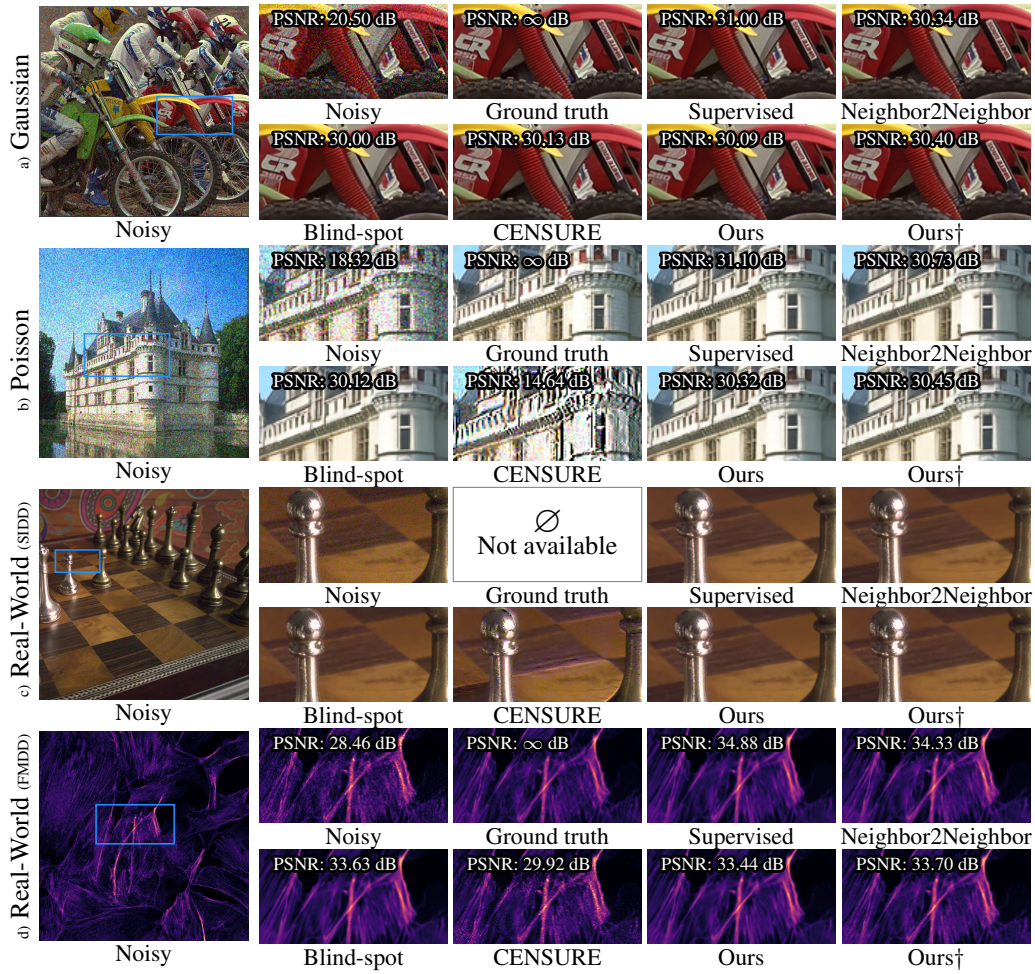


Figure 4: **Additional image denoising results.** Best viewed by zooming and in color.

B.1 On the loss function for the self-calibrated signal-reinjection technique

In (19), we chose the ℓ_1 loss to measure the discrepancy between $f_c(\boldsymbol{y})$ and $f_c(\boldsymbol{y}')$, which leads to a closed-form solution for the optimal $\hat{c}$ in the form of a weighted median (Edgeworth, 1888). This choice is not unique, however, and two natural alternatives can be considered:

1. Replacing the weighted median with a plain median:

$$c_{\text{med}} = \text{median}\left(\frac{f_0(\boldsymbol{y}') - f_0(\boldsymbol{y})}{f_0(\boldsymbol{y}') - f_0(\boldsymbol{y}) - (\boldsymbol{y}' - \boldsymbol{y})}\right), \tag{21}$$

2. Replacing the ℓ_1 loss with the ℓ_2 loss, which also admits a closed-form solution, given by

$$c_{\ell_2} = \arg\min_{c\in\mathbb{R}} \|f_c(\boldsymbol{y}) - f_c(\boldsymbol{y}')\|_2 \tag{22}$$

$$= \frac{\langle f_0(\boldsymbol{y}') - f_0(\boldsymbol{y}), f_0(\boldsymbol{y}') - f_0(\boldsymbol{y}) - (\boldsymbol{y}' - \boldsymbol{y})\rangle}{\|f_0(\boldsymbol{y}') - f_0(\boldsymbol{y}) - (\boldsymbol{y}' - \boldsymbol{y})\|_2^2}. \tag{23}$$

Table 4 compares the performance of the different choices for the constant c in (18). Interestingly, using the median instead of the weighted median appears to provide greater robustness, particularly in real-world settings. Conversely, the ℓ_2 loss achieves the best performance on synthetic noise, but is more susceptible to failure as the noise becomes more complex. Overall, the ℓ_1 formulation appears to strike a good balance between robustness and performance, making it a suitable choice across the diverse noise regimes considered in our experiments.

Table 3: Average PSNR (dB) results of different self-supervised denoising methods on color datasets with synthetic noise. The best and second best results are highlighted in red and blue, respectively.

	Noise level	Dataset	Baselines		Constraint-based methods					Others	
			SL	BM3D	B-Spot	UNS	CENS	**Ours**	**Ours†**	Nb2Nb	N2Sc
Gaussian	15 / 255	Kodak24	35.18	34.28	34.08	26.89	34.21	34.25	34.53	**34.72**	**34.69**
	25 / 255		32.78	32.15	31.90	22.01	32.05	32.05	**32.22**	**32.37**	32.02
	50 / 255		29.77	28.46	29.07	15.93	29.24	29.19	**29.27**	**29.42**	28.17
	75 / 255		28.14	26.82	27.49	12.82	**27.67**	27.60	27.65	**27.82**	22.91
	15 / 255	CBSD68	34.22	33.52	33.06	26.83	33.17	33.20	33.61	**33.82**	**33.82**
	25 / 255		31.61	30.71	30.70	22.10	30.83	30.84	**31.09**	**31.29**	31.00
	50 / 255		28.42	27.38	27.78	16.04	27.91	27.89	**28.00**	**28.16**	27.16
	75 / 255		26.74	25.74	26.21	12.94	26.33	26.30	**26.36**	**26.53**	22.49
Poisson	5 / 255	Kodak24	33.43	32.43	32.47	29.53	28.04	32.59	**32.81**	**32.99**	22.17
	10 / 255		31.83	30.76	30.99	22.19	24.29	31.09	**31.27**	**31.43**	18.35
	15 / 255		30.92	29.80	30.13	19.11	20.84	30.22	**30.39**	**30.54**	16.64
	20 / 255		30.28	29.13	29.53	17.54	18.45	29.59	**29.66**	**29.90**	15.49
	5 / 255	CBSD68	32.28	31.32	31.30	29.58	27.15	31.39	**31.73**	**31.93**	23.12
	10 / 255		30.56	29.57	29.73	22.52	23.61	29.81	**30.07**	**30.26**	18.54
	15 / 255		29.60	28.57	28.84	19.34	20.38	28.91	**28.94**	**29.32**	16.82
	20 / 255		28.94	27.87	28.23	17.67	18.19	**28.27**	27.42	**28.67**	15.68

Table 4: Average PSNR (dB) for different values of the constant c in our proposed signal-reinjection technique, evaluated on grayscale and color datasets. The best and second-best results are highlighted in red and blue, respectively.

	Noise	Dataset	$c = 0$	$c = c_{\text{med}}$	$c = c_{\ell_1}$	$c = c_{\ell_2}$
Gaussian	15 / 255	Set12	31.87	32.37	**32.40**	**32.41**
	25 / 255		29.95	30.27	**30.28**	**30.29**
	50 / 255		27.13	27.32	**27.33**	**27.33**
	15 / 255	BSD68	30.03	30.74	**30.82**	**30.89**
	25 / 255		28.28	28.71	**28.73**	**28.77**
	50 / 255		25.89	26.10	**26.11**	**26.12**
Poisson	3 / 255	Set12	30.93	31.30	**31.32**	**31.43**
	6 / 255		29.66	29.90	**29.92**	**29.96**
	9 / 255		28.84	**29.06**	**29.08**	28.79
	3 / 255	BSD68	29.21	29.84	**29.90**	**30.07**
	6 / 255		28.07	28.51	**28.55**	**28.67**
	9 / 255		27.37	**27.75**	**27.78**	27.72
Real-World	raw	SIDD Validation	50.64	**50.79**	**50.80**	50.47
	raw	SIDD Benchmark	50.04	**50.19**	**50.20**	49.80
	raw	DND Benchmark	47.41	47.69	**47.70**	**47.74**
	raw	Confocal Mice	37.31	**37.60**	**37.44**	36.81
	raw	Confocal BPAE-G	33.44	**34.04**	**33.62**	31.69
	raw	Two-Photon Mice	33.32	**33.43**	**33.44**	33.41
	raw	Two-Photon BPAE-G	31.38	31.76	**31.77**	**31.78**

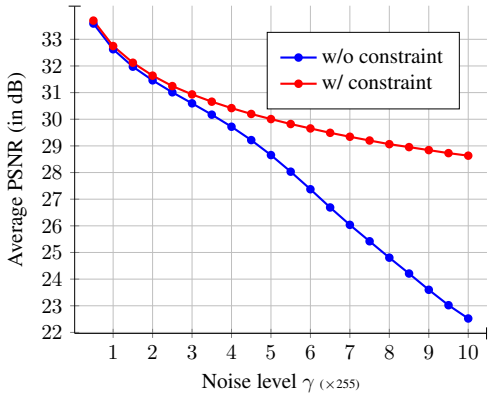


Figure 5: **Performance comparison of our proposed approach on the Set12 test set under synthetic Poisson noise, with and without enforcing the constraint in (16) on the MLP**. In the absence of this constraint, performance progressively degrades as the noise level increases, underscoring the importance of imposing the proposed structural form on the channel-wise MLP.

B.2 On the need to constrain the MLP for Poisson noise

When the MLP is not constrained according to (16), the resulting network is guaranteed to satisfy (E_1), but no such guarantee holds for the higher-order constraints (E_k) with $k \geq 2$. These constraints are nevertheless required by (PURE) to eliminate all γ-dependent terms. One may argue that, in practice, the higher-order terms can be neglected. Indeed, some authors (Le Montagner et al., 2014; Tachella et al., 2025a) rely on the empirical observation that higher-order derivatives are typically much smaller than their first-order counterpart, e.g., $|\partial_i^k f_i(\boldsymbol{y})| \ll |\partial_i f_i(\boldsymbol{y})|$ for $k \geq 2$. This approximation is by the way adopted in UNSURE (Tachella et al., 2025a) for Poisson noise, where enforcing an infinite hierarchy of constraints would be impractical.

It is therefore natural to ask whether the same approximation is sufficient in our setting. Figure 5 shows that this is only the case at sufficiently low noise levels. As γ increases, the PSNR degrades sharply, indicating that the neglected higher-order terms become increasingly significant and that the approximation consequently breaks down. This result highlights the need to explicitly constrain the MLP when considering Poisson noise, rather than relying on the assumption that higher-order derivative terms are negligible.

C Poisson Unbiased Risk Estimate (PURE)

An analogue to SURE for Poisson noise (12), often referred to as PURE, is given by (Luisier et al., 2010; Le Montagner et al., 2014):

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 = \mathbb{E}_{\boldsymbol{y},\gamma}\left[\|f(\boldsymbol{y})\|_2^2 + \|\boldsymbol{y}\|_2^2 - \gamma \mathbf{1}^\top \boldsymbol{y} - 2\sum_{i=1}^{n} y_i f_i(\boldsymbol{y} - \gamma \boldsymbol{e}_i)\right], \tag{24}$$

where $\boldsymbol{e}_i$ denotes the i-th canonical vector.

Assuming that, for fixed $\boldsymbol{y}_{-i}$, the function $y_i \mapsto f_i(\boldsymbol{y})$ is real-analytic with infinite radius of convergence[4], we have

$$f_i(\boldsymbol{y} - \gamma \boldsymbol{e}_i) = \sum_{k=0}^{+\infty} \frac{(-\gamma)^k}{k!} \partial_i^k f_i(\boldsymbol{y}), \tag{25}$$

[4]This is true for a standard CNN with GELU (Hendrycks & Gimpel, 2016) activation functions for example.

hence,

$$\sum_{i=1}^{n} y_i f_i(\boldsymbol{y} - \gamma \boldsymbol{e}_i) = \sum_{i=1}^{n} y_i \sum_{k=0}^{+\infty} \gamma^k \frac{(-1)^k}{k!} \partial_i^k f_i(\boldsymbol{y}) \tag{26}$$

$$= \sum_{i=1}^{n} y_i f_i(\boldsymbol{y}) + \sum_{k=1}^{+\infty} \gamma^k \frac{(-1)^k}{k!} \sum_{i=1}^{n} y_i \partial_i^k f_i(\boldsymbol{y}) \,. \tag{27}$$

Finally,

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|f(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 = \mathbb{E}_{\boldsymbol{y},\gamma} \left[\|f(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 - 2 \sum_{k=1}^{+\infty} \gamma^k \frac{(-1)^k}{k!} \sum_{i=1}^{n} y_i \partial_i^k f_i(\boldsymbol{y}) \right] + \text{const} \,, \tag{28}$$

where the constant is independent of f.

D Implementation details

Common backbone architecture For a fair comparison, we use the U-Net architecture adopted in Herbreteau & Meunier (2026), a lightweight variant of the popular DRUNet architecture (Zhang et al., 2022), as the shared neural network backbone across all approaches. Specifically, the network follows an encoder–decoder architecture with residual connections (He et al., 2016) and no bias terms (Mohan et al., 2020). Each of the four scales consists of two residual blocks of the form "3×3 conv $\rightarrow$ Softplus(β=100) $\rightarrow$ 3×3 conv". Spatial downsampling is performed using 2×2 convolutions with stride 2, while upsampling is performed using 2×2 transposed convolutions with the same stride. The channel widths at the four scales are 64, 128, 256, and 512, respectively. The resulting network contains approximately 17M trainable parameters.

Datasets for synthetic noise The models are trained on the large-scale dataset used in Zhang et al. (2022), comprising 8,694 images in total: 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 2,650 images from Flickr2K (Lim et al., 2017). During training, synthetic noisy images are generated on the fly by sampling from the noise models in (11) and (12) for the Gaussian and Poisson settings, respectively. For grayscale images, the noise parameters are sampled as $\sigma \sim \mathcal{U}([0, 50/255])$ and $\gamma \sim \mathcal{U}([0.1/255, 10/255])$. For color images, we use $\sigma \sim \mathcal{U}([0, 75/255])$ and $\gamma \sim \mathcal{U}([0.1/255, 20/255])$. For evaluation, we consider two widely used benchmark datasets for each image type. Specifically, we use Set12 and BSD68 (Martin et al., 2001) for grayscale images, and Kodak24 and CBSD68 (Martin et al., 2001) for color images.

Datasets for real-world noise For image denoising in the raw-RGB space, we use the SIDD Medium dataset (Abdelhamed et al., 2018) for training. The dataset comprises images captured by five smartphone cameras across 10 scenes under a range of lighting conditions. In total, 320 noisy images are used, corresponding to two noisy images per scene instance. For evaluation, we consider both the SIDD validation and benchmark datasets. For the benchmark test set, the ground-truth images are not publicly available to avoid any bias in the evaluation (the quality scores are assessed online). Following Zamir et al. (2022), we also evaluate our models directly on the Darmstadt Noise Dataset (DND) (Plotz & Roth, 2017) without additional training, as it exhibits characteristics similar to those of SIDD. Note that, following Huang et al. (2021), each single-channel image is split into four sub-images according to the Bayer pattern and subsequently stacked to form a four-channel image, which is then provided as input to the deep learning-based methods. For fluorescence microscopy, we consider the FMDD dataset (Zhang et al., 2019) and retain only the raw-image sets acquired using confocal and two-photon microscopy, resulting in nine sets in total. Each set contains 20 views, with 50 distinct noise realizations per view. For each view, the ground truth is computed by averaging the 50 noisy occurrences. Following the methodology of Huang et al. (2021), we exclude the 19th view of each set from training and reserve it exclusively for evaluation. In total, all models are trained on 8,550 noisy images.

Training details We adopt the same training protocol as Herbreteau & Meunier (2026). Specifically, all models are trained for 600,000 iterations, with each iteration consisting of a gradient-based update computed from a batch of randomly cropped 128×128 patches. For CENSURE, we instead use 64×64 patches to accelerate training, following the recommendation of Herbreteau & Meunier (2026). We use a batch size of 16 and the Adam optimizer (Kingma & Ba, 2015), with an initial learning rate of 10^{-4} that is halved every 150,000 iterations. The training data are augmented using random horizontal and vertical flips, as well as random rotations by $90°$. Unless otherwise specified by the original papers, we use the default settings provided by the DeepInverse library (Tachella et al., 2025b) for each approach.

Implementation choices specific to our method Our proposed architecture (see Fig. 1) involves two additional hyperparameters: the offset s, which defines the permutation π, and the number of output channels m of the π-blind-spot network. Throughout all experiments, we set $m = 64$, while $s = 3$ for grayscale images and $s = 2$ for all other settings (the exact same parameters were used for the method Noise2Self). For the shallow channel-wise MLP, we adopt a five-layer architecture, with 96 units in each hidden layer, and use Softplus activations (Dugas et al., 2000) with a sharpness parameter of $\beta = 100$. Note that replacing Softplus with the GELU activation function (Hendrycks & Gimpel, 2016), in order to satisfy the requirement that the mapping $y_i \mapsto f_i(\boldsymbol{y})$ be real-analytic with an infinite radius of convergence (see Section C), did not yield any improvement in performance. In practice, the channel-wise MLP is implemented using 1×1 convolutions. Finally, in the Poisson regime, where the neural network is constrained to satisfy (E_k) for all $k \geq 1$, we sample $\varepsilon \sim \mathcal{U}((0, 0.01])$ and add it pixel-wise to the noisy images during training to avoid zero-valued pixels, which could otherwise hinder the learning process. At test time, we set $\varepsilon = 0.005$ and to avoid introducing an intensity bias in the PSNR evaluation, we subtract ε from the final output, *i.e.*, we use $f(\boldsymbol{y} + \varepsilon \mathbf{1}) - \varepsilon \mathbf{1}$.

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. The source code is available in the Supplementary Material.

E PROOFS

Lemma 1. *Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$, with $n \geq 2$. If f is divergence-free then there exists $g \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ such that for all $\boldsymbol{x} \in \mathbb{R}^n$,*

$$f(\boldsymbol{x}) = \boldsymbol{J}_g(\boldsymbol{x})\boldsymbol{e}_1 - \operatorname{div} g(\boldsymbol{x})\boldsymbol{e}_1 \,, \tag{29}$$

where $\boldsymbol{e}_1$ denotes the first canonical basis vector of $\mathbb{R}^n$.

Proof. Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ be a smooth divergence-free vector field. Define, for all $i \in \{2, \dots, n\}$,

$$g_i : \boldsymbol{x} \in \mathbb{R}^n \mapsto \int_0^{x_1} f_i(\xi, x_2, x_3, \dots, x_n) d\xi - \delta_{i2} \int_0^{x_2} f_1(0, \xi, x_3, \dots, x_n) d\xi \,, \tag{30}$$

where δ_{ij} denotes the Kronecker delta. By construction, $f_i = \partial_1 g_i$ for all $i \in \{2, \dots, n\}$.

Moreover, for all $\boldsymbol{x} \in \mathbb{R}^n$,

$$\begin{aligned}
\sum_{i=2}^{n} \partial_i g_i(\boldsymbol{x}) &= -\partial_2 \int_0^{x_2} f_1(0, \xi, x_3, \dots, x_n) d\xi + \sum_{i=2}^{n} \partial_i \int_0^{x_1} f_i(\xi, x_2, x_3, \dots, x_n) d\xi && (31)\\
&= -f_1(0, x_2, \dots, x_n) + \int_0^{x_1} \sum_{i=2}^{n} \partial_i f_i(\xi, x_2, x_3, \dots, x_n) d\xi && (32)\\
&= -f_1(0, x_2, \dots, x_n) - \int_0^{x_1} \partial_1 f_1(\xi, x_2, x_3, \dots, x_n) d\xi && (33)\\
&= -f_1(0, x_2, \dots, x_n) - f_1(\boldsymbol{x}) + f_1(0, x_2, \dots, x_n) && (34)\\
&= -f_1(\boldsymbol{x}) \,, && (35)
\end{aligned}$$

where, in (33), we used the fact that f is divergence-free.

Define $g_1 : \mathbb{R}^n \to \mathbb{R}$ to be the zero function. Finally, for all $\boldsymbol{x} \in \mathbb{R}^n$,

$$f(\boldsymbol{x}) = \boldsymbol{J}_g(\boldsymbol{x})\boldsymbol{e}_1 - \operatorname{div} g(\boldsymbol{x})\boldsymbol{e}_1 \,. \tag{36}$$

□

Proof of Theorem 1.

Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ and $\boldsymbol{v} \in \mathbb{R}^n \backslash \{\boldsymbol{0}\}$, with $n \geq 2$.

($\Leftarrow$) Assume that $\exists g \in C^\infty(\mathbb{R}^n, \mathbb{R}^n), \forall \boldsymbol{x} \in \mathbb{R}^n, f(\boldsymbol{x}) = \boldsymbol{J}_g(\boldsymbol{x})\boldsymbol{v} - \operatorname{div} g(\boldsymbol{x})\boldsymbol{v}$.

We have

$$f_i = \sum_{j=1}^n v_j \partial_j g_i - v_i \sum_{j=1}^n \partial_j g_j \,, \tag{37}$$

hence, using the symmetry of second derivatives,

$$\operatorname{div} f = \sum_{i=1}^n \partial_i f_i = \sum_{i=1}^n \sum_{j=1}^n v_j \partial_i \partial_j g_i - \sum_{i=1}^n \sum_{j=1}^n v_i \partial_i \partial_j g_j \tag{38}$$

$$= \sum_{i=1}^n \sum_{j=1}^n v_j \partial_j \partial_i g_i - \sum_{i=1}^n \sum_{j=1}^n v_i \partial_i \partial_j g_j \tag{39}$$

$$= 0 \,. \tag{40}$$

($\Rightarrow$) Assume that f is divergence-free. Let $\boldsymbol{Q}$ be an orthogonal matrix satisfying $\boldsymbol{Q}\boldsymbol{e}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|$. Such a matrix can be obtained, for example, by completing $\boldsymbol{v}/\|\boldsymbol{v}\|$ to an orthonormal basis using the Gram–Schmidt process. Define $\tilde{f} : \boldsymbol{x} \in \mathbb{R}^n \mapsto \boldsymbol{Q}^\top f(\boldsymbol{Q}\boldsymbol{x})$. By the chain rule (Bertsekas, 1995), for all $\boldsymbol{x} \in \mathbb{R}^n$,

$$\boldsymbol{J}_{\tilde{f}}(\boldsymbol{x}) = \boldsymbol{Q}^\top \boldsymbol{J}_f(\boldsymbol{Q}\boldsymbol{x})\boldsymbol{Q} \,, \tag{41}$$

then, using the cyclic invariance of the trace,

$$\operatorname{div} \tilde{f}(\boldsymbol{x}) = \operatorname{tr}(\boldsymbol{J}_{\tilde{f}}(\boldsymbol{x})) = \operatorname{tr}(\boldsymbol{Q}^\top \boldsymbol{J}_f(\boldsymbol{Q}\boldsymbol{x})\boldsymbol{Q}) = \operatorname{tr}(\boldsymbol{Q}\boldsymbol{Q}^\top \boldsymbol{J}_f(\boldsymbol{Q}\boldsymbol{x})) = \operatorname{tr}(\boldsymbol{J}_f(\boldsymbol{Q}\boldsymbol{x})) = 0 \,, \tag{42}$$

since f is divergence-free. Therefore, $\tilde{f}$ is divergence-free. Then, according to Lemma 1, there exists $\tilde{g} \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ such that for all $\boldsymbol{x} \in \mathbb{R}^n$,

$$\tilde{f}(\boldsymbol{x}) = \boldsymbol{J}_{\tilde{g}}(\boldsymbol{x})\boldsymbol{e}_1 - \operatorname{div} \tilde{g}(\boldsymbol{x})\boldsymbol{e}_1 \,, \tag{43}$$

where $\boldsymbol{e}_1$ denotes the first canonical basis vector of $\mathbb{R}^n$.

Define $g : \boldsymbol{x} \in \mathbb{R}^n \mapsto \boldsymbol{Q}\tilde{g}(\boldsymbol{Q}^\top \boldsymbol{x})/\|\boldsymbol{v}\|$. We have, for all $\boldsymbol{x} \in \mathbb{R}^n$, $\tilde{g}(\boldsymbol{x}) = \|\boldsymbol{v}\|\boldsymbol{Q}^\top g(\boldsymbol{Q}\boldsymbol{x})$ and so, by the chain rule (Bertsekas, 1995),

$$\boldsymbol{J}_{\tilde{g}}(\boldsymbol{x}) = \|\boldsymbol{v}\|\boldsymbol{Q}^\top \boldsymbol{J}_g(\boldsymbol{Q}\boldsymbol{x})\boldsymbol{Q} \,. \tag{44}$$

In particular, $\operatorname{div} \tilde{g}(\boldsymbol{x}) = \|\boldsymbol{v}\| \operatorname{div} g(\boldsymbol{Q}\boldsymbol{x})$ and (43) rewrites

$$\boldsymbol{Q}^\top f(\boldsymbol{Q}\boldsymbol{x}) = \|\boldsymbol{v}\|\boldsymbol{Q}^\top \boldsymbol{J}_g(\boldsymbol{Q}\boldsymbol{x})\boldsymbol{Q}\boldsymbol{e}_1 - \|\boldsymbol{v}\| \operatorname{div} g(\boldsymbol{Q}\boldsymbol{x})\boldsymbol{e}_1 \,, \tag{45}$$

hence, since $\boldsymbol{Q}\boldsymbol{e}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|$,

$$f(\boldsymbol{Q}\boldsymbol{x}) = \|\boldsymbol{v}\|\boldsymbol{J}_g(\boldsymbol{Q}\boldsymbol{x})\boldsymbol{Q}\boldsymbol{e}_1 - \|\boldsymbol{v}\| \operatorname{div} g(\boldsymbol{Q}\boldsymbol{x})\boldsymbol{Q}\boldsymbol{e}_1 = \boldsymbol{J}_g(\boldsymbol{Q}\boldsymbol{x})\boldsymbol{v} - \operatorname{div} g(\boldsymbol{Q}\boldsymbol{x})\boldsymbol{v} \,. \tag{46}$$

Replacing $\boldsymbol{Q}\boldsymbol{x}$ with $\boldsymbol{x}$ yields that for all $\boldsymbol{x} \in \mathbb{R}^n$,

$$f(\boldsymbol{x}) = \boldsymbol{J}_g(\boldsymbol{x})\boldsymbol{v} - \operatorname{div} g(\boldsymbol{x})\boldsymbol{v} \,. \tag{47}$$

□

Proof of Theorem 2.

Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ and let π be a cyclic permutation of $\{1, \ldots, n\}$, with $n \geq 2$.

($\Leftarrow$) Assume that $\exists g \in C^\infty(\mathbb{R}^n, \mathbb{R}^n), \forall i \in \{1, \dots, n\},\ f_i = \partial_{\pi(i)} g_i - \partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)}$. Using the symmetry of second derivatives and the fact that, for any permutation π, $\sum_{i=1}^n a_i = \sum_{i=1}^n a_{\pi(i)}$,

$$\operatorname{div} f = \sum_{i=1}^n \partial_i f_i = \sum_{i=1}^n \partial_i \partial_{\pi(i)} g_i - \sum_{i=1}^n \partial_i \partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)} \tag{48}$$

$$= \sum_{i=1}^n \partial_i \partial_{\pi(i)} g_i - \sum_{i=1}^n \partial_{\pi(i)} \partial_i g_i \tag{49}$$

$$= 0\,. \tag{50}$$

($\Rightarrow$) Assume that f is divergence-free.

Define $g_1 : \mathbb{R}^n \to \mathbb{R}$ to be the zero function (which is infinitely differentiable) and sequentially $\forall i \in \{1, \dots, n-1\}, \forall \boldsymbol{x} \in \mathbb{R}^n$,

$$\phi_{\pi^i(1)}(\boldsymbol{x}) = \int_0^{x_{\pi^{i+1}(1)}} \left(f_{\pi^i(1)} + \partial_{\pi^{i-1}(1)} g_{\pi^{i-1}(1)}\right)(x_1, \dots, x_{\pi^{i+1}(1)-1}, \xi, x_{\pi^{i+1}(1)+1}, \dots, x_n) d\xi \tag{51}$$

and

$$g_{\pi^i(1)}(\boldsymbol{x}) = \phi_{\pi^i(1)}(\boldsymbol{x}) - \delta_{n-1,i}\Bigg(\phi_{\pi^i(1)}(0, x_2, \dots, x_n) + \int_0^{x_{\pi^i(1)}} f_1(0, \dots, x_{\pi^i(1)-1}, \xi, x_{\pi^i(1)+1}, \dots, x_n)\, d\xi\Bigg). \tag{52}$$

which are infinitely differentiable ($\delta_{i,j}$ denotes the Kronecker delta).

By construction, $\forall i \in \{1, \dots, n-1\}$,

$$\partial_{\pi^{i+1}(1)} g_{\pi^i(1)} = f_{\pi^i(1)} + \partial_{\pi^{i-1}(1)} g_{\pi^{i-1}(1)}\,, \tag{53}$$

(the correction term for $g_{\pi^{n-1}(1)}$ does not depend on $x_{\pi^n(1)} = x_1$).

Moreover, since π is a cyclic permutation, we have

$$\{\pi(1), \dots, \pi^{n-1}(1)\} = \{2, \dots, n\}\,. \tag{54}$$

Hence, $\forall i \in \{2, \dots, n\}$,

$$f_i = \partial_{\pi(i)} g_i - \partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)}\,. \tag{55}$$

Moreover, since f is divergence-free, we have, using the symmetry of second derivatives and noticing a telescoping sum

$$\partial_1 f_1 = -\sum_{i=2}^n \partial_i f_i = -\sum_{i=1}^{n-1} \partial_{\pi^i(1)} f_{\pi^i(1)} \tag{56}$$

$$= -\sum_{i=1}^{n-1} \left(\partial_{\pi^i(1)} \partial_{\pi^{i+1}(1)} g_{\pi^i(1)} - \partial_{\pi^i(1)} \partial_{\pi^{i-1}(1)} g_{\pi^{i-1}(1)}\right) \tag{57}$$

$$= -\sum_{i=1}^{n-1} \left(\partial_{\pi^i(1)} \partial_{\pi^{i+1}(1)} g_{\pi^i(1)} - \partial_{\pi^{i-1}(1)} \partial_{\pi^i(1)} g_{\pi^{i-1}(1)}\right) \tag{58}$$

$$= -\left(\partial_{\pi^{n-1}(1)} \partial_{\pi^n(1)} g_{\pi^{n-1}(1)} - \partial_{\pi^0(1)} \partial_{\pi^1(1)} g_{\pi^0(1)}\right) \tag{59}$$

$$= \partial_1 \partial_{\pi^1(1)} g_1 - \partial_{\pi^{n-1}(1)} \partial_1 g_{\pi^{n-1}(1)} \tag{60}$$

$$= -\partial_1 \partial_{\pi^{n-1}(1)} g_{\pi^{n-1}(1)}\,. \tag{61}$$

Also,

$$\partial_{\pi^{n-1}(1)} g_{\pi^{n-1}(1)}(0, x_2, \dots, x_n) \tag{62}$$

$$= \partial_{\pi^{n-1}(1)} \phi_{\pi^{n-1}(1)}(0, x_2, \dots, x_n) - \partial_{\pi^{n-1}(1)} \phi_{\pi^{n-1}(1)}(0, x_2, \dots, x_n) - f_1(0, x_2, \dots, x_n) \tag{63}$$

$$= -f_1(0, x_2, \dots, x_n)\,. \tag{64}$$

Then, $\forall \boldsymbol{x} \in \mathbb{R}^n$,

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= \int_0^{x_1} \partial_1 f_1(\xi, x_2, \dots, x_n) d\xi + f_1(0, x_2, \dots, x_n) && (65)\\
&= \int_0^{x_1} -\partial_1 \partial_{\pi^{n-1}(1)} g_{\pi^{n-1}(1)}(\xi, x_2, \dots, x_n) d\xi + f_1(0, x_2, \dots, x_n) && (66)\\
&= -\partial_{\pi^{n-1}(1)} g_{\pi^{n-1}(1)}(\boldsymbol{x}) + \partial_{\pi^{n-1}(1)} g_{\pi^{n-1}(1)}(0, x_2, \dots, x_n) + f_1(0, x_2, \dots, x_n) && (67)\\
&= -\partial_{\pi^{n-1}(1)} g_{\pi^{n-1}(1)}(\boldsymbol{x}) - f_1(0, x_2, \dots, x_n) + f_1(0, x_2, \dots, x_n) && (68)\\
&= -\partial_{\pi^{n-1}(1)} g_{\pi^{n-1}(1)}(\boldsymbol{x}) && (69)\\
&= \partial_{\pi(1)} g_1(\boldsymbol{x}) - \partial_{\pi^{n-1}(1)} g_{\pi^{n-1}(1)}(\boldsymbol{x}) \,. && (70)
\end{aligned}
$$

Finally, $\forall i \in \{1, \dots, n\}$,

$$f_i = \partial_{\pi(i)} g_i - \partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)} \,. \tag{71}$$

□

Proposition 3. *Let $\mathcal{Z} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \mathbf{1}, \boldsymbol{x} \rangle = 0\}$, where $\mathbf{1}$ denotes the all-ones vector of $\mathbb{R}^n$.*

1. *Let $\boldsymbol{u} \in \mathbb{R}^n \backslash \mathcal{Z}$.*
$$\mathcal{Z} = \mathrm{Im}(\langle \mathbf{1}, \boldsymbol{u} \rangle \boldsymbol{I} - \boldsymbol{u}\mathbf{1}^\top) \,. \tag{72}$$
2. *Let π be a cyclic permutation of $\{1, \dots, n\}$ and $\boldsymbol{P}_\pi$ denote its associated matrix.*
$$\mathcal{Z} = \mathrm{Im}(\boldsymbol{I} - \boldsymbol{P}_\pi) \,. \tag{73}$$

Proof. Both results are proved by double inclusion.

1. ($\supseteq$) Let $\boldsymbol{x} \in \mathrm{Im}(\langle \mathbf{1}, \boldsymbol{u} \rangle \boldsymbol{I} - \boldsymbol{u}\mathbf{1}^\top)$. Then, there exists $\boldsymbol{y} \in \mathbb{R}^n$ such that $\boldsymbol{x} = (\langle \mathbf{1}, \boldsymbol{u} \rangle \boldsymbol{I} - \boldsymbol{u}\mathbf{1}^\top)\boldsymbol{y}$. We have
$$\langle \mathbf{1}, \boldsymbol{x} \rangle = \langle \mathbf{1}, \langle \mathbf{1}, \boldsymbol{u} \rangle \boldsymbol{y} - \langle \mathbf{1}, \boldsymbol{y} \rangle \boldsymbol{u} \rangle = \langle \mathbf{1}, \boldsymbol{u} \rangle \langle \mathbf{1}, \boldsymbol{y} \rangle - \langle \mathbf{1}, \boldsymbol{y} \rangle \langle \mathbf{1}, \boldsymbol{u} \rangle = 0 \,, \tag{74}$$
hence, $\boldsymbol{x} \in \mathcal{Z}$.

   ($\subseteq$) Let $\boldsymbol{x} \in \mathcal{Z}$. Define $\boldsymbol{y} = \boldsymbol{x} / \langle \mathbf{1}, \boldsymbol{u} \rangle$. We have $\boldsymbol{x} = \langle \mathbf{1}, \boldsymbol{u} \rangle \boldsymbol{y} - \langle \mathbf{1}, \boldsymbol{y} \rangle \boldsymbol{u}$ since $\langle \mathbf{1}, \boldsymbol{y} \rangle = 0$. Hence, $\boldsymbol{x} = (\langle \mathbf{1}, \boldsymbol{u} \rangle \boldsymbol{I} - \boldsymbol{u}\mathbf{1}^\top)\boldsymbol{y}$ and so $\boldsymbol{x} \in \mathrm{Im}(\langle \mathbf{1}, \boldsymbol{u} \rangle \boldsymbol{I} - \boldsymbol{u}\mathbf{1}^\top)$.

2. ($\supseteq$) Let $\boldsymbol{x} \in \mathrm{Im}(\boldsymbol{I} - \boldsymbol{P}_\pi)$. Then, there exists $\boldsymbol{y} \in \mathbb{R}^n$ such that $\boldsymbol{x} = (\boldsymbol{I} - \boldsymbol{P}_\pi)\boldsymbol{y}$. We have
$$\langle \mathbf{1}, \boldsymbol{x} \rangle = \langle \mathbf{1}, \boldsymbol{y} - \boldsymbol{P}_\pi \boldsymbol{y} \rangle = \langle \mathbf{1}, \boldsymbol{y} \rangle - \langle \mathbf{1}, \boldsymbol{P}_\pi \boldsymbol{y} \rangle = \langle \mathbf{1}, \boldsymbol{y} \rangle - \langle \mathbf{1}, \boldsymbol{y} \rangle = 0 \,, \tag{75}$$
hence, $\boldsymbol{x} \in \mathcal{Z}$.

   ($\subseteq$) Let $\boldsymbol{x} \in \mathcal{Z}$. Define $y_1 = 0$ and sequentially for all $i \in \{1, \dots, n-1\}$,
$$y_{\pi^i(1)} = x_{\pi^i(1)} + y_{\pi^{i-1}(1)} \,. \tag{76}$$
   Hence, as $\boldsymbol{x} \in \mathcal{Z}$,
$$x_{\pi^n(1)} = -\sum_{i=1}^{n-1} x_{\pi^i(1)} = -\sum_{i=1}^{n-1} (y_{\pi^i(1)} - y_{\pi^{i-1}(1)}) = y_{\pi^n(1)} - y_{\pi^{n-1}(1)} \,. \tag{77}$$
   Finally, for all $i \in \{1, \dots, n\}$,
$$y_{\pi^i(1)} = x_{\pi^i(1)} + y_{\pi^{i-1}(1)} \,. \tag{78}$$
   Thus,
$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{P}_\pi \boldsymbol{y} \,, \tag{79}$$
   and so $\boldsymbol{x} = (\boldsymbol{I} - \boldsymbol{P}_\pi)\boldsymbol{y}$, hence $\boldsymbol{x} \in \mathrm{Im}(\boldsymbol{I} - \boldsymbol{P}_\pi)$.

□

Proof of Proposition 1.

($\Rightarrow$) Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ satisfying (E_1). Define $\phi = f \circ \exp$. By construction, $f = \phi \circ \log$ over $(0, +\infty)^n$. Moreover, by the chain rule (Bertsekas, 1995), $\forall \boldsymbol{x} \in \mathbb{R}^n$,

$$\nabla \phi(\boldsymbol{x}) = \mathrm{diag}(\exp \boldsymbol{x}) \nabla f(\exp \boldsymbol{x})\,, \tag{80}$$

where $\nabla f(\boldsymbol{x}) \triangleq \boldsymbol{J}_f(\boldsymbol{x})^\top$. Hence, $\forall \boldsymbol{x} \in \mathbb{R}^n$,

$$\mathrm{div}\, \phi(\boldsymbol{x}) = \mathrm{tr}(\mathrm{diag}(\exp \boldsymbol{x}) \nabla f(\exp \boldsymbol{x})) = \sum_{i=1}^n \exp(x_i) \partial_i f_i(\exp \boldsymbol{x}) = 0\,. \tag{81}$$

($\Leftarrow$) Assume that $f = \phi \circ \log$ over $(0, +\infty)^n$ with $\phi \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ satisfying (E_0).

By the chain rule (Bertsekas, 1995), $\forall \boldsymbol{x} \in (0, +\infty)^n$,

$$\nabla f(\boldsymbol{x}) = \mathrm{diag}(1/\boldsymbol{x}) \nabla \phi(\log \boldsymbol{x})\,, \tag{82}$$

where the division is understood element-wise. Hence, $\forall \boldsymbol{x} \in (0, +\infty)^n$,

$$\sum_{i=1}^n x_i \partial_i f_i(\boldsymbol{x}) = \mathrm{tr}(\mathrm{diag}(\boldsymbol{x}) \nabla f(\boldsymbol{x})) = \mathrm{tr}(\nabla \phi(\log \boldsymbol{x})) = \sum_{i=1}^n \partial_i \phi_i(\log \boldsymbol{x}) = 0\,. \tag{83}$$

□

Proof of Corollary 1.

Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ and π be a cyclic permutation of $\{1, \ldots, n\}$, with $n \geq 2$.

Using successively Proposition 1 and Theorem 2, we have

$$\begin{aligned} f \text{ satisfies } (\mathrm{E}_1) &\Leftrightarrow \exists\, \phi \in C^\infty(\mathbb{R}^n, \mathbb{R}^n) \text{ satisfying } (\mathrm{E}_0), \forall \boldsymbol{x} \in (0, +\infty)^n, \\ &\qquad f(\boldsymbol{x}) = \phi(\log \boldsymbol{x}), \qquad (84) \\ &\Leftrightarrow \exists\, \psi \in C^\infty(\mathbb{R}^n, \mathbb{R}^n), \forall i \in \{1, \ldots, n\}, \forall \boldsymbol{x} \in (0, +\infty)^n, \\ &\qquad f_i(\boldsymbol{x}) = \partial_{\pi(i)} \psi_i(\log \boldsymbol{x}) - \partial_{\pi^{-1}(i)} \psi_{\pi^{-1}(i)}(\log \boldsymbol{x}) \qquad (85) \\ &\Leftrightarrow \exists\, g \in C^\infty((0, +\infty)^n, \mathbb{R}^n), \forall i \in \{1, \ldots, n\}, \forall \boldsymbol{x} \in (0, +\infty)^n, \\ &\qquad f_i(\boldsymbol{x}) = x_{\pi(i)} \partial_{\pi(i)} g_i(\boldsymbol{x}) - x_{\pi^{-1}(i)} \partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)}(\boldsymbol{x})\,, \qquad (86) \end{aligned}$$

where $g \triangleq \psi \circ \log$. The last equivalence follows from the chain rule (Bertsekas, 1995), since $\forall \boldsymbol{x} \in (0, \infty)^n$,

$$\nabla g(\boldsymbol{x}) = \mathrm{diag}(1/\boldsymbol{x}) \nabla \psi(\log \boldsymbol{x})\,, \tag{87}$$

where the division is understood element-wise. In particular, $\forall \boldsymbol{x} \in (0, \infty)^n$,

$$\partial_{\pi(i)} g_i(\boldsymbol{x}) = \partial_{\pi(i)} \psi_i(\log \boldsymbol{x}) / x_{\pi(i)} \quad \text{and} \quad \partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)}(\boldsymbol{x}) = \partial_{\pi^{-1}(i)} \psi_{\pi^{-1}(i)}(\log \boldsymbol{x}) / x_{\pi^{-1}(i)}\,. \tag{88}$$

□

Lemma 2. *Let $f \in C^\infty((0, +\infty)^n, \mathbb{R})$ and $i < j \in \{1, \ldots, n\}$.*

$$\partial_i f = \partial_j f \tag{89}$$

if and only if $\forall \boldsymbol{x} \in (0, \infty)^n$,

$$f(\boldsymbol{x}) = f(x_1, \ldots, x_{i-1}, \frac{x_i + x_j}{2}, x_{i+1}, \ldots, x_{j-1}, \frac{x_i + x_j}{2}, x_{j+1}, \ldots, x_n)\,. \tag{90}$$

Proof. ($\Leftarrow$) Assume that (90) holds. From the chain rule, $\forall \boldsymbol{x} \in (0, \infty)^n$,

$$\begin{aligned} \partial_i f(\boldsymbol{x}) = &\frac{1}{2} \partial_i f(x_1, \ldots, x_{i-1}, \frac{x_i + x_j}{2}, x_{i+1}, \ldots, x_{j-1}, \frac{x_i + x_j}{2}, x_{j+1}, \ldots, x_n) \\ &+ \frac{1}{2} \partial_j f(x_1, \ldots, x_{i-1}, \frac{x_i + x_j}{2}, x_{i+1}, \ldots, x_{j-1}, \frac{x_i + x_j}{2}, x_{j+1}, \ldots, x_n) = \partial_j f(\boldsymbol{x})\,. \end{aligned} \tag{91}$$

($\Rightarrow$) Assume that $\partial_i f = \partial_j f$. Let $\boldsymbol{x} \in (0,\infty)^n$. Define

$$\varphi : t \in (0, x_i + x_j) \mapsto f(x_1, \dots, x_{i-1}, t, x_{i+1}, \dots, x_{j-1}, x_i + x_j - t, x_{j+1}, \dots, x_n)\,. \tag{92}$$

From the chain rule, $\forall t \in (0, x_i + x_j)$,

$$\begin{aligned}\varphi'(t) = \partial_i f(x_1, \dots, x_{i-1}, t, x_{i+1}, \dots, x_{j-1}, x_i + x_j - t, x_{j+1}, \dots, x_n)\\ - \partial_j f(x_1, \dots, x_{i-1}, t, x_{i+1}, \dots, x_{j-1}, x_i + x_j - t, x_{j+1}, \dots, x_n) = 0\,.\end{aligned} \tag{93}$$

Hence, φ is constant over $(0, x_i + x_j)$. In particular, $\varphi(x_i) = \varphi(\frac{x_i+x_j}{2})$, *i.e.*, (90) holds. □

Lemma 3. *Let $f \in C^\infty((0,+\infty)^n, \mathbb{R})$ and $i \neq j \in \{1, \dots, n\}$.*

$$\partial_i \partial_j^2 f = \partial_i^2 \partial_j f \tag{94}$$

if and only if $\exists h_1, h_2, h_3 \in C^\infty((0,\infty)^{n-1}, \mathbb{R}), \forall \boldsymbol{x} \in (0,\infty)^n$,

$$f(\boldsymbol{x}) = h_1(x_i + x_j, \boldsymbol{x}_{-i,-j}) + h_2(x_i, \boldsymbol{x}_{-i,-j}) + h_3(x_j, \boldsymbol{x}_{-i,-j})\,, \tag{95}$$

where $\boldsymbol{x}_{-i,-j}$ denotes the vector $\boldsymbol{x}$ with the i-th and j-th components omitted.

Proof. ($\Leftarrow$) Assume that $\exists h_1, h_2, h_3 \in C^\infty((0,\infty)^{n-1}, \mathbb{R}), \forall \boldsymbol{x} \in (0,\infty)^n$,

$$f(\boldsymbol{x}) = h_1(x_i + x_j, \boldsymbol{x}_{-i,-j}) + h_2(x_i, \boldsymbol{x}_{-i,-j}) + h_3(x_j, \boldsymbol{x}_{-i,-j})\,. \tag{96}$$

Then, as the partial derivatives commute,

$$\partial_i \partial_j^2 f(\boldsymbol{x}) = \partial_i \partial_j^2 h_1(x_i + x_j, \boldsymbol{x}_{-i,-j}) = \partial_i^2 \partial_j h_1(x_i + x_j, \boldsymbol{x}_{-i,-j}) = \partial_i^2 \partial_j f(\boldsymbol{x})\,, \tag{97}$$

noticing that the terms involving h_2 and h_3 vanish.

($\Rightarrow$) Assume that

$$\partial_i \partial_j^2 f = \partial_i^2 \partial_j f. \tag{98}$$

Define $q = \partial_j f - \partial_i f$. Then

$$\partial_i \partial_j q = 0. \tag{99}$$

This means that the function $\partial_j q$ is independent of x_i. Consequently, $\exists b \in C^\infty((0,\infty)^{n-1}, \mathbb{R})$,

$$\partial_j q(\boldsymbol{x}) = b(x_j, \boldsymbol{x}_{-i,-j})\,. \tag{100}$$

Then,

$$q(\boldsymbol{x}) = q(x_1, \dots, x_{j-1}, 1, x_{j+1}, \dots, x_n) + \int_1^{x_j} \partial_j q(x_1, \dots, x_{j-1}, \xi, x_{j+1}, \dots, x_n) d\xi \tag{101}$$

$$= \underbrace{q(x_1, \dots, x_{j-1}, 1, x_{j+1}, \dots, x_n)}_{A(x_i, \boldsymbol{x}_{-i,-j})} + \underbrace{\int_1^{x_j} b(\xi, \boldsymbol{x}_{-i,-j}) d\xi}_{B(x_j, \boldsymbol{x}_{-i,-j})}\,. \tag{102}$$

Hence, there exist $A, B \in C^\infty((0,+\infty)^{n-1}, \mathbb{R})$ such that

$$q(\boldsymbol{x}) = \partial_j f(\boldsymbol{x}) - \partial_i f(\boldsymbol{x}) = A(x_i, \boldsymbol{x}_{-i,-j}) + B(x_j, \boldsymbol{x}_{-i,-j})\,. \tag{103}$$

Now define

$$g : \boldsymbol{x} \in (0,+\infty)^n \mapsto f(\boldsymbol{x}) + \int_1^{x_i} A(\xi, \boldsymbol{x}_{-i,-j})\, d\xi - \int_1^{x_j} B(\xi, \boldsymbol{x}_{-i,-j})\, d\xi\,. \tag{104}$$

We have $\forall \boldsymbol{x} \in (0,\infty)^n$,

$$\partial_j g(\boldsymbol{x}) - \partial_i g(\boldsymbol{x}) = \partial_j f(\boldsymbol{x}) - \partial_i f(\boldsymbol{x}) - B(x_j, \boldsymbol{x}_{-i,-j}) - A(x_i, \boldsymbol{x}_{-i,-j}) = 0. \tag{105}$$

Hence

$$\partial_i g = \partial_j g. \tag{106}$$

Then, according to Lemma 2, $\forall \boldsymbol{x} \in (0,\infty)^n$,

$$g(\boldsymbol{x}) = g(x_1, \dots, x_{i-1}, \frac{x_i + x_j}{2}, x_{i+1}, \dots, x_{j-1}, \frac{x_i + x_j}{2}, x_{j+1}, \dots, x_n)\,. \tag{107}$$

Finally, $\forall \boldsymbol{x} \in (0,\infty)^n$,

$$f(\boldsymbol{x}) = g(x_1, \dots, x_{i-1}, \frac{x_i + x_j}{2}, x_{i+1}, \dots, x_{j-1}, \frac{x_i + x_j}{2}, x_{j+1}, \dots, x_n) \\ - \int_1^{x_i} A(\xi, \boldsymbol{x}_{-i,-j})\, d\xi + \int_1^{x_j} B(\xi, \boldsymbol{x}_{-i,-j})\, d\xi\,, \tag{108}$$

which we can rewrite, $\forall \boldsymbol{x} \in (0,\infty)^n$,

$$f(\boldsymbol{x}) = h_1(x_i + x_j, \boldsymbol{x}_{-i,-j}) + h_2(x_i, \boldsymbol{x}_{-i,-j}) + h_3(x_j, \boldsymbol{x}_{-i,-j})\,, \tag{109}$$

with $h_1, h_2, h_3 \in C^\infty((0,\infty)^{n-1}, \mathbb{R})$. □

Proof of Proposition 2.

Let $f \in C^\infty(\mathbb{R}^n, \mathbb{R}^n)$ and π be a cyclic permutation of $\{1, \dots, n\}$, with $n \geq 2$. According to Corollary 1, f satisfies (E_1) if and only if there exists $g \in C^\infty((0,+\infty)^n, \mathbb{R}^n)$ such that $\forall i \in \{1, \dots, n\}, \forall \boldsymbol{x} \in (0,\infty)^n$,

$$f_i(\boldsymbol{x}) = x_{\pi(i)} \partial_{\pi(i)} g_i(\boldsymbol{x}) - x_{\pi^{-1}(i)} \partial_{\pi^{-1}(i)} g_{\pi^{-1}(i)}(\boldsymbol{x})\,. \tag{110}$$

Moreover, we have $\forall k \geq 2$,

$$\sum_{i=1}^n x_i \partial_i^k f_i(\boldsymbol{x}) = \sum_{i=1}^n x_i x_{\pi(i)} \partial_{\pi(i)} \partial_i^k g_i(\boldsymbol{x}) - \sum_{i=1}^n x_i x_{\pi^{-1}(i)} \partial_{\pi^{-1}(i)} \partial_i^k g_{\pi^{-1}(i)}(\boldsymbol{x}) \tag{111}$$

$$= \sum_{i=1}^n x_i x_{\pi(i)} \partial_{\pi(i)} \partial_i^k g_i(\boldsymbol{x}) - \sum_{i=1}^n x_{\pi(i)} x_i \partial_i \partial_{\pi(i)}^k g_i(\boldsymbol{x}) \tag{112}$$

$$= \sum_{i=1}^n x_i x_{\pi(i)} \left(\partial_{\pi(i)} \partial_i^k g_i(\boldsymbol{x}) - \partial_i \partial_{\pi(i)}^k g_i(\boldsymbol{x}) \right)\,. \tag{113}$$

Hence, if f is of the form (110) with the additional constraint on g that $\partial_{\pi(i)} \partial_i^k g_i = \partial_i \partial_{\pi(i)}^k g_i$ for all $k \geq 2$, then f satisfies (E_k) for all $k \geq 1$. Using Lemma 3, we have $\partial_{\pi(i)} \partial_i^2 g_i = \partial_i \partial_{\pi(i)}^2 g_i$ if and only if, for all $i \in \{1, \dots, n\}$, there exist $h_1, h_2, h_3 \in C^\infty((0,\infty)^{n-1}, \mathbb{R})$ such that

$$g_i(\boldsymbol{x}) = h_1(x_i + x_{\pi(i)}, \boldsymbol{x}_{-i,-\pi(i)}) + h_2(x_i, \boldsymbol{x}_{-i,-\pi(i)}) + h_3(x_{\pi(i)}, \boldsymbol{x}_{-i,-\pi(i)})\,. \tag{114}$$

It is straightforward to verify that, when g_i has the form (114), one has $\partial_{\pi(i)} \partial_i^k g_i = \partial_i \partial_{\pi(i)}^k g_i$ for all $k \geq 3$. □