Equivariant plug-and-play image reconstruction

Supplementary Material

8. Details on the non-linear example

Derivation of $D_{\mathcal{G}}$ We have

$$\begin{aligned} D_{\mathcal{G}}(x) &= \frac{1}{|\mathcal{G}|}\sum T_g^{-1}B_2\,\mathrm{prox}_{\gamma\lambda\|\cdot\|_1}(B_1T_gx) \\ &= \frac{1}{|\mathcal{G}|}\sum T_g^{-1}(B_1+P)T_g\,\mathrm{prox}_{\gamma\lambda\|\cdot\|_1}(B_1x) \\ &= \left(B_1+\frac{1}{|\mathcal{G}|}\sum T_g^{-1}PT_g\right)\mathrm{prox}_{\gamma\lambda\|\cdot\|_1}(B_1x) \end{aligned} \tag{7}$$

yielding the desired result. The second step uses the fact that $B_2 = B_1 + P$ and that B_1 and prox are $\mathcal{G}$-equivariant functions. The third step just uses that B_1 is a $\mathcal{G}$-equivariant function.

Numerical details for Figure 2 For both the leftmost and rightmost examples, we consider the group $\mathcal{G}$ consisting of permuations of the coordinates of the vectors. This is a group with a single element g, the matrix representation of its linear application being the unitary matrix

$$T_g = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{8}$$

In the leftmost example, we use $A = \mathrm{diag}(2, 1)$, $B_1 = I$ (B_1 is thus $\mathcal{G}$-equivariant) and $\lambda = 10$. The perturbation and its $\mathcal{G}$-average are

$$P = \begin{pmatrix} -0.228 & -0.023 \\ 0.066 & 0.1 \end{pmatrix} \quad P_{\mathcal{G}} = \begin{pmatrix} -0.064 & 0.022 \\ 0.022 & -0.064 \end{pmatrix},$$

with associated norms $\|P\|_F = 0.26$, $\|P_{\mathcal{G}}\|_F = 0.10$. The (PnP) algorithm is ran with $\gamma = 5e-2$.

In the rightmost example, we use $A = \mathrm{diag}(2, 5e-4)$, $B_1 = I$ and $\lambda = 2$. The perturbation and its $\mathcal{G}$-average are

$$P = \begin{pmatrix} 0.0275 & 0.0244 \\ 0.0112 & -0.1842 \end{pmatrix}, P_{\mathcal{G}} = \begin{pmatrix} -0.0783 & 0.0178 \\ 0.0178 & -0.0783 \end{pmatrix},$$

with associated norms $\|P\|_F = 0.0469$, $\|P_{\mathcal{G}}\|_F = 0.0366$. The (PnP) algorithm is ran with $\gamma = 0.2$.

9. MC sampling and Reynolds averaging

We compare in Table 5 the performance of the equivariant architecture when training with the proposed Monte-Carlo (MC) scheme vs the true averaging. It shows no difference in final performance while the MC strategy decrease the computational complexity by a factor 4.

Architecture	Dataset	Monte-Carlo Sample	Reynolds Average
DnCNN	BSD10	30.698 ± 1.645	30.684 ± 1.645
DRUNet	fastMRI	30.678 ± 0.740	30.646 ± 0.752
LipDnCNN	Set3C	32.705 ± 0.868	32.706 ± 0.868

Table 5. Performance of algorithms from Fig. 4 when relying on Monte-Carlo estimates and averaged equivariant architectures.

10. Equivariant algorithms

The equivariant counterpart of (PnP) is

$$\begin{aligned} &\text{Sample } g_k \sim \mathcal{G} \\ &\text{Set } \widetilde{\mathrm{D}}_{\mathcal{G},k}(x) = T_{g_k}^{-1}\,\mathrm{D}(T_{g_k}x) \\ &x_{k+1} = \widetilde{\mathrm{D}}_{\mathcal{G},k}\left(x_k - \gamma A^\top(Ax_k - y)\right). \end{aligned} \tag{eq. PnP}$$

The equivariant counterpart of (RED) is

$$\begin{aligned} &\text{Sample } g_k \sim \mathcal{G} \\ &\text{Set } \widetilde{\mathrm{D}}_{\mathcal{G},k}(x) = T_{g_k}^{-1}\,\mathrm{D}(T_{g_k}x) \\ &x_{k+1} = x_k - \gamma A^\top(Ax_k - y) \\ &\qquad - \gamma\lambda(x_k - \widetilde{\mathrm{D}}_{\mathcal{G},k}(x_k)). \end{aligned} \tag{eq. RED}$$

The equivariant counterpart of (ULA) is

$$\begin{aligned} &\text{Sample } g_k \sim \mathcal{G} \\ &\text{Set } \widetilde{\mathrm{D}}_{\mathcal{G},k}(x) = T_{g_k}^{-1}\,\mathrm{D}(T_{g_k}x) \\ &x_{k+1} = x_k - \gamma A^\top(Ax_k - y) \\ &\qquad - \gamma\lambda(x_k - \widetilde{\mathrm{D}}_{\mathcal{G},k}(x_k)) + \sqrt{2\gamma}\epsilon_k. \end{aligned} \tag{eq. ULA}$$