Appendix

A Additional experimental results

In this section, we complement the experimental investigation of Section 5 with additional results.

A.1 Additional CartoonSet results and discussion

Pareto frontier: In Fig. 13, we show Pareto-optimality[1] of subspace methods. Both Sub-DIP NGD and L-BFGS show performances superior to DIP and E-DIP in terms of time to convergence and conv. PSNR, for all the studied levels of problem ill-posedness (that is, with respect to the number of angles). This confirms observations made in Section 5 and complements results in Fig. 2. We note that conv. is based on the stopping criterion in Algorithm 1. Interestingly, the time required to reach the conv. PSNR values is for all the studied methods largely independent of the number of observation angles. Just as expected, all methods' conv. PSNR is higher for more well-conditioned settings. Furthermore, if the time at max PSNR is considered, as in Fig. 14, then E-DIP is within the Pareto optimality frontier. In Fig. 15 we show three examples of reconstructions on the CartoonSet for the three angle settings in the ablative study in Section 5.1. Even for the sparsest view (45 angles), Sub-DIP reconstructions exhibit barely any noise.

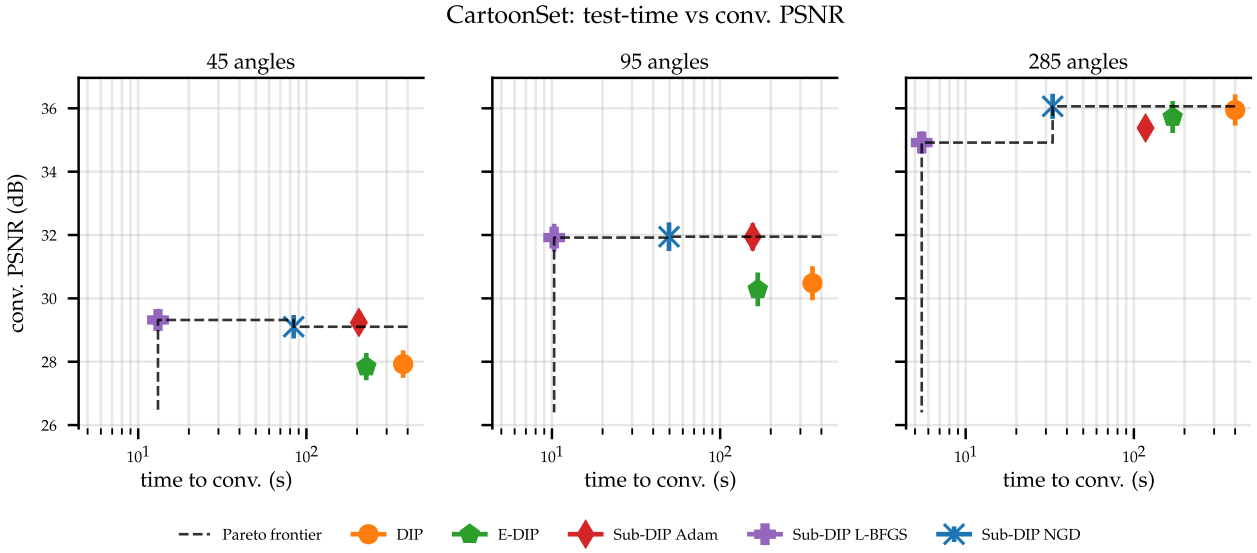

Figure 13: Pareto-curves of conv. PSNR vs optimisation time on the CartoonSet for reconstructions from 45 angles (left), 95 angles (middle) and 285 angles (right). We provide mean and standard deviation of the PSNR, computed across 50 cartoon images. Note that the x-axis is given in log-scale and that the std is hard to observe due to the shared range along the y-axis.

Selection and construction of subspaces: Firstly, we compare the reconstruction quality of Sub-DIP NGD using an SVD basis extracted from a pre-training trajectory (as detailed in the paper) with Sub-DIP NGD using a randomly sampled unit-norm basis of equal dimensionality. The goal is to investigate if there is a benefit. Fig. 16 shows PSNR reconstruction trajectories, averaged over 25 images. The result confirms that using a subspace based on the pre-training trajectory has a clearly superior performance, justifying the choices made in the paper. The effect is more pronounced in the less ill-posed problems (that is, as the number of angles increases).

In Fig. 17, we compare the reconstruction quality with respect to the numerical scheme that is used to compute the utilised low-dimensional SVD space. Specifically, our comparison involves the traditional SVD

[1]In this work, when we refer to Pareto-optimality, we adopt a pragmatic perspective, indicating that the method is positioned along the Pareto frontier as recognised with the investigated methods.

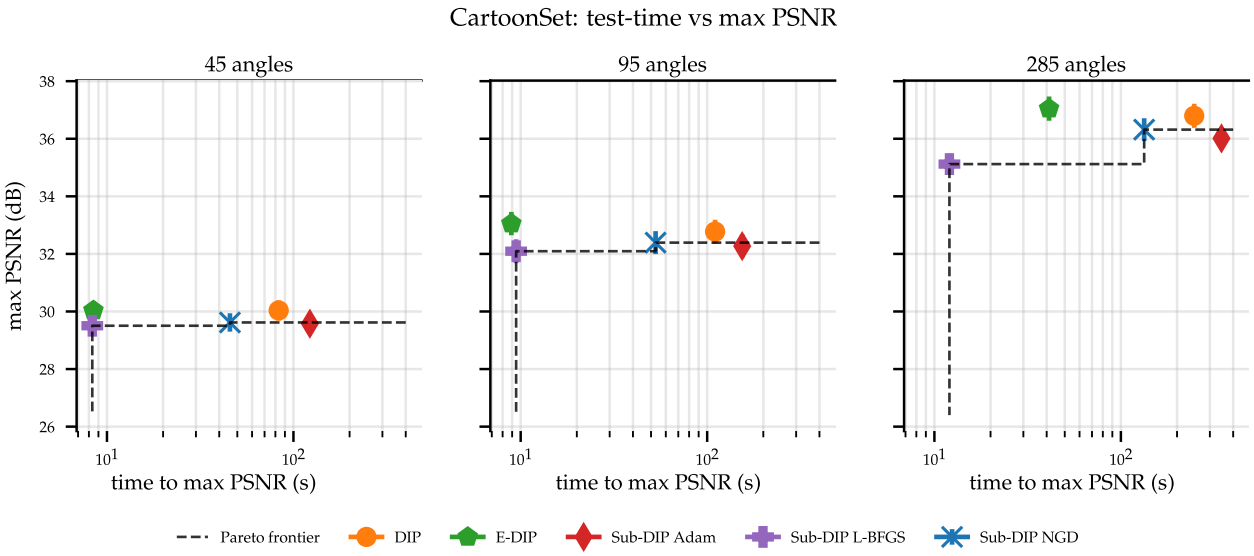

Figure 14: Pareto-curves of max. PSNR vs optimisation time on the CartoonSet for reconstructions from 45 angles (left), 95 angles (middle) and 285 angles (right). We provide mean and standard deviation of the PSNR computed across 50 cartoon images. Note that the x-axis is given in log-scale and that the std is hard to observe due to the shared range along the y-axis.

method, which entails explicitly constructing a matrix of parameters sampled at various points along the training trajectory, followed by computing its SVD. We contrast this with the more computationally efficient approach known as incremental SVD (Brand, 2002). The results for max and conv. PSNR (averaged over 25 images) report that the two methods show on par performance.

Subspace sensitivity to shift in the forward operator: In this section we carry out an investigation on how sensitivity to shift in the forward model the subspace extraction is. We carry out this investigation on CartoonSet, to have it complementary to the ablative analysis.

In Fig. 18, we report the PSNR mean and standard deviation over 25 CartoonSet images reconstructing from 45, 95, and 285 angles, using subspaces extracted on 45 and 285 angles. We extract the subspace using 45 angles and then reconstruction to settings of 95 and 285; additionally, we explore the scenario of extracting the subspace using 285 angles and testing it with 45 and 95 angles. Our findings reveal that transferring the subspace extracted at 285 angles for testing on 45 and 95 angles yields max PSNR values nearly identical to those obtained when extracting and testing on 45 or 95 angles separately. This demonstrates that the subspace we have identified has excellent transfer properties, making our method highly effective for tasks involving reconstruction. This is especially true in scenarios where there is expected variability in the forward operator between reconstructive tasks. On a minor note, it is worth mentioning that extracting at 45 angles and testing at 285 results in a slight decrease of approximately 0.5 dB in max PSNR.

A.2 Additional μCT Walnut results and discussion

Optimisation trajectories: Fig. 19 shows optimisation trajectories of the used optimisers on the Walnut dataset, cf. Section 5.2. We study the optimisation behaviour in terms of PSNR vs time, PSNR vs steps and loss vs steps. As discussed in the main text, second order subspace methods converge in less time than first order method. The rightmost plot shows that loss functions for DIP and E-DIP decrease at a constant rate in the log of the number of steps, even after passing their PSNR peak. In contrast, loss curves of subspace methods saturate at their minimum values (at around 7×10^{-2}), which coincides with their max PSNR.

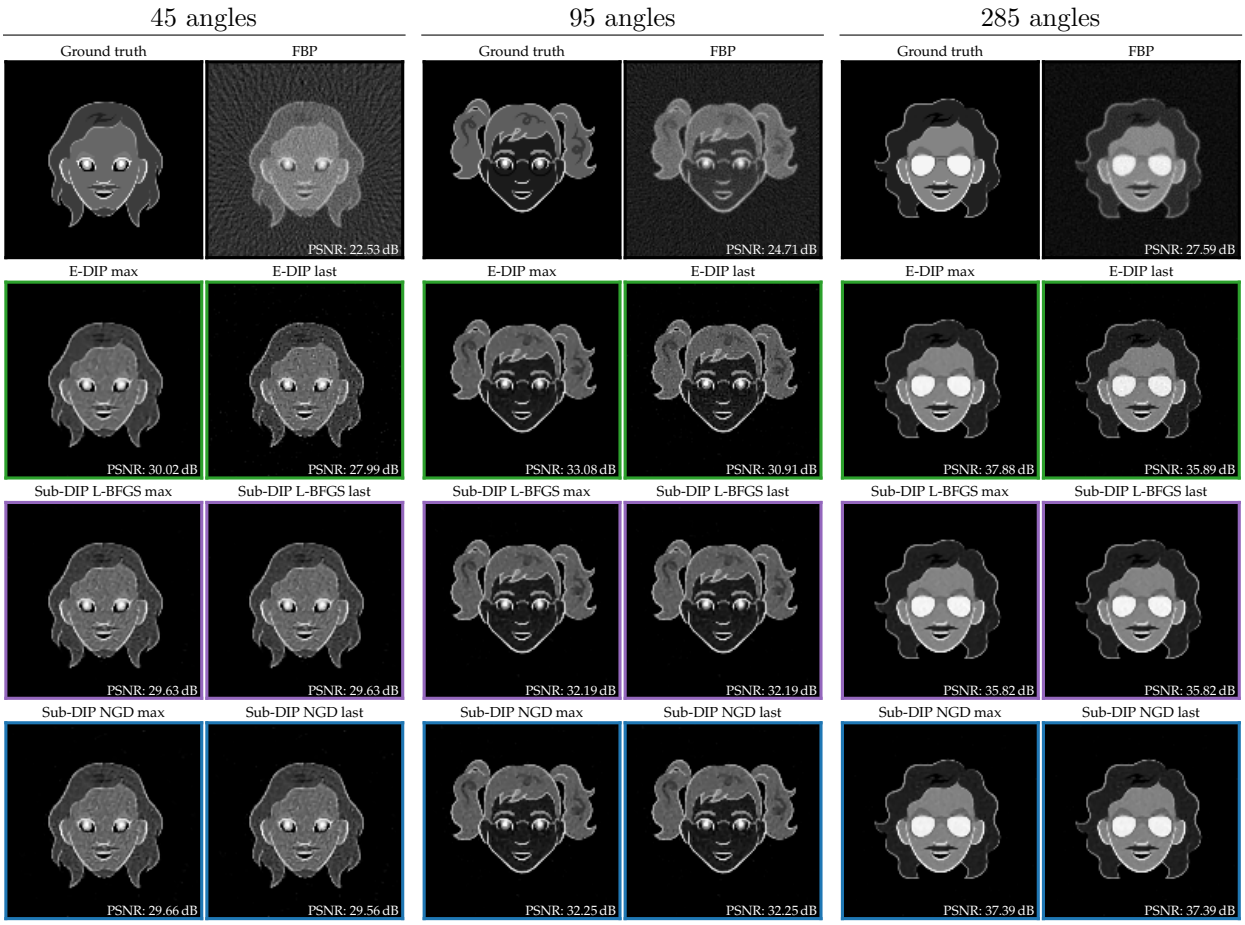

Figure 15: Reconstructions for the CartoonSet, from 45, 95 and 285 acquisition angles for different example images and DIP methods.

A.3 Additional Mayo results and discussion

Optimisation trajectories: Figs. 20 and 21 show the optimisation trajectories for the Mayo Clinic dataset on 100 and 300 angle CT tasks, respectively. Figures on the left study the PSNR with respect to elapsed time; figures in the middle study PSNR vs number of optimisation steps, and figures on the right study optimisation loss vs number of optimisation steps. The results consistently show that second order subspace methods exhibit fast and stable convergence, with no observable performance degradation. On the other hand, DIP and E-DIP show high max PSNR, but subsequently overfit to noise, as expected. Moreover, in Fig. 21 we compare the performance of Sub-DIP NGD and E-DIP, where θ_{pre} and U are obtained using a dataset of images of a similar distribution and structure to the Mayo dataset.

Namely, dashed lines indicate optimisation trajectories with the initial parameters and extracted basis selected through pre-training on the LoDoPaB dataset (Leuschner et al., 2021). The results indicate that this, task-specific, pre-training allows faster and overall improved performance behaviour.

A.4 Additional Set5 results and discussion

We have two more figures to complement Fig. 11: Fig. 22 and Fig. 23. These figures explore the relationship between the subspace dimension and the level of noise in the data. This exploration is conducted for all five images in the Set5 collection. Additionally, we have conducted the same investigation using two methods: Sub-DIP NGD and Sub-DIP Adam.

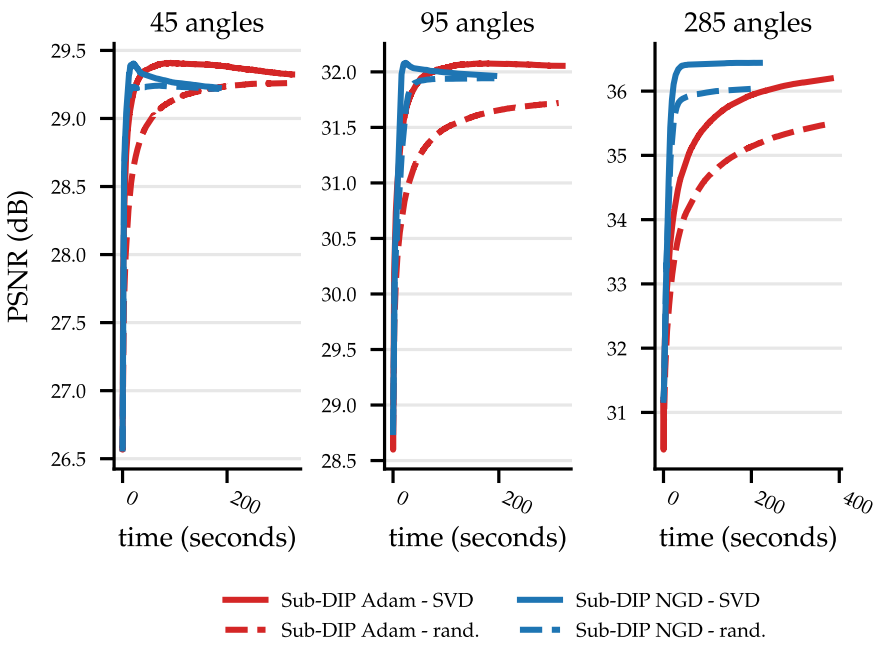

Figure 16: Average PSNR trajectories for 45, 95, and 285 angles, comparing a randomly selected low-dimensional subspace, and a subspace computed through SVD on the pre-training trajectory, on Sub-DIP Adam and Sub-DIP NGD.

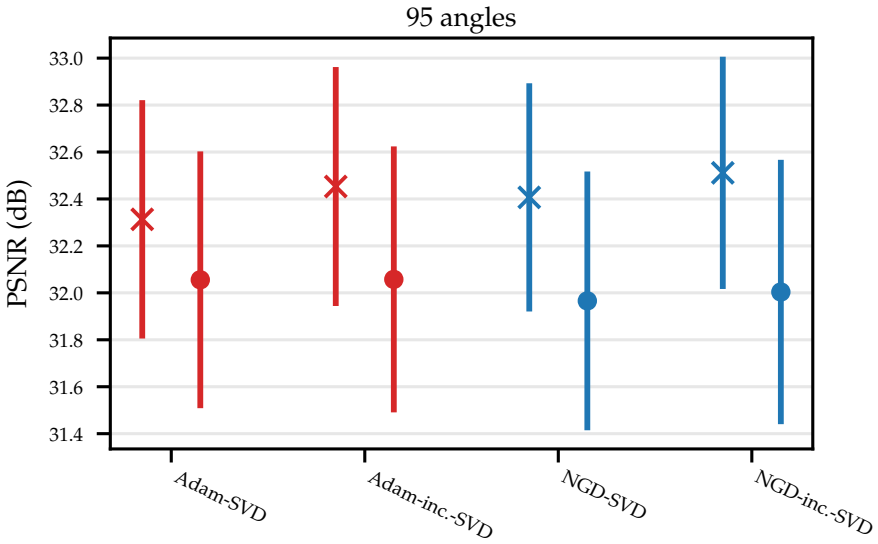

Figure 17: Comparison of max and conv. PSNR of traditional and incremental SVD approaches, used to construct the pre-training subspace, for Sub-DIP Adam and Sub-DIP NGD.

The image reconstruction metrics PSNR and SSIM for Sub-DIP NGD and Sub-DIP Adam are tabulated in Table 1 and Table 2. PSNR has been the standard reconstruction metric in many applications in both industry and research, and has been used in most previous studies on DIP. However, SSIM is an image reconstruction metric claimed to capture perceptual quality better than PSNR. We include tables with SSIM and PSNR values for the Set 5 dataset.

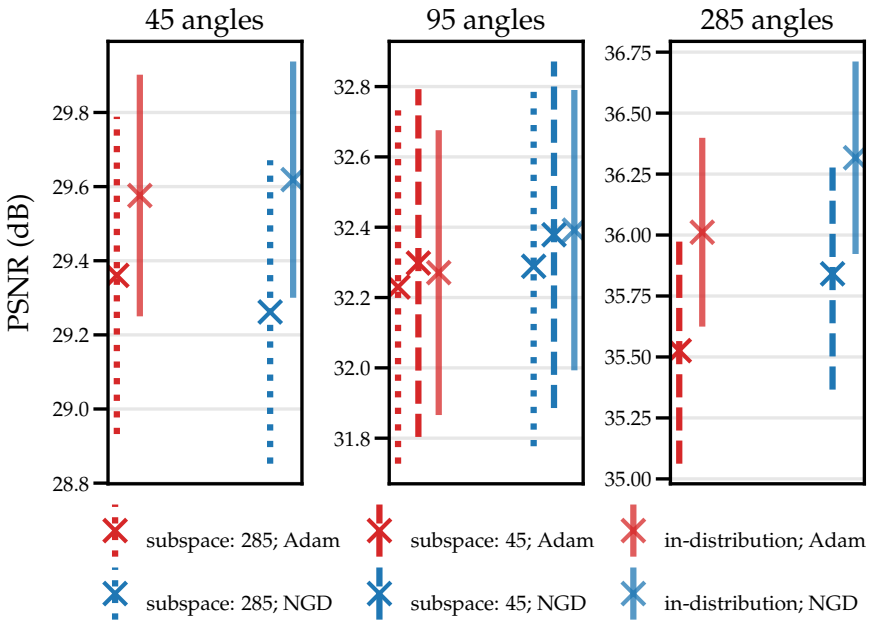

Figure 18: PSNR mean and standard deviation over 25 CartoonSet images from 45, 95, and 285 angles, using subspaces extracted on 45 and 285 angles. Note that in-distribution implies that the subspace is extracted on the number of angles used in the respective reconstructive task, and here is reported as a baseline to assess the affect of the transfer; intuitively, this presents an upper bound on what you could achieve in the transfer setting.

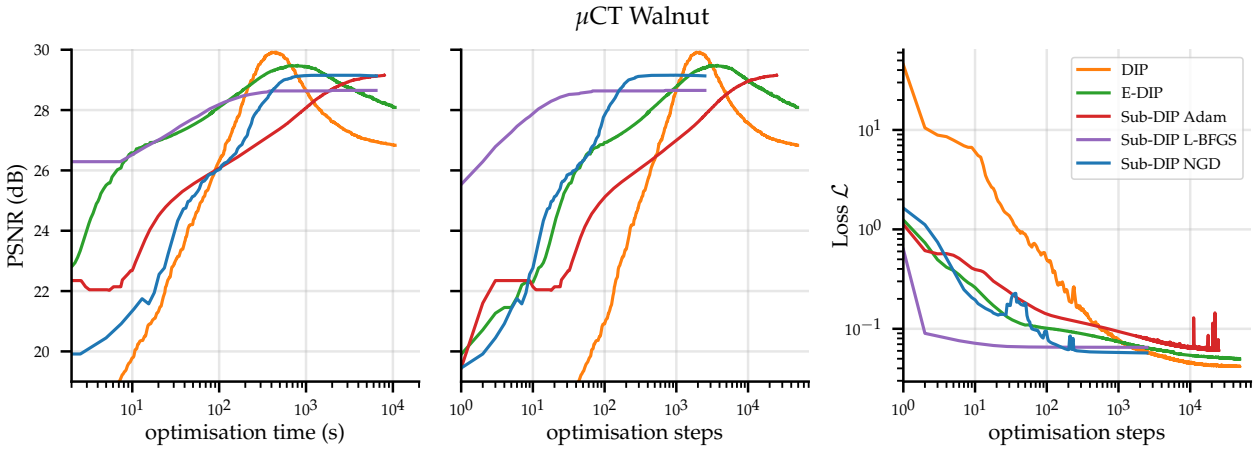

Figure 19: The training curves of DIP, E-DIP and three versions of Sub-DIP on the µCT Walnut dataset. PSNR vs time (left), PSNR vs steps (middle) and loss vs steps (right). All curves (expect for E-DIP) are averaged over 3 seeds, which affect the initialisation point for the U-Nets, initialisation of the subspace parameters and random probes used for NGD. Note that the x-axis is given in log-scale.

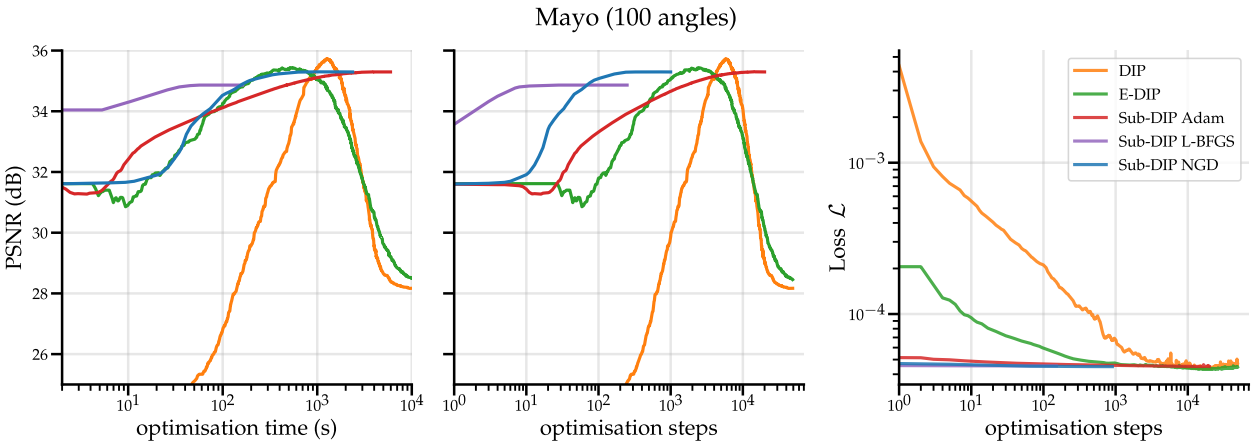

Figure 20: The optimisation curves of DIP, E-DIP and three versions of Sub-DIP on the 100 angle Mayo dataset. PSNR vs time (left), PSNR vs steps (middle) and loss vs steps (right). All curves are averaged over 10 images. Note that the x-axis is given in log-scale.

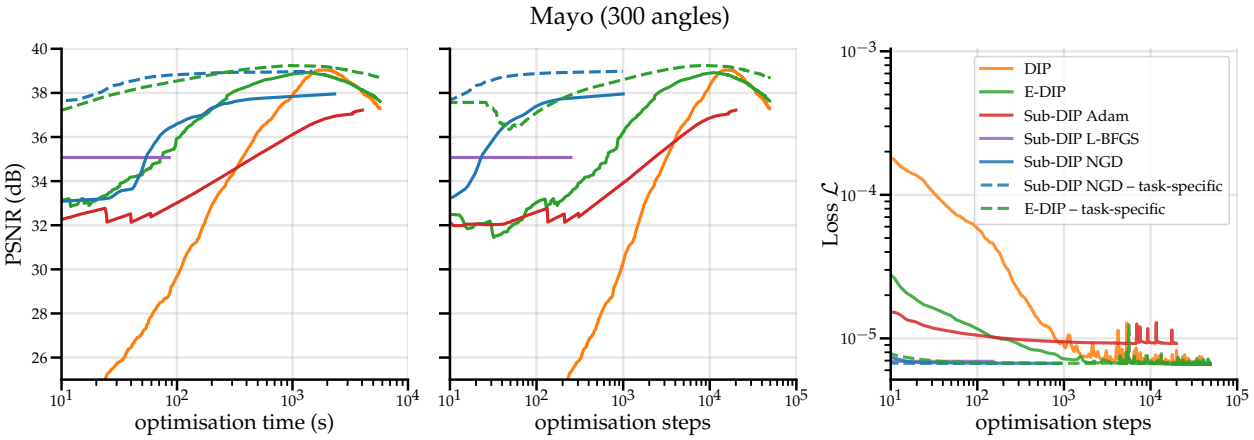

Figure 21: The optimisation curves of DIP, E-DIP and three versions of Sub-DIP on the 300 angle Mayo dataset. PSNR vs time (left), PSNR vs steps (middle) and loss vs steps (right). All curves are averaged over 10 images. Note that the x-axis is given in log-scale.

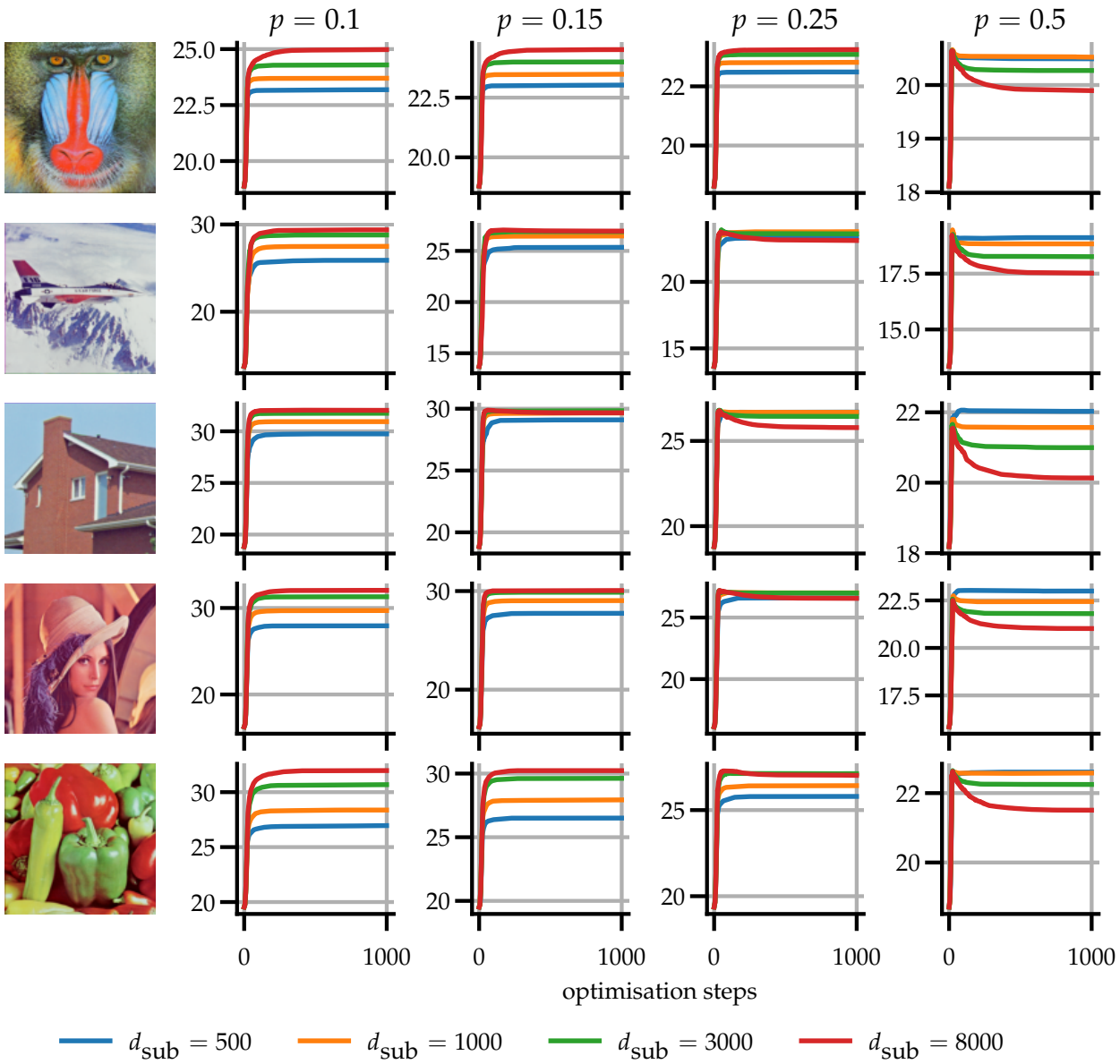

Figure 22: Investigation of the regularising effect of the dimensionality of the chosen subspace on Set5. We report PSNR trajectories using Sub-DIP NGD. Our analysis encompasses four distinct noise levels and four dimensions of the subspace.

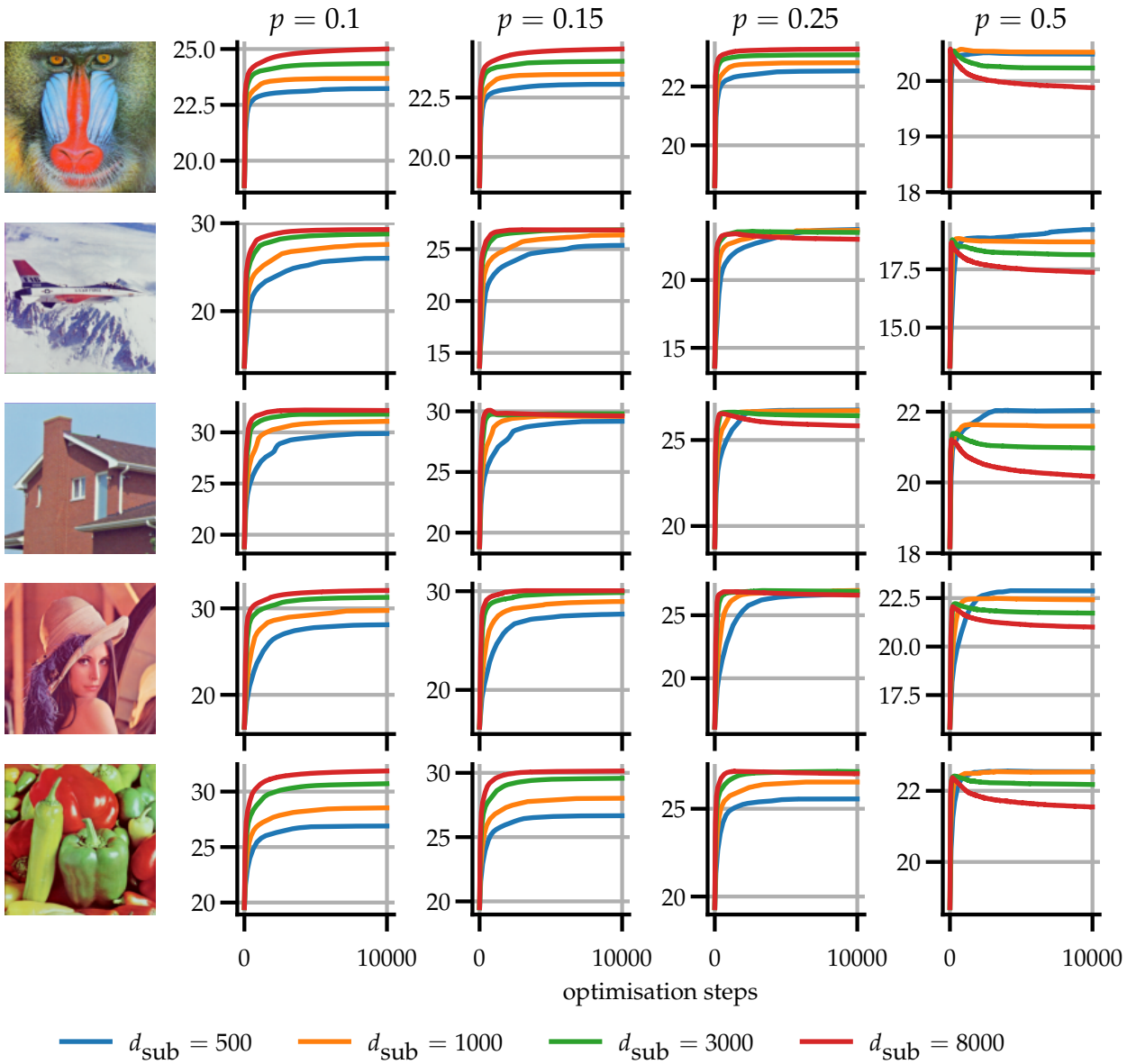

Figure 23: Investigation of the regularising effect of the dimensionality of the chosen subspace on Set5. We report PSNR trajectories using Sub-DIP Adam. Our analysis encompasses three distinct noise levels and four dimensions of the subspace

Table 1: Image reconstruction metrics for the denoising experiments on Set5. Values are reported for the reconstruction with the minimum loss value among all iterations; unless the optimisation is unstable, this is near the last iteration. Therefore we exclude the baseline DIP from this table, as it is highly over-fitting without early stopping.

		$p = 0.1$		$p = 0.15$		$p = 0.25$		$p = 0.5$	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
baboon	Sub-DIP Adam	25.00	0.765	24.54	0.743	23.27	0.674	20.52	0.502
baboon	Sub-DIP NGD	24.98	0.764	24.51	0.740	23.24	0.673	20.52	0.503
jet F16	Sub-DIP Adam	29.28	0.792	26.84	0.672	23.01	0.497	18.68	0.340
jet F16	Sub-DIP NGD	29.38	0.801	26.94	0.680	23.11	0.504	18.83	0.343
house	Sub-DIP Adam	32.14	0.804	29.62	0.691	25.83	0.518	21.59	0.356
house	Sub-DIP NGD	32.03	0.789	29.67	0.690	25.78	0.517	21.57	0.354
Lena	Sub-DIP Adam	32.07	0.878	30.04	0.808	26.60	0.662	22.41	0.482
Lena	Sub-DIP NGD	32.04	0.880	30.06	0.810	26.56	0.661	22.44	0.484
peppers	Sub-DIP Adam	31.85	0.888	30.15	0.831	27.00	0.708	22.53	0.531
peppers	Sub-DIP NGD	31.96	0.890	30.23	0.834	27.02	0.708	22.58	0.534

Table 2: Image reconstruction metrics for the deblurring experiments on Set5. Values are reported for the reconstruction with the minimum loss value among all iterations; unless the optimisation is unstable, this is near the last iteration. Therefore we exclude the baseline DIP from this table, as it is highly over-fitting without early stopping.

		$\kappa = 0.8$		$\kappa = 1.6$	
		PSNR	SSIM	PSNR	SSIM
baboon	Sub-DIP Adam	24.39	0.735	22.45	0.605
baboon	Sub-DIP NGD	24.40	0.736	22.35	0.601
jet F16	Sub-DIP Adam	29.47	0.852	23.02	0.507
jet F16	Sub-DIP NGD	29.65	0.861	23.65	0.585
house	Sub-DIP Adam	32.53	0.820	25.87	0.537
house	Sub-DIP NGD	32.61	0.819	25.99	0.554
Lena	Sub-DIP Adam	31.88	0.895	26.79	0.691
Lena	Sub-DIP NGD	31.74	0.898	27.26	0.731
peppers	Sub-DIP Adam	31.61	0.905	27.56	0.787
peppers	Sub-DIP NGD	31.83	0.908	27.59	0.799

B Description of our implementation of Natural Gradient Descent

The standard NGD update rule (Amari, 2013) for a smooth function $\mathcal{L}_\gamma$, is given by

$$c_{t+1} = c_t - \alpha_t \tilde{F}(c_t)^{-1} \nabla \mathcal{L}_\gamma(c_t), \tag{10}$$

where $\nabla \mathcal{L}_\gamma$ is the gradient of the loss function (including the TV regulariser's contribution), $\alpha_t > 0$ is a step-size and $\tilde{F}(c_t)$ is the exact Fisher information matrix (FIM) computed at c_t. Ignoring the contribution of the regulariser, and denoting $J_f := \nabla_\theta f(x^\dagger, \theta)\,|_{\theta=\gamma(c)}$, FIM is of the form

$$\tilde{F}(c) = (AJ_f MU)^\top A J_f MU. \tag{11}$$

We derive the FIM with respect to θ from its definition as the expected outer-product between gradients of the projection error term, see (2). The gradient of the data-fitting term with respect to θ is

$$\begin{aligned} \nabla_\theta \left[\tfrac{1}{2}\|Af(x^\dagger,\theta) - y\|_2^2\right] &= \nabla_x \left[\tfrac{1}{2}\|Ax - y\|_2^2\right]\big|_{x=f(x^\dagger,\theta)} \nabla_\theta f(x^\dagger,\theta) \\ &= (Af(x^\dagger,\theta) - y)^\top A J_f, \end{aligned} \tag{12}$$

with now $J_f = \nabla_\theta f(x^\dagger, \theta)$. By definition, we then have

$$\begin{aligned} \tilde{F}(\theta) &= \mathbb{E}_{v\sim\mathcal{N}(\mu,I_{d_y})}\left[(J_f^\top A^\top (Af(x^\dagger,\theta) - v))^{\otimes^2}\right] = \mathbb{E}_{v\sim\mathcal{N}(\mu,I_{d_y})}\left[J_f^\top A^\top (Af(x^\dagger,\theta) - v)^{\otimes^2} A J_f\right] \\ &= J_f^\top A^\top \mathbb{E}_{v\sim\mathcal{N}(\mu,I_{d_y})}\left[(Af(x^\dagger,\theta) - v)^{\otimes^2}\right] A J_f, \end{aligned} \tag{13}$$

where μ is defined as $Af(x^\dagger,\theta)$ and $\otimes^2$ denotes the outer product of a vector v with itself (i.e., $z^{\otimes^2} = zz^\top$). The expectation in (13) simplifies to

$$\begin{aligned} \mathbb{E}_{v\sim\mathcal{N}(\mu,I_{d_y})}\left[(Af(x^\dagger,\theta) - v)^{\otimes^2}\right] &= \mathbb{E}_{v\sim\mathcal{N}(\mu,I_{d_y})}\left[Af(x^\dagger,\theta) f(x^\dagger,\theta)^\top A^\top + vv^\top - 2vf(x^\dagger,\theta)^\top A^\top\right] \\ &= Af(x^\dagger,\theta) f(x^\dagger,\theta)^\top A^\top + \mathbb{E}_{v\sim\mathcal{N}(\mu,I_{d_y})}[vv^\top] - 2\mathbb{E}_{v\sim\mathcal{N}(\mu,I_{d_y})}[v] f(x^\dagger,\theta)^\top A^\top \\ &= I_{d_y}. \end{aligned}$$

Thus, the FIM with respect to θ is given by

$$\tilde{F}(\theta) = J_f^\top A^\top A J_f. \tag{14}$$

Alternatively, the equivalence between NGD and generalised Gauss-Newton (GGN) methods (Kunstner et al., 2019; Schraudolph, 2002), valid for exponential family likelihoods (Kunstner et al., 2019), can be exploited for this problem. Namely, the data fidelity is proportional to the negative exponential log-likelihood under the noise model in (1). The Hessian H of the data fidelity with respect to θ is given by

$$\nabla_\theta^2 \|Af(x^\dagger,\theta) - y\|_2^2 = H_f^\top A^\top A f(x^\dagger,\theta) - H_f^\top A^\top y + J_f^\top A^\top A J_f, \tag{15}$$

where $H_f := \nabla_\theta^2 f(x^\dagger,\theta) \in \mathbb{R}^{d_\theta \times d_\theta}$ is the network Hessian. GGN methods are then recovered by ignoring the second order terms in (15), giving

$$G(\theta) = J_f^\top A^\top A J_f = \tilde{F}(\theta). \tag{16}$$

Note that (11) is then trivially recovered from (14) or (16) by introducing the network reparametrisation in (5). Namely, an analogous computation yields

$$\nabla_c \left[\tfrac{1}{2}\|Af(x^\dagger,\gamma(c)) - y\|_2^2\right] = (Af(x^\dagger,\gamma(c)) - y)^\top A J_f MU. \tag{17}$$

Plugging this in and taking the expectation recovers (11).

Assuming the NGD-GGN equivalence, the curvature of DIP loss (2) has no contribution coming from the TV regulariser, since the latter consists of the absolute value of the finite differences of pixel values, cf. (3), and thus almost everywhere has zero second derivatives. Note also that for image restoration tasks in Section 5.4, TV regularisation is not utilised.

We depart from (10) in two ways. First, we use a stochastic estimate of the FIM and second, we relax the update rule by adding a number of hyperparameters. These are set adaptively with a modified version of (Martens & Grosse, 2015a)'s Levenberg–Marquardt-style algorithm, described below.

B.1 Stochastic update of the Fisher information matrix

As indicated in Section 4, we compute a Monte-Carlo estimate of the FIM at step t as

$$\hat{F}_t = \frac{1}{n}\sum_{i=1}^{n}(z_i^\top AJ_fMU)^\top z_i^\top AJ_fMU, \tag{18}$$

with $J_f \coloneqq \nabla_\theta f(x^\dagger, \gamma(c_t)) \in \mathbb{R}^{d_x \times d_\theta}$ being the Jacobian of the U-Net at the current full-dimensional parameter vector, and n the number of random probes $z_i \sim \mathcal{N}(0, I_{d_y})$. This aims at overcoming the computational intractability arising from J_f. That is, we approximate the matrix–matrix multiplications in (10) via Monte-Carlo sampling (Martinsson & Tropp, 2020). Then we update the FIM moving average as

$$F_{t+1} = \beta F_t + (1-\beta)\hat{F}_t \quad \text{and} \quad \beta \in (0,1). \tag{19}$$

We use our *online FIM estimate* F_t to estimate the descent direction according to the natural gradient at c_t as $\Delta_t = -F_t^{-1}\nabla_c\mathcal{L}_\gamma(c_t)$.

To evaluate (18) we use $n = 100$ probes per optimisation step for the CartoonSet ablative study in Section 5.1. On the Walnut and Mayo datasets (in Section 5.2 and Section 5.3), due to the increased computational cost of the Jacobian vector products, we only use $n = 50$ probes per optimisation step. Finally, across all experiments, to update the FIM moving average, cf. (19), β is kept fixed to 0.95.

B.2 Adaptive fine-tuning of FIM hyperparameters

To compute a new parameter setting $c_t + \delta$, we choose δ to locally minimise a quadratic model of the training objective $\mathcal{L}_\gamma$, defined as

$$M_t(\delta) = \mathcal{L}_\gamma(c_t) + \nabla_c\mathcal{L}_\gamma(c_t)^\top\delta + \frac{s}{2}\delta^\top(\lambda I_{d_{\text{sub}}} + \tilde{F}(c_t))\delta. \tag{20}$$

Since $\tilde{F}(c_t)$ is the FIM (guaranteed PSD) and not the Hessian, the above can be seen as a convex approximation of the second order Taylor series expansion of $\mathcal{L}_\gamma$ at c_t. Since neural network loss-functions are non-quadratic and non-convex, the FIM may provide a poor approximation to the loss. To correct for this, we introduce two parameters, s and λ, leading to the modified curvature $s(\lambda I_{d_{\text{sub}}} + \tilde{F})$. The parameter $\lambda > 0$ ensures the positive definiteness of the FIM that may be violated due to numerical instabilities, and also provides an isotropic increase in curvature (Martens, 2020), limiting the norm of the loss gradient and bringing it closer to the steepest descent direction. Novel to our method is the scaling parameter $s \in (0,1)$, which can reduce the effect of the curvature on the quadratic model, thus allowing larger step-sizes.

We employ the Levenberg-Marquardt style methodology, see Martens & Sutskever (2012, Section 8.5), to update both the damping parameter λ and the scaling parameter s. This involves computing

$$\rho = \frac{\mathcal{L}_\gamma(c_t + \Delta_t) - \mathcal{L}_\gamma(c_t)}{M_t(\Delta_t) - M_t(0)}. \tag{21}$$

Thus, for ρ close to 1 the quadratic model is good at approximating the objective, and if ρ is substantially smaller than 1 then it is a poor estimator. Following Martens & Sutskever (2012), ρ is evaluated every $T = 5$ iterations. If $\rho < 0.25$ the damping is updated via $\lambda \leftarrow \left(\frac{3}{4}\right)^{-T}\lambda$. Conversely, if $\rho > 0.75$ then $\lambda \leftarrow \left(\frac{3}{4}\right)^{T}\lambda$. If needed, the resulting value is clipped to ensure it stays in the interval $[\lambda_{\min}, 100]$.

Across all experiments, the damping coefficient λ is initialised to 100. The different dimensionality of the subspace d_{sub} necessitates adjusting $\lambda_{\min}$ in order to avoid numerical instabilities when solving the linear system against $F(c_t)$, required to compute the update direction Δ_t. Due to the small dimensionality of d_{sub} used in the CartoonSet ablative study, $\lambda_{\min}$ is set to 10^{-8}. On the Walnut and the Mayo data, we set $\lambda_{\min}$ to 1 due to the high-dimensionality of the considered subspace.

Ideally, the parameter s would be small during the early iterations and would increase towards 1 as we approach the optimum, where we want optimisation to slow down. We use a similar update condition: if

$\rho < 0.95$ then $s \leftarrow \left(\frac{3}{4}\right)^{-T} s$ and if $\rho > 1.05$ then $s \leftarrow \left(\frac{3}{4}\right)^{T} s$. If needed, the resulting value is clipped to ensure it stays in the interval $[s_{\min}, 1]$. Note that the rule under which s is updated is much tighter that the one used for the damping parameter λ. This is done to ensure larger step-sizes can be taken in the early stages of the optimisation, speeding up the convergence.

We set $s_{\min}$ to 10^{-3} for the CartoonSet (across all three angles setting). For the Walnut and the Mayo datasets, as well as for the image restoration tasks, we set $s_{\min}$ to 5×10^{-6}.

B.3 Getting the final parameter update

We further speed up the convergence by introducing a momentum update, as in Scarpetta et al. (1999); Martens & Grosse (2015a). This results in update directions of the form $\delta = \alpha_t \Delta_t + \mu_t \delta_0$, where $\Delta_t = -F_t^{-1} \nabla_c \mathcal{L}_\gamma(c_t)$, F_t is our moving average estimate of the FIM, and δ_0 is the direction of the previous update. Coefficients are then updated as $c_{t+1} = c_t + \delta$. Parameters α_t and μ_t are chosen by minimising the local quadratic model M_t. Plugging such a δ into M_t and minimising over α_t and μ_t gives a two-dimensional linear system

$$\begin{pmatrix} \alpha_t \\ \mu_t \end{pmatrix} = -s^{-1} \begin{pmatrix} \Delta_t^\top \tilde{F}(c_t) \Delta_t + \lambda \|\Delta_t\|_2^2 & \Delta_t \tilde{F}(c_t) \delta_0 + \lambda \Delta_t^\top \delta_0 \\ \Delta_t^\top \tilde{F}(c_t) \delta_0 + \lambda \Delta_t^\top \delta_0 & \delta_0^\top \tilde{F}(c_t) \delta_0 + \lambda \|\delta_0\|_2^2 \end{pmatrix}^{-1} \begin{pmatrix} \nabla \mathcal{L}_\gamma(c_t)^\top \Delta_t \\ \nabla \mathcal{L}_\gamma(c_t)^\top \delta_0 \end{pmatrix}. \tag{22}$$

Note that, although it is not tractable to compute the full FIM at every optimisation step and we use a rolling estimate F_t, we may interact with the true FIM $\tilde{F}$ through matrix vector products. This allows solving the above systems quickly.

C Additional experimental setup description

C.1 Raw PSNR vs min-loss PSNR

All the reported PSNR values are obtained using the min-loss PSNR strategy standard in DIP literature (Baguer et al., 2020). For sparse problems, both the training loss and "raw" reconstruction PSNR can exhibit very rapidly varying behaviour across optimisation steps. In order to display PSNR values, at each optimisation step, we define the min-loss PSNR as the PSNR corresponding to the time-step with lowest training loss up to the current time. We illustrate the difference between raw and min-loss PSNR in Fig. 24. Interestingly Sub-DIP NGD differs from full parameter methods in that it does not suffer from noisy optimisation, showing that the approach enjoys excellent stability during the training.

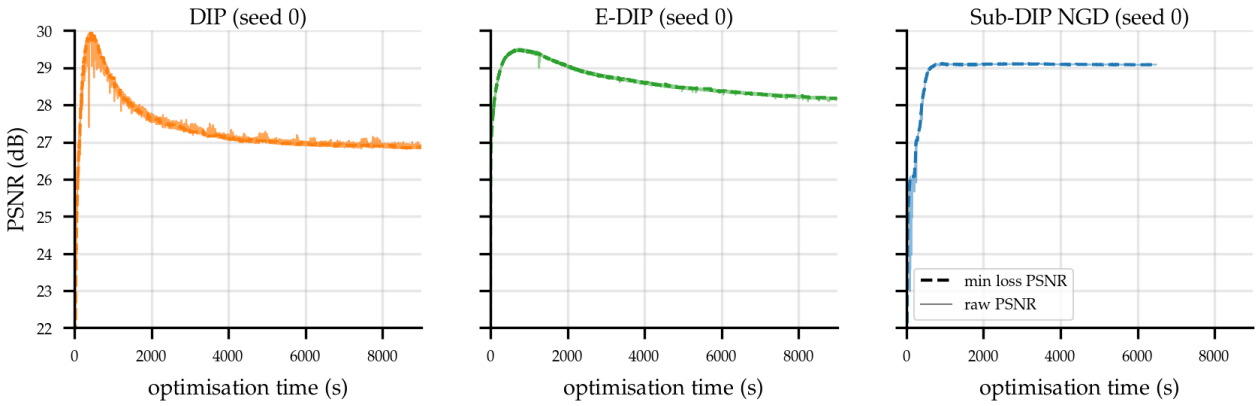

Figure 24: Min-loss and raw PSNR values obtained when reconstructing the Walnut data described in Section 5.2 using seed 0. Min-loss PSNR acts as a smoothed version of raw PSNR, avoiding jumps from optimisation instabilities (left).

C.2 CT forward model operators

Parallel-beam geometry for CartoonSet: Parallel-beam geometries described in Section 5.1 are computed using the ODL library (Adler et al., 2017, github.com/odlgroup/odl) with the "`astra_cuda`" backend

(van Aarle et al., 2015). The angles are adjusted to start with 0 instead of the default $\pi/(2 \cdot \text{num\_angles})$. The number of detector pixels is chosen automatically by ODL such that the discretised image is sufficiently sampled. Since the back-projection operation would approximate the adjoint of the forward projection (due to discretisation differences), we assemble the matrix by calling the forward projection operation for every standard basis vector, $A = A[e_1, e_2, \dots e_{d_x}]$, where $d_x = 128^2$ is the total number of image pixels. The resulting matrix A and its transposed $A^\top$ are used for the forward model and its adjoint, respectively.

Pseudo-2D fan-beam geometry for Walnut: We restrict the 3D cone-beam ASTRA geometry provided with the dataset to a central 2D slice for the first Walnut at the second source position (tubeV2). To do so, we use a sub-sampled set of measurements, which corresponds to a sparse fan-beam-like geometry. From the original 1200 projections (equally distributed over 2π) of size 972×768 we first select the appropriate detector row matching the slice position (which varies for different detector columns and angles due to a tilt in the setup), yielding measurement data of size 1200×768. We then sub-sample in both angle and column dimensions by factors of 20 and 6, respectively, leaving $d_y = 60 \times 128 = 7680$ measurements. As for the operator used in the ablation study on the CartoonSet, we assemble the matrix by calling the forward projection operation for every standard basis vector and use A and $A^\top$ for the forward model and its adjoint, but stored in the sparse matrix form because of the large dimensions. Due to the special selection of detector pixels in order to create the single-slice pseudo-2D geometry, back-projection via ASTRA is not applicable here, so our slightly slower sparse-matrix-based implementation is mandatory.

Fan-beam geometry for Mayo: We simulate the fan-beam geometries using ODL with the "`astra_cuda`" backend. Source and detector radius are chosen to correspond to 700 image pixels, roughly corresponding to the size of a clinical CT gantry (diameter ca. 80 cm). The original image size $(512\,\text{px})^2$ is used for the 100 angle case, but we crop an image of size $(362\,\text{px})^2$ for the 300 angle case, thereby restricting the area to the region inside a circle (outside of which some CT images have invalid values) defining a circular field of view. Like for the CartoonSet, we adjust the angles to start with 0. The number of detector pixels is chosen automatically by ODL such that the discretised image is sufficiently sampled. We assemble matrices A and $A^\top$ as for CartoonSet and Walnut datasets.

C.3 Architectures

The U-Net architecture used for the CartoonSet experiments is shown in Fig. 25. The other tomographic experiments (on the µCT Walnut and Mayo data) use a larger architecture with two more scales and 128 channels in each layer, as shown in Fig. 26. The architecture used for image restoration experiments is instead shown in Fig. 27.

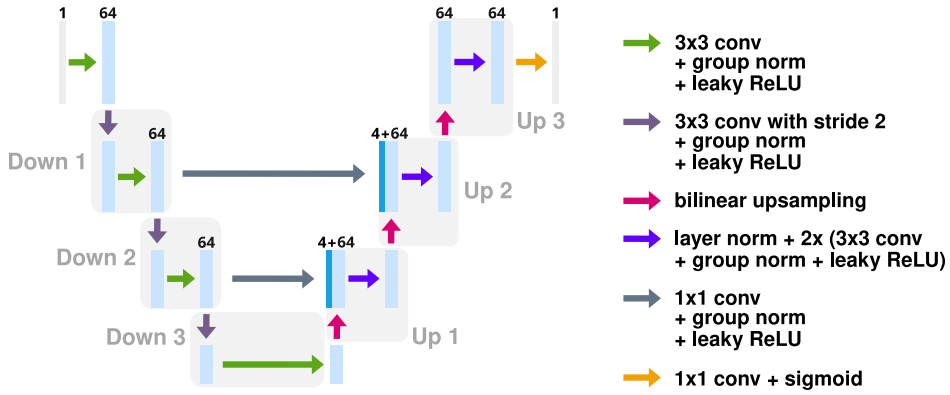

Figure 25: U-Net architecture used with the CartoonSet data. The numbers of channels are indicated above each feature vector.

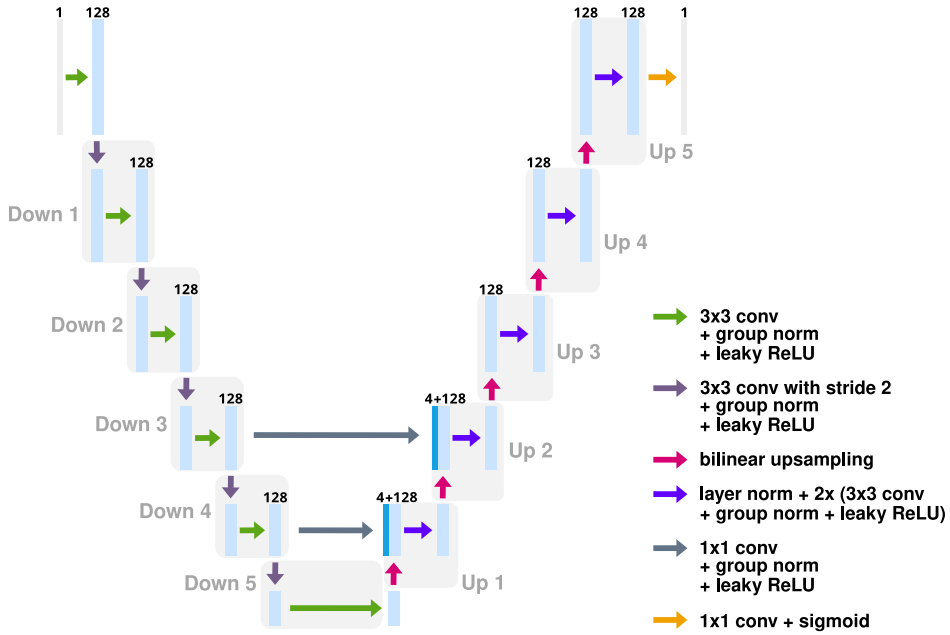

Figure 26: The larger U-Net architecture used with Walnut and Mayo Clinic datasets. The numbers of channels are indicated above each feature vector.

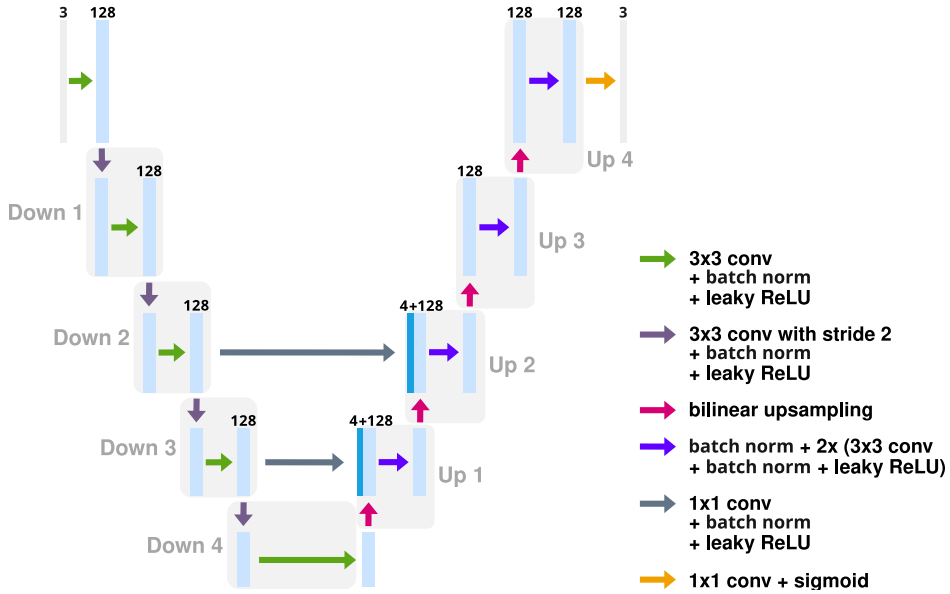

Figure 27: The U-Net architecture used with Set5 dataset of natural images. The numbers of channels are indicated above each feature vector.

C.4 Other experimental hyperparameters

Table 3 reports the parameter λ, used to weigh the contribution of the TV regulariser in (2). For the CartoonSet, λ is selected on a validation set consisting of 5 sample images. Similarly, for the Mayo data (for both 100 and 300 angle settings), λ is selected on a validation set of of 3 sample images. For the Walnut dataset, λ is selected by visual inspection of the reconstruction. Across all tomographic experiments, when optimising DIP or E-DIP with Adam, we keep the learning rate to 10^{-4} and 3×10^{-5}, respectively. Finally, Sub-DIP Adam uses a learning rate of 10^{-3}. For image restoration, $\lambda = 0$ across all settings.

Table 3: λ values used for TV scaling in (2).

	CartoonSet			Walnut	Mayo Clinic	
# angles	45	95	285	120	100	300
λ	3×10^{-5}	3×10^{-5}	3×10^{-5}	6.5×10^{-6}	10^{-4}	10^{-4}