Supplementary Materials

A Posterior samples facilitate calibrated/fair detection

Say that $s \in \{1, 0\}$ denotes the presence or absence of a particular pathology (e.g., brain tumor), and say that we train a soft classifier $c(\cdot)$ on ground-truth images $\boldsymbol{x}$ and calibrate it such that

$$c(\boldsymbol{x}) = \Pr\{s = 1|\boldsymbol{x}\}. \tag{A.1}$$

Now say that we observe a distorted/corrupted/incomplete measurement $\boldsymbol{y} = \mathcal{M}(\boldsymbol{x})$. We would like to infer the probability that the pathology is present given $\boldsymbol{y}$, i.e., compute $\Pr\{s = 1|\boldsymbol{y}\}$. Note that

$$\Pr\{s = 1|\boldsymbol{y}\} = \int \Pr\{s = 1, \boldsymbol{x}|\boldsymbol{y}\} \, \mathrm{d}\boldsymbol{x} = \int \Pr\{s = 1|\boldsymbol{x}, \boldsymbol{y}\} p(\boldsymbol{x}|\boldsymbol{y}) \, \mathrm{d}\boldsymbol{x} \tag{A.2}$$

$$= \int \Pr\{s = 1|\boldsymbol{x}\} p(\boldsymbol{x}|\boldsymbol{y}) \, \mathrm{d}\boldsymbol{x} = \int c(\boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{y}) \, \mathrm{d}\boldsymbol{x} = \mathrm{E}\{c(\boldsymbol{x})|\boldsymbol{y}\} \tag{A.3}$$

$$= \lim_{P\to\infty} \frac{1}{P} \sum_{i=1}^{P} c(\widehat{\boldsymbol{x}}_i) \;\text{ for }\; \widehat{\boldsymbol{x}}_i \sim \text{i.i.d. } p(\boldsymbol{x}|\boldsymbol{y}), \tag{A.4}$$

where the last equality follows from the law of large numbers. So, given access to many independent posterior samples $\{\widehat{\boldsymbol{x}}_i\}$, equation (A.4) says that we can simply plug them into our calibrated classifier $c(\cdot)$ and average the result to compute $\Pr\{s = 1|\boldsymbol{y}\}$. Conversely, if we have access to only the posterior mean $\widehat{\boldsymbol{x}}_{\mathsf{mmse}} = \mathrm{E}\{\boldsymbol{x}|\boldsymbol{y}\}$, then because

$$\Pr\{s = 1|\boldsymbol{y}\} = \mathrm{E}\{c(\boldsymbol{x})|\boldsymbol{y}\} \neq c(\mathrm{E}\{\boldsymbol{x}|\boldsymbol{y}\}) = c(\widehat{\boldsymbol{x}}_{\mathsf{mmse}}) \tag{A.5}$$

for any non-linear $c(\cdot)$, the plug-in probability estimate will be incorrect. In fact, there exists no point estimate $\widehat{\boldsymbol{x}}$ that gives the correct $\Pr\{s = 1|\boldsymbol{y}\}$ for general $c(\cdot)$.

Although above we defined $c(\cdot)$ as a (soft) binary pathology classifier, the same results hold if we define $c(\cdot)$ as a K-ary classifier of any protected attribute, such as race, gender, etc. This implies that, if we have a machine-learning system that has been calibrated to classify fairly on clean ground-truth data $\boldsymbol{x}$, then the use of posterior samples $\{\widehat{\boldsymbol{x}}_i\}$ enables it to classify fairly on distorted/corrupted/incomplete measurements $\boldsymbol{y} = \mathcal{M}(\boldsymbol{x})$, whereas the use of generic point-estimates $\widehat{\boldsymbol{x}}$ does not.

B Proof of Proposition 3.1

Here we prove Proposition 3.1. To begin, for an N-pixel image, we rewrite (8)-(9) as

$$\mathcal{L}_{1,P}(\boldsymbol{\theta}) = \textstyle\sum_{j=1}^{N} \mathrm{E}_{\mathsf{y}} \left\{ \mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}} \left\{ |x_j - \frac{1}{P} \sum_{i=1}^{P} \widehat{x}_{ij}| \, \middle| \boldsymbol{y} \right\} \right\} \tag{B.1}$$

$$\mathcal{L}_{\mathsf{SD},P}(\boldsymbol{\theta}) = \textstyle\sum_{j=1}^{N} \mathrm{E}_{\mathsf{y}} \left\{ \mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}} \left\{ \frac{\gamma_P}{P} \sum_{i=1}^{P} |\widehat{x}_{ij} - \frac{1}{P} \sum_{k=1}^{P} \widehat{x}_{kj}| \, \middle| \boldsymbol{y} \right\} \right\}, \tag{B.2}$$

where $x_j \triangleq [\boldsymbol{x}]_j$, $\widehat{x}_{ij} \triangleq [\widehat{\boldsymbol{x}}_i]_j$, and

$$\gamma_P \triangleq \sqrt{\tfrac{\pi P}{2(P-1)}}. \tag{B.3}$$

To simplify the notation in the sequel, we will consider an arbitrary fixed value of j and use the abbreviations

$$x_j \to X, \qquad\qquad \widehat{x}_{ij} \to \widehat{X}_i. \tag{B.4}$$

Recall that $\boldsymbol{x}$ and $\{\widehat{\boldsymbol{x}}_i\}$ are mutually independent when conditioned on $\boldsymbol{y}$ because the code vectors $\{\boldsymbol{z}_i\}$ are generated independently of both $\boldsymbol{x}$ and $\boldsymbol{y}$. In the context of Proposition 3.1, we also assume that the vector elements x_j and $\widehat{x}_{ij}$ are independent Gaussian when conditioned on $\boldsymbol{y}$. This implies that we can make the notational shift

$$p_{\mathsf{x}|\mathsf{y}}(x_j|\boldsymbol{y}) \to \mathcal{N}(X; \mu_0, \sigma_0^2), \qquad\qquad p_{\widehat{\mathsf{x}}|\mathsf{y}}(\widehat{x}_{ij}|\boldsymbol{y}) \to \mathcal{N}(\widehat{X}_i; \mu, \sigma^2), \tag{B.5}$$

where X and $\{\widehat{X}_i\}$ are mutually independent. With this simplified notation, we note that $[\widehat{\boldsymbol{x}}_{\mathsf{mmse}}]_j \to \mu_0$, and that mode collapse corresponds to $\sigma = 0$.

Furthermore, if $\boldsymbol{\theta}$ can completely control (μ, σ), then (12) can be rewritten as

$$(\mu_*, \sigma_*) = \arg\min_{\mu,\sigma} \left\{ \mathcal{L}_{1,P}(\mu, \sigma) - \beta_{\mathsf{SD}} \mathcal{L}_{\mathsf{SD},P}(\mu, \sigma) \right\} \;\Rightarrow\; \begin{cases} \mu_* = \mu_0 \\ \sigma_* = \sigma_0 \end{cases} \tag{B.6}$$

with

$$\mathcal{L}_{1,P}(\mu, \sigma) = \mathrm{E}_{X, \widehat{X}_1, \ldots \widehat{X}_P} \{ |X - \tfrac{1}{P} \textstyle\sum_{i=1}^P \widehat{X}_i| \} \tag{B.7}$$

$$\mathcal{L}_{\mathsf{SD},P}(\mu, \sigma) = \mathrm{E}_{\widehat{X}_1, \ldots \widehat{X}_P} \{ \tfrac{\gamma_P}{P} \textstyle\sum_{i=1}^P |\widehat{X}_i - \tfrac{1}{P} \sum_{k=1}^P \widehat{X}_k| \}. \tag{B.8}$$

Although σ_* must be positive, it turns out that we do not need to enforce this in the optimization (B.6) because it will arise naturally.

To further analyze (B.7) and (B.8), we define

$$\widehat{\mu} \triangleq \tfrac{1}{P} \textstyle\sum_{i=1}^P \widehat{X}_i \tag{B.9}$$

$$\widehat{\sigma} \triangleq \tfrac{\gamma_P}{P} \textstyle\sum_{i=1}^P |\widehat{X}_i - \widehat{\mu}|. \tag{B.10}$$

The quantity $\widehat{\mu}$ can be recognized as the unbiased estimate of the mean μ of $\widehat{X}_i$, and we now show that $\widehat{\sigma}$ is an unbiased estimate of the SD σ of $\widehat{X}_i$ in the case that $\widehat{X}_i$ is Gaussian. To see this, first observe that the i.i.d. $\mathcal{N}(\mu, \sigma^2)$ property of $\{\widehat{X}_i\}$ implies that $\widehat{X}_i - \widehat{\mu} = (1 - \frac{1}{P})\widehat{X}_i - \frac{1}{P} \sum_{k \neq i} \widehat{X}_k$ is Gaussian with mean zero and variance $(1 - \frac{1}{P})^2 \sigma^2 + \frac{P-1}{P^2} \sigma^2 = \frac{P-1}{P} \sigma^2$. The variable $|\widehat{X}_i - \widehat{\mu}|$ is thus half-normal distributed with mean $\sqrt{\frac{2(P-1)}{\pi P} \sigma^2}$ [48]. Because $\{\widehat{X}_i\}$ are i.i.d., the variable $\frac{1}{P} \sum_{i=1}^P |\widehat{X}_i - \widehat{\mu}|$ has the same mean as $|\widehat{X}_i - \widehat{\mu}|$. Finally, multiplying $\frac{1}{P} \sum_{i=1}^P |\widehat{X}_i - \widehat{\mu}|$ by γ_P yields $\widehat{\sigma}$ from (B.10), and multiplying its mean using the expression for γ_P from (B.3) implies

$$\mathrm{E}\{\widehat{\sigma}\} = \sigma, \tag{B.11}$$

and so $\widehat{\sigma}$ is an unbiased estimator of σ, the SD of $\widehat{X}_i$.

With the above definitions of $\widehat{\mu}$ and $\widehat{\sigma}$, the optimization cost in (B.6) can be written as

$$\mathcal{L}_{1,P}(\mu, \sigma) - \beta_{\mathsf{SD}} \mathcal{L}_{\mathsf{SD},P}(\mu, \sigma) = \mathrm{E}_{X, \widehat{X}_1, \ldots \widehat{X}_P} \left\{ |X - \widehat{\mu}| \right\} - \beta_{\mathsf{SD}} \, \mathrm{E}_{\widehat{X}_1, \ldots \widehat{X}_P} \left\{ \widehat{\sigma} \right\} \tag{B.12}$$

$$= \mathrm{E}_{X, \widehat{X}_1, \ldots \widehat{X}_P} \left\{ |X - \widehat{\mu}| \right\} - \beta_{\mathsf{SD}} \sigma, \tag{B.13}$$

where in the last step we exploited the unbiased property of $\widehat{\sigma}$. To proceed further, we note that the i.i.d. Gaussian property of $\{\widehat{X}_i\}$ implies $\widehat{\mu} \sim \mathcal{N}(\mu, \sigma^2/P)$, after which the mutual independence of $\{\widehat{X}_i\}$ and X yields

$$X - \widehat{\mu} \sim \mathcal{N}(\mu_0 - \mu, \sigma_0^2 + \sigma^2/P). \tag{B.14}$$

Taking the absolute value of a Gaussian random yields a folded-normal random variable [48]. Using the mean and variance in (B.14), the expressions in [48] yield

$$\mathrm{E}_{X, \widehat{X}_1, \ldots \widehat{X}_P} \left\{ |X - \widehat{\mu}| \right\} = \sqrt{\frac{2(\sigma_0^2 + \sigma^2/P)}{\pi}} \exp\Big(- \frac{(\mu_0 - \mu)^2}{2(\sigma_0^2 + \sigma^2/P)} \Big) + (\mu_0 - \mu) \operatorname{erf}\Big(\frac{\mu_0 - \mu}{\sqrt{2(\sigma_0^2 + \sigma^2/P)}} \Big). \tag{B.15}$$

Thus the optimization cost (B.13) can be written as

$$J(\mu, \sigma) = \sqrt{\frac{2(\sigma_0^2 + \sigma^2/P)}{\pi}} \exp\Big(- \frac{(\mu - \mu_0)^2}{2(\sigma_0^2 + \sigma^2/P)} \Big) + (\mu - \mu_0) \operatorname{erf}\Big(\frac{\mu - \mu_0}{\sqrt{2(\sigma_0^2 + \sigma^2/P)}} \Big) - \beta_{\mathsf{SD}} \sigma. \tag{B.16}$$

Since $J(\cdot, \cdot)$ is convex, the minimizer $(\mu_*, \sigma_*) = \arg\min_{\mu,\sigma} J(\mu, \sigma)$ satisfies $\nabla J(\mu_*, \sigma_*) = (0, 0)$. To streamline the derivation, we define

$$c \triangleq \sqrt{2(\sigma_0^2 + \sigma^2/P)/\pi}, \qquad\qquad s \triangleq \sqrt{\sigma_0^2 + \sigma^2/P} \tag{B.17}$$

so that

$$J(\mu,\sigma) = c\exp\Big(-\frac{(\mu-\mu_0)^2}{2s^2}\Big) + (\mu-\mu_0)\operatorname{erf}\Big(\frac{\mu-\mu_0}{\sqrt{2s^2}}\Big) - \beta_{\mathsf{SD}}\sigma. \tag{B.18}$$

Because c and s are invariant to μ, we get

$$\frac{\partial J(\mu,\sigma)}{\partial\mu} = -c\exp\Big(-\frac{(\mu-\mu_0)^2}{2s^2}\Big)\frac{\mu-\mu_0}{s^2} + \operatorname{erf}\Big(\frac{\mu-\mu_0}{\sqrt{2s^2}}\Big) + (\mu-\mu_0)\frac{2}{\sqrt{\pi}}\exp\Big(-\frac{(\mu-\mu_0)^2}{2s^2}\Big), \tag{B.19}$$

which equals zero if and only if $\mu = \mu_0$. Thus we have determined that $\mu_* = \mu_0$. Plugging $\mu_* = \mu_0$ into (B.16), we find

$$J(\mu_*,\sigma) = \sqrt{2(\sigma_0^2+\sigma^2/P)/\pi} - \beta_{\mathsf{SD}}\sigma. \tag{B.20}$$

Taking the derivative with respect to σ, we get

$$\frac{\partial J(\mu_*,\sigma)}{\partial\sigma} = \sqrt{\frac{2}{\pi P(P\sigma_0^2/\sigma^2+1)}} - \beta_{\mathsf{SD}} \tag{B.21}$$

$$= \sqrt{\frac{2}{\pi P(P\sigma_0^2/\sigma^2+1)}} - \sqrt{\frac{2}{\pi P(P+1)}}, \tag{B.22}$$

where in the last step we applied the value of β_{SD} from (11). It can now be seen that $\frac{\partial J(\mu_*,\sigma)}{\partial\sigma} = 0$ if and only if $\sigma = \sigma_0$, which implies that $\sigma_* = \sigma_0$. Thus we have established (B.6), which completes the proof of Proposition 3.1.

C Derivation of Proposition 3.2

Here we prove Proposition 3.2. To start, we establish some notation and conditional-mean properties:

$$\begin{array}{rll}
\widehat{\boldsymbol{x}}_{\mathsf{mmse}} \triangleq \mathrm{E}_{\mathsf{x}|\mathsf{y}}\{\boldsymbol{x}|\boldsymbol{y}\} & & \\
\boldsymbol{e}_{\mathsf{mmse}} \triangleq \boldsymbol{x} - \widehat{\boldsymbol{x}}_{\mathsf{mmse}}, & & \mathbf{0} = \mathrm{E}_{\mathsf{x}|\mathsf{y}}\{\boldsymbol{e}_{\mathsf{mmse}}|\boldsymbol{y}\} \\
\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta}) \triangleq G_{\boldsymbol{\theta}}(\boldsymbol{z}_i,\boldsymbol{y}), & & \overline{\boldsymbol{x}}(\boldsymbol{\theta}) \triangleq \mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})|\boldsymbol{y}\} \\
\widehat{\boldsymbol{x}}_{(P)}(\boldsymbol{\theta}) \triangleq \frac{1}{P}\sum_{i=1}^P \widehat{\boldsymbol{x}}_i(\boldsymbol{\theta}), & & \overline{\boldsymbol{x}}(\boldsymbol{\theta}) = \mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{\widehat{\boldsymbol{x}}_{(P)}(\boldsymbol{\theta})|\boldsymbol{y}\} \\
\boldsymbol{d}_i(\boldsymbol{\theta})) \triangleq \widehat{\boldsymbol{x}}_i(\boldsymbol{\theta}) - \overline{\boldsymbol{x}}(\boldsymbol{\theta}), & & \mathbf{0} = \mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\boldsymbol{d}_i(\boldsymbol{\theta})|\boldsymbol{y}\}\ \forall\boldsymbol{\theta} \\
\boldsymbol{d}_{(P)}(\boldsymbol{\theta}) \triangleq \frac{1}{P}\sum_{i=1}^P \boldsymbol{d}_i(\boldsymbol{\theta}), & & \mathbf{0} = \mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{\boldsymbol{d}_{(P)}(\boldsymbol{\theta})|\boldsymbol{y}\}\ \forall\boldsymbol{\theta}
\end{array} \tag{C.1}$$

Our first step is to write (14) as

$$\mathcal{L}_{2,P}(\boldsymbol{\theta}) = \mathrm{E}_{\mathsf{y}}\big\{\,\mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}\big\}. \tag{C.2}$$

Leveraging the fact that $\widehat{\boldsymbol{x}}_{\mathsf{mmse}}$ and $\overline{\boldsymbol{x}}(\boldsymbol{\theta})$ are deterministic given $\boldsymbol{y}$, we write the inner term in (C.2) as

$$\begin{aligned}
&\mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} \\
&= \mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{\|\widehat{\boldsymbol{x}}_{\mathsf{mmse}}+\boldsymbol{e}_{\mathsf{mmse}}-\overline{\boldsymbol{x}}(\boldsymbol{\theta})-\boldsymbol{d}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} && \text{(C.3)} \\
&= \mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{\|\widehat{\boldsymbol{x}}_{\mathsf{mmse}}-\overline{\boldsymbol{x}}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} \\
&\quad + 2\operatorname{Re}\mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{(\widehat{\boldsymbol{x}}_{\mathsf{mmse}}-\overline{\boldsymbol{x}}(\boldsymbol{\theta}))^{\mathsf{H}}(\boldsymbol{e}_{\mathsf{mmse}}-\boldsymbol{d}_{(P)}(\boldsymbol{\theta}))|\boldsymbol{y}\} \\
&\quad + \mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{\|\boldsymbol{e}_{\mathsf{mmse}}-\boldsymbol{d}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} && \text{(C.4)} \\
&= \|\widehat{\boldsymbol{x}}_{\mathsf{mmse}}-\overline{\boldsymbol{x}}(\boldsymbol{\theta})\|_2^2 + 2\operatorname{Re}\big[(\widehat{\boldsymbol{x}}_{\mathsf{mmse}}-\overline{\boldsymbol{x}}(\boldsymbol{\theta}))^{\mathsf{H}}\underbrace{\mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{(\boldsymbol{e}_{\mathsf{mmse}}-\boldsymbol{d}_{(P)}(\boldsymbol{\theta}))|\boldsymbol{y}\}}_{=\mathbf{0}}\big] \\
&\quad + \mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{\|\boldsymbol{e}_{\mathsf{mmse}}-\boldsymbol{d}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} && \text{(C.5)} \\
&= \|\widehat{\boldsymbol{x}}_{\mathsf{mmse}}-\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})|\boldsymbol{y}\}\|_2^2 + \mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_\mathsf{P}|\mathsf{y}}\{\|\boldsymbol{e}_{\mathsf{mmse}}-\boldsymbol{d}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}. && \text{(C.6)}
\end{aligned}$$

where in (C.5) we used the fact that $\boldsymbol{d}_{(P)}$ and $\boldsymbol{e}_{\mathsf{mmse}}$ are both zero-mean when conditioned on $\boldsymbol{y}$. We now leverage the fact that $\{\boldsymbol{z}_i\}$ are independent of $\boldsymbol{x}$ and $\boldsymbol{y}$ to write

$$\begin{aligned}
&\mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|\boldsymbol{e}_{\mathsf{mmse}}-\boldsymbol{d}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}\\
&=\mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|\boldsymbol{e}_{\mathsf{mmse}}\|_2^2|\boldsymbol{y}\}+2\,\mathrm{Re}\,\mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\boldsymbol{e}_{\mathsf{mmse}}^{\mathsf{H}}\boldsymbol{d}_{(P)}(\boldsymbol{\theta})|\boldsymbol{y}\}+\mathrm{E}_{\mathsf{x},\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|\boldsymbol{d}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} \qquad \text{(C.7)}\\
&=\mathrm{E}_{\mathsf{x}|\mathsf{y}}\{\|\boldsymbol{e}_{\mathsf{mmse}}\|_2^2|\boldsymbol{y}\}+2\,\mathrm{Re}\,\big[\underbrace{\mathrm{E}_{\mathsf{x}|\mathsf{y}}\{\boldsymbol{e}_{\mathsf{mmse}}|\boldsymbol{y}\}}_{=\mathbf{0}}^{\mathsf{H}}\underbrace{\mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\boldsymbol{d}_{(P)}(\boldsymbol{\theta})|\boldsymbol{y}\}}_{=\mathbf{0}}\big]+\mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|\boldsymbol{d}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}. \qquad \text{(C.8)}
\end{aligned}$$

Finally, we can leverage the fact that $\{\boldsymbol{z}_i\}$ are i.i.d. to write

$$\begin{aligned}
\mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|\boldsymbol{d}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}&=\mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|\tfrac{1}{P}\textstyle\sum_{i=1}^P\boldsymbol{d}_i(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} && \text{(C.9)}\\
&=\tfrac{1}{P^2}\textstyle\sum_{i=1}^P\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\|\boldsymbol{d}_i(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} && \text{(C.10)}\\
&=\tfrac{1}{P}\,\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\|\boldsymbol{d}_i(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}\text{ for any }i && \text{(C.11)}\\
&=\tfrac{1}{P}\,\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\mathrm{tr}[\boldsymbol{d}_i(\boldsymbol{\theta})\boldsymbol{d}_i(\boldsymbol{\theta})^{\mathsf{H}}]|\boldsymbol{y}\} && \text{(C.12)}\\
&=\tfrac{1}{P}\,\mathrm{tr}\,\big[\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\boldsymbol{d}_i(\boldsymbol{\theta})\boldsymbol{d}_i(\boldsymbol{\theta})^{\mathsf{H}}\}|\boldsymbol{y}\}\big] && \text{(C.13)}\\
&=\tfrac{1}{P}\,\mathrm{tr}\,\big[\mathrm{Cov}_{\mathsf{z}_i|\mathsf{y}}\{\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})|\boldsymbol{y}\}\big]. && \text{(C.14)}
\end{aligned}$$

Combining (C.2), (C.6), (C.8), and (C.14), we get the bias-variance decomposition

$$\mathcal{L}_{2,P}(\boldsymbol{\theta})=\mathrm{E}_{\mathsf{y}}\Big\{\|\widehat{\boldsymbol{x}}_{\mathsf{mmse}}-\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})|\boldsymbol{y}\}\|_2^2+\tfrac{1}{P}\,\mathrm{tr}\,\big[\mathrm{Cov}_{\mathsf{z}_i|\mathsf{y}}\{\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})|\boldsymbol{y}\}\big]+\mathrm{E}_{\mathsf{x}|\mathsf{y}}\{\|\boldsymbol{e}_{\mathsf{mmse}}\|_2^2|\boldsymbol{y}\}\Big\}. \qquad \text{(C.15)}$$

We now see that if $\boldsymbol{\theta}$ has complete control over the $\boldsymbol{y}$-conditional mean and covariance of $\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})$, then minimizing (C.15) over $\boldsymbol{\theta}$ will cause

$$\begin{aligned}
\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})|\boldsymbol{y}\}&=\widehat{\boldsymbol{x}}_{\mathsf{mmse}} && \text{(C.16)}\\
\mathrm{Cov}_{\mathsf{z}_i|\mathsf{y}}\{\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})|\boldsymbol{y}\}&=\mathbf{0}, && \text{(C.17)}
\end{aligned}$$

which proves Proposition 3.2.

D Derivation of (19)

To show that the expression for $\mathcal{L}_{\mathsf{var},P}$ in (19) holds, we first rewrite (18) as

$$\mathcal{L}_{\mathsf{var},P}(\boldsymbol{\theta})=\tfrac{1}{P-1}\textstyle\sum_{i=1}^P\mathrm{E}_{\mathsf{y}}\{\mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})-\widehat{\boldsymbol{x}}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} \qquad \text{(D.1)}$$

where the definitions from (C.1) imply

$$\begin{aligned}
&\mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})-\widehat{\boldsymbol{x}}_{(P)}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}\\
&=\mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|\overline{\boldsymbol{x}}(\boldsymbol{\theta})+\boldsymbol{d}_i(\boldsymbol{\theta})-\boldsymbol{d}_{(P)}(\boldsymbol{\theta})-\overline{\boldsymbol{x}}(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} && \text{(D.2)}\\
&=\mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|\boldsymbol{d}_i(\boldsymbol{\theta})-\tfrac{1}{P}\textstyle\sum_{j=1}^P\boldsymbol{d}_j(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} && \text{(D.3)}\\
&=\mathrm{E}_{\mathsf{z}_1,\ldots,\mathsf{z}_P|\mathsf{y}}\{\|(1-\tfrac{1}{P})\boldsymbol{d}_i(\boldsymbol{\theta})-\tfrac{1}{P}\textstyle\sum_{j\neq i}\boldsymbol{d}_j(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} && \text{(D.4)}\\
&=(1-\tfrac{1}{P})^2\,\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\|\boldsymbol{d}_i(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}+\tfrac{P-1}{P^2}\,\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\|\boldsymbol{d}_i(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\} && \text{(D.5)}\\
&=\tfrac{P-1}{P}\,\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\|\boldsymbol{d}_i(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}\text{ for any }i. && \text{(D.6)}
\end{aligned}$$

For (D.5), we leveraged the zero-mean and i.i.d. nature of $\{\boldsymbol{d}_i(\boldsymbol{\theta})\}$ conditioned on $\boldsymbol{y}$. By plugging (D.6) into (D.1), we get

$$\begin{aligned}
\mathcal{L}_{\mathsf{var},P}(\boldsymbol{\theta})&=\tfrac{1}{P}\textstyle\sum_{i=1}^P\mathrm{E}_{\mathsf{y}}\{\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\|\boldsymbol{d}_i(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}\} && \text{(D.7)}\\
&=\mathrm{E}_{\mathsf{y}}\{\mathrm{E}_{\mathsf{z}_i|\mathsf{y}}\{\|\boldsymbol{d}_i(\boldsymbol{\theta})\|_2^2|\boldsymbol{y}\}\}\text{ for any }i && \text{(D.8)}\\
&=\mathrm{E}_{\mathsf{y}}\{\mathrm{tr}[\mathrm{Cov}_{\mathsf{z}_i|\mathsf{y}}\{\widehat{\boldsymbol{x}}_i(\boldsymbol{\theta})|\boldsymbol{y}\}]\}, && \text{(D.9)}
\end{aligned}$$

where (D.8) follows because $\{\boldsymbol{d}_i(\boldsymbol{\theta})\}$ are i.i.d. when conditioned on $\boldsymbol{y}$ and (D.9) follows from manipulations similar to those used for (C.14).

E Proof of Proposition 3.3

Here we prove Proposition 3.3. Recall from (C.1) that $\widehat{\boldsymbol{x}}_{\mathsf{mmse}} \triangleq \mathrm{E}\{\boldsymbol{x}|\boldsymbol{y}\}$ and $\boldsymbol{e}_{\mathsf{mmse}} \triangleq \boldsymbol{x} - \widehat{\boldsymbol{x}}_{\mathsf{mmse}}$. To reduce clutter, we will abbreviate $\boldsymbol{e}_{\mathsf{mmse}}$ by $\boldsymbol{e}$ in this appendix. Also, for true-posterior samples $\widehat{\boldsymbol{x}}_i \sim p_{\mathsf{x}|\mathsf{y}}(\cdot|\boldsymbol{y})$, we define

$$\widehat{\boldsymbol{e}}_i \triangleq \widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_{\mathsf{mmse}}. \tag{E.1}$$

Then using $\widehat{\boldsymbol{x}}_{(P)} \triangleq \frac{1}{P}\sum_{i=1}^P \widehat{\boldsymbol{x}}_i$ and from $\mathcal{E}_P$ from (20), we have

$$\begin{aligned}
\mathcal{E}_P &= \mathrm{E}\{\|\widehat{\boldsymbol{x}}_{(P)} - \boldsymbol{x}\|^2|\boldsymbol{y}\} && \text{(E.2)}\\
&= \mathrm{E}\{\|(\tfrac{1}{P}\textstyle\sum_{i=1}^P \widehat{\boldsymbol{x}}_i) - \boldsymbol{x}\|^2|\boldsymbol{y}\} && \text{(E.3)}\\
&= \mathrm{E}\{\|\tfrac{1}{P}\textstyle\sum_{i=1}^P (\widehat{\boldsymbol{x}}_i - \boldsymbol{x})\|^2|\boldsymbol{y}\} && \text{(E.4)}\\
&= \tfrac{1}{P^2}\mathrm{E}\{\|\textstyle\sum_{i=1}^P (\widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_{\mathsf{mmse}} + \widehat{\boldsymbol{x}}_{\mathsf{mmse}} - \boldsymbol{x})\|^2|\boldsymbol{y}\} && \text{(E.5)}\\
&= \tfrac{1}{P^2}\mathrm{E}\{\|\textstyle\sum_{i=1}^P (\widehat{\boldsymbol{e}}_i - \boldsymbol{e})\|^2|\boldsymbol{y}\} && \text{(E.6)}\\
&= \tfrac{1}{P^2}\mathrm{E}\{\textstyle\sum_{i=1}^P (\widehat{\boldsymbol{e}}_i - \boldsymbol{e})^{\mathsf{H}} \sum_{j=1}^P (\widehat{\boldsymbol{e}}_j - \boldsymbol{e})|\boldsymbol{y}\} && \text{(E.7)}\\
&= \tfrac{1}{P^2}\textstyle\sum_{i=1}^P \sum_{j=1}^P \mathrm{E}\{(\widehat{\boldsymbol{e}}_i - \boldsymbol{e})^{\mathsf{H}}(\widehat{\boldsymbol{e}}_j - \boldsymbol{e})|\boldsymbol{y}\} && \text{(E.8)}\\
&= \tfrac{1}{P^2}\textstyle\sum_{i=1}^P \mathrm{E}\{(\widehat{\boldsymbol{e}}_i - \boldsymbol{e})^{\mathsf{H}}(\widehat{\boldsymbol{e}}_i - \boldsymbol{e})|\boldsymbol{y}\} + \tfrac{1}{P^2}\textstyle\sum_{i=1}^P \sum_{j\neq i} \mathrm{E}\{(\widehat{\boldsymbol{e}}_i - \boldsymbol{e})^{\mathsf{H}}(\widehat{\boldsymbol{e}}_j - \boldsymbol{e})|\boldsymbol{y}\} && \text{(E.9)}\\
&= \tfrac{1}{P^2}\textstyle\sum_{i=1}^P \big[\mathrm{E}\{\|\widehat{\boldsymbol{e}}_i\|^2|\boldsymbol{y}\} - 2\,\mathrm{Re}\,\mathrm{E}\{\widehat{\boldsymbol{e}}_i^{\mathsf{H}}\boldsymbol{e}|\boldsymbol{y}\} + \mathrm{E}\{\|\boldsymbol{e}\|^2|\boldsymbol{y}\}\big]\\
&\quad + \tfrac{1}{P^2}\textstyle\sum_{i=1}^P \sum_{j\neq i} \mathrm{Re}\,\big[\mathrm{E}\{\widehat{\boldsymbol{e}}_i^{\mathsf{H}}\widehat{\boldsymbol{e}}_j|\boldsymbol{y}\} - \mathrm{E}\{\widehat{\boldsymbol{e}}_i^{\mathsf{H}}\boldsymbol{e}|\boldsymbol{y}\} - \mathrm{E}\{\boldsymbol{e}^{\mathsf{H}}\widehat{\boldsymbol{e}}_j|\boldsymbol{y}\} + \mathrm{E}\{\|\boldsymbol{e}\|^2|\boldsymbol{y}\}\big] && \text{(E.10)}\\
&= \tfrac{1}{P^2}\textstyle\sum_{i=1}^P \mathrm{E}\{\|\widehat{\boldsymbol{e}}_i\|^2|\boldsymbol{y}\} + \tfrac{1}{P}\mathrm{E}\{\|\boldsymbol{e}\|^2|\boldsymbol{y}\} + \tfrac{P(P-1)}{P^2}\mathrm{E}\{\|\boldsymbol{e}\|^2|\boldsymbol{y}\}, && \text{(E.11)}
\end{aligned}$$

where certain terms vanished because the i.i.d. and zero-mean properties of $\{\boldsymbol{e}, \widehat{\boldsymbol{e}}_1, \ldots, \widehat{\boldsymbol{e}}_P\}$ imply

$$\mathrm{E}\{\widehat{\boldsymbol{e}}_i^{\mathsf{H}}\widehat{\boldsymbol{e}}_j|\boldsymbol{y}\} = \mathrm{E}\{\widehat{\boldsymbol{e}}_i|\boldsymbol{y}\}^{\mathsf{H}}\mathrm{E}\{\widehat{\boldsymbol{e}}_j|\boldsymbol{y}\} = 0 \tag{E.12}$$

$$\mathrm{E}\{\widehat{\boldsymbol{e}}_i^{\mathsf{H}}\boldsymbol{e}|\boldsymbol{y}\} = \mathrm{E}\{\widehat{\boldsymbol{e}}_i|\boldsymbol{y}\}^{\mathsf{H}}\mathrm{E}\{\boldsymbol{e}|\boldsymbol{y}\} = 0 \tag{E.13}$$

$$\mathrm{E}\{\boldsymbol{e}^{\mathsf{H}}\widehat{\boldsymbol{e}}_j|\boldsymbol{y}\} = \mathrm{E}\{\boldsymbol{e}|\boldsymbol{y}\}^{\mathsf{H}}\mathrm{E}\{\widehat{\boldsymbol{e}}_j|\boldsymbol{y}\} = 0. \tag{E.14}$$

Finally, note that $\mathrm{E}\{\|\boldsymbol{e}\|^2|\boldsymbol{y}\} = \mathcal{E}_{\mathsf{mmse}}$ from (C.1). Furthermore, because $\{\boldsymbol{x}, \widehat{\boldsymbol{x}}_1, \ldots, \widehat{\boldsymbol{x}}_P\}$ are independent samples of $p_{\mathsf{x}|\mathsf{y}}(\cdot|\boldsymbol{y})$ under the assumptions of Proposition 3.3, we have $\mathrm{E}\{\|\boldsymbol{e}\|^2|\boldsymbol{y}\} = \mathrm{E}\{\|\widehat{\boldsymbol{e}}_i\|^2|\boldsymbol{y}\}$ and so (E.11) becomes

$$\mathcal{E}_P = \frac{1}{P^2}\sum_{i=1}^P \mathcal{E}_{\mathsf{mmse}} + \frac{1}{P}\mathcal{E}_{\mathsf{mmse}} + \frac{P(P-1)}{P^2}\mathcal{E}_{\mathsf{mmse}} = \frac{P+1}{P}\mathcal{E}_{\mathsf{mmse}}. \tag{E.15}$$

This result holds for any $P \geq 1$, which implies the ratio

$$\frac{\mathcal{E}_1}{\mathcal{E}_P} = \frac{2P}{P+1}. \tag{E.16}$$

F CFID implementation details

With the Gaussian approximation described in Section 4.1, where $p_{\underline{\mathsf{x}}|\underline{\mathsf{y}}}$ and $p_{\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}}$ are approximated by $\mathcal{N}(\boldsymbol{\mu}_{\underline{\mathsf{x}}|\underline{\mathsf{y}}}, \boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}})$ and $\mathcal{N}(\boldsymbol{\mu}_{\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}}, \boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}})$, respectively, the CWD in (24) reduces to

$$\mathrm{CFID} \triangleq \mathrm{E}_{\mathsf{y}}\left\{\|\boldsymbol{\mu}_{\underline{\mathsf{x}}|\underline{\mathsf{y}}} - \boldsymbol{\mu}_{\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}}\|_2^2 + \mathrm{tr}\left[\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}} + \boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}} - 2\big(\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}}^{1/2}\boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}}\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}}^{1/2}\big)^{1/2}\right]\right\}. \tag{F.1}$$

The values in (F.1) are computed using

$$\boldsymbol{\mu}_{\underline{\mathsf{x}}|\underline{\mathsf{y}}} = \boldsymbol{\mu}_{\underline{\mathsf{x}}} + \boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{y}}}\boldsymbol{\Sigma}_{\underline{\mathsf{y}}\underline{\mathsf{y}}}^{-1}(\underline{\boldsymbol{y}} - \boldsymbol{\mu}_{\underline{\mathsf{y}}}) \tag{F.2}$$

$$\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}} = \boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}} - \boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{y}}}\boldsymbol{\Sigma}_{\underline{\mathsf{y}}\underline{\mathsf{y}}}^{-1}\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{y}}}^{\top} \tag{F.3}$$

$$\boldsymbol{\mu}_{\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}} = \boldsymbol{\mu}_{\underline{\widehat{\mathsf{x}}}} + \boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\mathsf{y}}}\boldsymbol{\Sigma}_{\underline{\mathsf{y}}\underline{\mathsf{y}}}^{-1}(\underline{\boldsymbol{y}} - \boldsymbol{\mu}_{\underline{\mathsf{y}}}) \tag{F.4}$$

$$\boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}} = \boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\widehat{\mathsf{x}}}} - \boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\mathsf{y}}}\boldsymbol{\Sigma}_{\underline{\mathsf{y}}\underline{\mathsf{y}}}^{-1}\boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\mathsf{y}}}^{\top}. \tag{F.5}$$

Plugging (F.2)-(F.5) into (F.1), the CFID can be written as [22, Lemma 2]:

$$\begin{aligned}\text{CFID} = \|\boldsymbol{\mu}_{\underline{\mathsf{x}}} - \boldsymbol{\mu}_{\widehat{\underline{\mathsf{x}}}}\|_2^2 + \operatorname{tr}\Big[(\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{y}}} - \boldsymbol{\Sigma}_{\widehat{\underline{\mathsf{x}}}\underline{\mathsf{y}}})\boldsymbol{\Sigma}_{\underline{\mathsf{y}}\underline{\mathsf{y}}}^{-1}(\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{y}}} - \boldsymbol{\Sigma}_{\widehat{\underline{\mathsf{x}}}\underline{\mathsf{y}}})^\top\Big] \\ + \operatorname{tr}\Big[\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}} + \boldsymbol{\Sigma}_{\widehat{\underline{\mathsf{x}}}\widehat{\underline{\mathsf{x}}}|\underline{\mathsf{y}}} - 2\big(\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}}^{1/2}\boldsymbol{\Sigma}_{\widehat{\underline{\mathsf{x}}}\widehat{\underline{\mathsf{x}}}|\underline{\mathsf{y}}}\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}}^{1/2}\big)^{1/2}\Big],\end{aligned} \tag{F.6}$$

where $\boldsymbol{\Sigma}_{\underline{\mathsf{y}}\underline{\mathsf{y}}}^{-1}$ is typically implemented using a pseudo-inverse.

We now detail how the means and covariances in (F.6) are computed. We start with a dataset $\{(\boldsymbol{x}_t, \boldsymbol{y}_t)\}_{t=1}^n$ of truth/measurement pairs. For each $\boldsymbol{y}_t$, we generate a set of P posterior samples $\{\widehat{\boldsymbol{x}}_{ti}\}_{i=1}^P$. We merge these samples with P repetitions of $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ to obtain $\{(\boldsymbol{x}_{ti}, \boldsymbol{y}_{ti}, \widehat{\boldsymbol{x}}_{ti})\}_{i=1}^P$ for $t = 1 \dots n$. These terms are processed by a feature-generating network to yield the feature embeddings $\{(\underline{\boldsymbol{x}}_{ti}, \underline{\boldsymbol{y}}_{ti}, \widehat{\underline{\boldsymbol{x}}}_{ti})\}_{i=1}^P$, which are then packed into matrices $\underline{\boldsymbol{X}}$, $\underline{\boldsymbol{Y}}$, and $\widehat{\underline{\boldsymbol{X}}}$ with Pn rows. We used the VGG-16 feature-generating network [49] for our MRI experiments, since [41] found that it gave results that correlated much better with radiologists' perceptions, while we used the standard Inception-v3 network [50] for our inpainting experiments. The embeddings are then used to compute the sample-mean values

$$\boldsymbol{\mu}_{\underline{\mathsf{x}}} \triangleq \tfrac{1}{Pn}\mathbf{1}^\top\underline{\boldsymbol{X}}, \quad \boldsymbol{\mu}_{\underline{\mathsf{y}}} \triangleq \tfrac{1}{Pn}\mathbf{1}^\top\underline{\boldsymbol{Y}}, \quad \boldsymbol{\mu}_{\widehat{\underline{\mathsf{x}}}} \triangleq \tfrac{1}{Pn}\mathbf{1}^\top\widehat{\underline{\boldsymbol{X}}}. \tag{F.7}$$

We then subtract the sample mean from each row of $\underline{\boldsymbol{X}}$, $\underline{\boldsymbol{Y}}$, and $\widehat{\underline{\boldsymbol{X}}}$ to give the zero-mean embedding matrices $\underline{\boldsymbol{X}}_{\mathsf{zm}} \triangleq \underline{\boldsymbol{X}} - \mathbf{1}\boldsymbol{\mu}_{\underline{\mathsf{x}}}^\top$, $\underline{\boldsymbol{Y}}_{\mathsf{zm}} \triangleq \underline{\boldsymbol{Y}} - \mathbf{1}\boldsymbol{\mu}_{\underline{\mathsf{y}}}^\top$, and $\widehat{\underline{\boldsymbol{X}}}_{\mathsf{zm}} \triangleq \widehat{\underline{\boldsymbol{X}}} - \mathbf{1}\boldsymbol{\mu}_{\widehat{\underline{\mathsf{x}}}}^\top$, which are then used to compute the sample covariance matrices

$$\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}} \triangleq \tfrac{1}{Pn}\underline{\boldsymbol{X}}_{\mathsf{zm}}^\top\underline{\boldsymbol{X}}_{\mathsf{zm}}, \quad \boldsymbol{\Sigma}_{\underline{\mathsf{y}}\underline{\mathsf{y}}} \triangleq \tfrac{1}{Pn}\underline{\boldsymbol{Y}}_{\mathsf{zm}}^\top\underline{\boldsymbol{Y}}_{\mathsf{zm}}, \quad \boldsymbol{\Sigma}_{\widehat{\underline{\mathsf{x}}}\widehat{\underline{\mathsf{x}}}} \triangleq \tfrac{1}{Pn}\widehat{\underline{\boldsymbol{X}}}_{\mathsf{zm}}^\top\widehat{\underline{\boldsymbol{X}}}_{\mathsf{zm}} \tag{F.8a}$$

$$\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{y}}} \triangleq \tfrac{1}{Pn}\underline{\boldsymbol{X}}_{\mathsf{zm}}^\top\underline{\boldsymbol{Y}}_{\mathsf{zm}}, \quad \boldsymbol{\Sigma}_{\widehat{\underline{\mathsf{x}}}\underline{\mathsf{y}}} \triangleq \tfrac{1}{Pn}\widehat{\underline{\boldsymbol{X}}}_{\mathsf{zm}}^\top\underline{\boldsymbol{Y}}_{\mathsf{zm}}. \tag{F.8b}$$

We plug the sample statistics from (F.7)-(F.8) into (F.2)-(F.5), which yields the statistics needed to compute the CFID in (F.6). In [22], the authors use $P = 1$ in all of their experiments. To be consistent with how we evaluated the other metrics, we use $P = 32$ unless otherwise noted.

G MR imaging details

We now give details on magnetic resonance (MR) image recovery. Suppose that the goal is to recover the N-pixel MR image $\boldsymbol{i} \in \mathbb{C}^N$ from the multicoil measurements $\{\boldsymbol{k}_c\}_{c=1}^C$, where [39]

$$\boldsymbol{k}_c = \boldsymbol{M}\boldsymbol{F}\boldsymbol{S}_c\boldsymbol{i} + \boldsymbol{n}_c. \tag{G.1}$$

In (G.1), C refers to the number of coils, $\boldsymbol{k}_c \in \mathbb{C}^M$ are the measurements from the cth coil, $\boldsymbol{M} \in \mathbb{R}^{M\times N}$ is a sub-sampling operator containing rows from $\boldsymbol{I}_N$—the $N \times N$ identity matrix, $\boldsymbol{F} \in \mathbb{C}^{N\times N}$ is the unitary 2D discrete Fourier transform, $\boldsymbol{S}_c \in \mathbb{C}^{N\times N}$ is a diagonal matrix containing the sensitivity map of the cth coil, and $\boldsymbol{n}_c \in \mathbb{C}^M$ is noise. From (G.1), it can be seen that the MR measurements are collected in the spatial Fourier domain, otherwise known as the "k-space." The sensitivity maps $\{\boldsymbol{S}_c\}$ are estimated from $\{\boldsymbol{k}_c\}$ using ESPIRiT [40] (in our case via SigPy [51]), which yields maps with the property $\sum_{c=1}^C \boldsymbol{S}_c^{\mathsf{H}}\boldsymbol{S}_c = \boldsymbol{I}_N$. The ratio $R \triangleq \frac{N}{M}$ is known as the acceleration rate.

There are different ways that one could apply the generative posterior sampling framework to multicoil MR image recovery. One is to configure the generator to produce posterior samples $\widehat{\boldsymbol{i}}$ of the complex image $\boldsymbol{i}$. Another is to configure the generator to produce posterior samples $\widehat{\boldsymbol{x}}$ of the stack $\boldsymbol{x} \triangleq [\boldsymbol{x}_1^\top, \dots, \boldsymbol{x}_C^\top]^\top$ of "coil images" $\boldsymbol{x}_c \triangleq \boldsymbol{S}_c\boldsymbol{i}$ and later coil-combining them to yield a complex image estimate $\widehat{\boldsymbol{i}} \triangleq [\boldsymbol{S}_1^{\mathsf{H}}, \dots, \boldsymbol{S}_C^{\mathsf{H}}]\widehat{\boldsymbol{x}}$. We take the latter approach. Furthermore, rather than feeding our generator with k-space measurements $\boldsymbol{k}_c$, we choose to feed it with aliased coil images $\boldsymbol{y}_c \triangleq \boldsymbol{F}^{\mathsf{H}}\boldsymbol{M}^\top\boldsymbol{k}_c$. Writing (G.1) in terms of the coil images, we obtain

$$\boldsymbol{y}_c = \boldsymbol{F}^{\mathsf{H}}\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{F}\boldsymbol{x}_c + \boldsymbol{w}_c, \tag{G.2}$$

where $\boldsymbol{w}_c \triangleq \boldsymbol{F}^{\mathsf{H}}\boldsymbol{M}^\top\boldsymbol{n}_c$. Then we can stack $\{\boldsymbol{y}_c\}$ and $\{\boldsymbol{w}_c\}$ column-wise into vectors $\boldsymbol{y}$ and $\boldsymbol{w}$, and set $\boldsymbol{A} = \boldsymbol{I}_C \otimes \boldsymbol{F}^{\mathsf{H}}\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{F} \in \mathbb{C}^{NC\times NC}$, to obtain the formulation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$ described in Section 1.

To train our generator, we assume to have access to paired training examples $\{(\boldsymbol{x}_t, \boldsymbol{y}_t)\}$, where $\boldsymbol{x}_t$ is a stack of coil images and $\boldsymbol{y}_t$ is the corresponding stack of k-space coil measurements. The fastMRI multicoil dataset [33] provides $\{(\boldsymbol{x}_t, \boldsymbol{k}_t)\}$, from which we can easily obtain $\{(\boldsymbol{x}_t, \boldsymbol{y}_t)\}$.

H Data-consistency

In this section, we describe a data-consistency procedure that can be optionally used when our cGAN is used to solve a *linear* inverse problem, i.e., to recover $\boldsymbol{x}$ from $\boldsymbol{y}$ under a model of the form

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}, \tag{H.1}$$

where $\boldsymbol{A}$ is a known linear operator and $\boldsymbol{w}$ is unknown noise. The motivation is that, in some applications, such as medical imaging or inpainting, the end user may feel comfortable knowing that the generated samples $\{\widehat{\boldsymbol{x}}_i\}$ are consistent with the measurements $\boldsymbol{y}$ in that

$$\boldsymbol{y} = \boldsymbol{A}\widehat{\boldsymbol{x}}_i. \tag{H.2}$$

When (H.2) holds, $\boldsymbol{A}^+\boldsymbol{y} = \boldsymbol{A}^+\boldsymbol{A}\widehat{\boldsymbol{x}}_i$ must also hold, where $(\cdot)^+$ denotes the pseudo-inverse. The quantity $\boldsymbol{A}^+\boldsymbol{A}$ can be recognized as the orthogonal projection matrix associated with the row space of $\boldsymbol{A}$. So, (H.2) requires the component of $\widehat{\boldsymbol{x}}_i$ in the row space of $\boldsymbol{A}$ to equal $\boldsymbol{A}^+\boldsymbol{y}$, while placing no constraints on the component of $\widehat{\boldsymbol{x}}_i$ in the nullspace of $\boldsymbol{A}$. This suggests the following data-consistency procedure:

$$\widehat{\boldsymbol{x}}_i = (\boldsymbol{I} - \boldsymbol{A}^+\boldsymbol{A})\widehat{\boldsymbol{x}}_i^{\mathsf{raw}} + \boldsymbol{A}^+\boldsymbol{y}. \tag{H.3}$$

where $\widehat{\boldsymbol{x}}_i^{\mathsf{raw}}$ is the raw generator output. We note that a version of this idea for point estimation was proposed in [3].

The data-consistency procedure (H.3) ensures that any generative method will generate only the component of $\boldsymbol{x}$ that lies in the nullspace of $\boldsymbol{A}$. Consequently, (H.3) is admissible only when $\boldsymbol{A}$ has a non-trivial nullspace. Also, because no attempt is made to remove the noise $\boldsymbol{w}$ in $\boldsymbol{y}$, this approach is recommended only for low-noise applications. For high-noise applications, an extension based on the dual-decomposition approach [52] would be more appropriate, but we leave this to future work.

When applying (H.3) to the MRI formulation in Appendix G, we note that $\boldsymbol{A} = \boldsymbol{I}_C \otimes \boldsymbol{F}^{\mathsf{H}}\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{F}$ is an orthogonal projection matrix, and so $\boldsymbol{I} - \boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{A} = \boldsymbol{I} \otimes \boldsymbol{F}^{\mathsf{H}}(\boldsymbol{I} - \boldsymbol{M}^\top\boldsymbol{M})\boldsymbol{F}$.

I Implementation details

The code for our model can be found here: `https://github.com/matt-bendel/rcGAN`.

I.1 MRI

I.1.1 cGAN training

At each training iteration, our cGAN's generator takes in n_{batch} measurement samples $\boldsymbol{y}_t$ and P_{train} code vectors for every $\boldsymbol{y}_t$, and performs an optimization step on the loss

$$\mathcal{L}_{\mathsf{G}}(\boldsymbol{\theta}) \triangleq \beta_{\mathsf{adv}}\mathcal{L}_{\mathsf{adv}}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \mathcal{L}_{1,P_{\mathsf{train}}}(\boldsymbol{\theta}) - \beta_{\mathsf{SD}}\mathcal{L}_{\mathsf{SD},P_{\mathsf{train}}}(\boldsymbol{\theta}), \tag{I.1}$$

where by default we use $\beta_{\mathsf{adv}} =$ 1e-5, $n_{\mathsf{batch}} = 36$, $P_{\mathsf{train}} = 2$, and update β_{SD} via (23) using $P_{\mathsf{val}} = 8$. Then, using the $P_{\mathsf{train}}n_{\mathsf{batch}}$ generator outputs, our cGAN's discriminator performs an optimization step on the loss

$$\mathcal{L}_{\mathsf{D}}(\boldsymbol{\phi}) = -\mathcal{L}_{\mathsf{adv}}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \alpha_1\mathcal{L}_{\mathsf{gp}}(\boldsymbol{\phi}) + \alpha_2\mathcal{L}_{\mathsf{drift}}(\boldsymbol{\phi}), \tag{I.2}$$

with gradient penalty $\mathcal{L}_{\mathsf{gp}}$ from [24]. As per [29], $\mathcal{L}_{\mathsf{drift}}$ is a drift penalty, $\alpha_1 = 10$, $\alpha_2 = 0.001$, and one discriminator update was used per generator update. The models were trained for 100 epochs using the Adam optimizer [53] with a learning rate of 1e-3, $\beta_1 = 0$, and $\beta_2 = 0.99$, as in [29]. Running PyTorch on a server with 4 Tesla A100 GPUs, each with 82 GB of memory, the training of an MRI cGAN took approximately 1 day.

Adler and Öktem's cGAN [8] uses generator loss $\beta_{\mathsf{adv}}\mathcal{L}_{\mathsf{adv}}^{\mathsf{adler}}(\boldsymbol{\theta}, \boldsymbol{\phi})$, where $\mathcal{L}_{\mathsf{adv}}^{\mathsf{adler}}(\boldsymbol{\theta}, \boldsymbol{\phi})$ was described in (5), and discriminator loss $-\mathcal{L}_{\mathsf{adv}}^{\mathsf{adler}}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \alpha_1\mathcal{L}_{\mathsf{gp}}(\boldsymbol{\phi}) + \alpha_2\mathcal{L}_{\mathsf{drift}}(\boldsymbol{\phi})$ with the values of $\alpha_1 =$10, $\alpha_2 =$0.001, and $\beta_{\mathsf{adv}} =$1, as in the original paper.

Ohayon et al.'s cGAN [23] uses generator loss $\beta_{\mathsf{adv}}\mathcal{L}_{\mathsf{adv}}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \mathcal{L}_{2,P_{\mathsf{train}}}(\boldsymbol{\theta})$, where $\mathcal{L}_{2,P_{\mathsf{train}}}(\boldsymbol{\theta})$ was described in (14), and discriminator loss $-\mathcal{L}_{\mathsf{adv}}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \alpha_1\mathcal{L}_{\mathsf{gp}}(\boldsymbol{\phi}) + \alpha_2\mathcal{L}_{\mathsf{drift}}(\boldsymbol{\phi})$ with the values α_1 =10, α_2 =0.001, and β_{adv} =1e-5. We modify β_{adv} to re-balance the loss due to an increased magnitude of our discriminator's outputs.

All three cGANs used the same generator and discriminator architectures (detailed below), except that Adler and Öktem's discriminator used extra input channels to facilitate the 3-input loss $\mathcal{L}^{\mathsf{adler}}_{\mathsf{adv}}(\boldsymbol{\theta}, \boldsymbol{\phi})$ from (5).

I.1.2 cGAN generator architecture

For our MRI experiments, we take inspiration from the UNet architecture from [36], using it as the basis for the cGAN generators. The primary input $\boldsymbol{y}$ is concatenated with the code vector $\boldsymbol{z}$ and fed through the UNet. The network consists of 4 pooling layers with 128 initial channels. However, instead of pooling, we opt to use convolutions with kernels of size 3×3, "same" padding, and a stride of 2 when downsampling. Likewise, we upsample using transpose convolutions, again with kernels of size 3×3, "same" padding, and a stride of 2. All other convolutions utilize kernels of size 3×3, "same" padding, and a stride of 1.

Within each encoder and decoder layer we include a residual block, the architecture of which can be found in [8]. We use instance-norm for all normalization layers and parametric ReLUs as our activation functions, in which the network learns the optimal "negative slope." Finally, we include 5 residual blocks at the base of the UNet, in between the encoder and decoder. This is done in an effort to artificially increase the depth of the network and is inspired by [54]. Our generator has $86\,734\,334$ trainable parameters.

I.1.3 cGAN discriminator architecture

Our discriminator is a standard CNN with 6 layers and 1 fully-connected layer. In the first 3 layers, we use convolutions with kernels of size 3×3, "same" padding. We reduce spatial resolution with average pooling, using 2×2 kernels with a stride of 2. We use batch-norm as our normalization layer and leaky ReLUs with a "negative-slope" of 0.2 as our activation functions. The network outputs an estimated Wasserstein score for the whole image.

I.1.4 E2E-VarNet

For the Sriram et al.'s E2E-VarNet [37], we use the same training procedure and hyperparameters outlined in [19] other than replacing the sampling pattern with the GRO undersampling mask. As in [19], we use the SENSE-based coil-combined image as the ground truth instead of the RSS image.

I.1.5 Langevin approach

For Jalal et al.'s MRI approach [19], we do not modify the original implementation from [38] other than replacing the default sampling pattern with the GRO undersampling mask. We generated 32 samples for 72 different test images using a batch-size of 4, which took roughly 6 days. These samples were generated on a server with 4 NVIDIA V100 GPUs, each with 32 GB of memory. We used 4 samples per batch (and recorded the time to generate 4 samples in Table 1) because the code from [38] is written to generate one sample per GPU.

I.2 Inpainting

I.2.1 Our cGAN

For our generator and discriminator, we use the CoModGAN networks from [9]. Unlike CoModGAN, however, we train our cGAN with $\mathcal{L}_{1,\mathsf{SD},P_{\mathsf{train}}}$ regularization and we do not use MBSD at the discriminator. We use the same general training and testing procedure described in Section 4.2, but with $\beta_{\mathsf{adv}} =$ 5e-3, $n_{\mathsf{batch}} = 100$, and 110 epochs of cGAN training. Running PyTorch on a server with 4 Tesla A100 GPUs, each with 82 GB of memory, the training takes approximately 2 days.

Table J.1: The mean and covariance components of CFID, along with the total CFID, for the generative models in the MRI and inpainting experiments. For the MRI experiment, CFID^1 used 72 test samples and $P = 32$, CFID^2 used 2 376 test samples and $P = 8$, and CFID^3 used all 14 576 samples and $P = 1$. For the inpainting experiment, CFID^1 used 1 000 test images and $P = 32$, CFID^2 used 3 000 test and validation images and $P = 8$, and CFID^3 used all 30 000 images and $P = 1$.

Model	$\text{CFID}^1_{\mathsf{mean}}\downarrow$	$\text{CFID}^1_{\mathsf{cov}}\downarrow$	$\text{CFID}^1\downarrow$	$\text{CFID}^2_{\mathsf{mean}}\downarrow$	$\text{CFID}^2_{\mathsf{cov}}\downarrow$	$\text{CFID}^2\downarrow$	$\text{CFID}^3_{\mathsf{mean}}\downarrow$	$\text{CFID}^3_{\mathsf{cov}}\downarrow$	$\text{CFID}^3\downarrow$
				$R = 4$ MRI					
Langevin (Jalal [19])	1.89	3.40	5.29	-	-	-	-	-	-
cGAN (Adler [8])	3.12	3.27	6.39	2.79	1.48	4.27	2.71	1.10	3.82
cGAN (Ohayon [23])	1.94	2.12	4.06	2.27	1.00	3.27	2.29	0.66	2.95
cGAN (Ours)	**0.98**	**2.12**	**3.10**	**0.86**	**0.68**	**1.54**	**0.86**	**0.43**	**1.29**
				$R = 8$ MRI					
Langevin (Jalal [19])	2.61	4.73	7.34	-	-	-	-	-	-
cGAN (Adler [8])	5.00	5.10	10.10	4.16	2.14	6.30	4.09	1.63	5.72
cGAN (Ohayon [23])	2.73	3.31	6.04	3.07	1.52	4.59	3.30	0.97	4.27
cGAN (Ours)	**1.55**	**3.32**	**4.87**	**1.24**	**0.99**	**2.23**	**1.17**	**0.62**	**1.79**
				Inpainting					
Score SDE (Song [20])	0.97	**38.69**	**39.66**	-	-	-	0.90	**4.21**	5.11
CoModGAN (Zhao [9])	0.42	41.21	41.63	0.35	25.39	25.74	0.32	4.98	5.29
cGAN (Ours)	**0.32**	39.41	39.73	**0.25**	**22.32**	**22.58**	**0.24**	4.45	**4.69**

I.2.2 CoModGAN

We use the PyTorch implementation of CoModGAN from [44] and train the model to inpaint a 128×128 centered square on 256×256 CelebA-HQ images. The total training time on a server with 4 NVIDIA A100 GPUs, each with 82 GB of memory, is roughly 2 days.

I.2.3 Score-based SDE

For the inpainting experiment in Section 4.3, we compare against Song et al.'s more recent SDE technique [20], for which we use the publicly available pretrained weights, the suggested settings for the 256×256 CelebA-HQ dataset, and the code from the official PyTorch implementation [45]. We generate 32 samples for all $1\,000$ images in our test set, using a batch-size of 20 and generating 32 samples for each batch element concurrently. The total generation time on a server with 4 NVIDIA A100 GPUs, each with 82 GB of memory, is roughly 9 days.

J Additional experimental results

J.1 CFID decomposition into mean and covariance components

In this section, we investigate the small-sample bias effects of CFID, which have been previously noted in [22]. To do this, we write the CFID from (F.1) as a sum of two terms: a term that quantifies the conditional-mean error and a term that quantifies the conditional-covariance error:

$$\text{CFID} = \text{CFID}_{\mathsf{mean}} + \text{CFID}_{\mathsf{cov}} \tag{J.1}$$

$$\text{CFID}_{\mathsf{mean}} \triangleq \mathrm{E}_{\underline{\mathsf{y}}}\{\|\boldsymbol{\mu}_{\underline{\mathsf{x}}|\underline{\mathsf{y}}} - \boldsymbol{\mu}_{\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}}\|_2^2\} \tag{J.2}$$

$$\text{CFID}_{\mathsf{cov}} \triangleq \mathrm{tr}\left[\boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}} + \boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}} - 2\big(\boldsymbol{\Sigma}^{1/2}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}}\boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}}\boldsymbol{\Sigma}^{1/2}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}}\big)^{1/2}\right]. \tag{J.3}$$

To verify that (J.3) quantifies the error in $\boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}}$, notice that (J.3) equals zero when $\boldsymbol{\Sigma}_{\underline{\widehat{\mathsf{x}}}\underline{\widehat{\mathsf{x}}}|\underline{\mathsf{y}}} = \boldsymbol{\Sigma}_{\underline{\mathsf{x}}\underline{\mathsf{x}}|\underline{\mathsf{y}}}$ and is otherwise positive (by Cauchy Schwarz).

In Table J.1, we report $\text{CFID}_{\mathsf{mean}}$ and $\text{CFID}_{\mathsf{cov}}$ for the MRI and inpainting experiments, in addition to the total CFID (also shown in Tables 1 and 4). As before, we computed CFID on three test sets for each experiment, which contained 72, 2 376, and 14 576 samples respectively for MRI, and 1000, 3000, and 30 000 samples respectively for inpainting. Due to the slow sample-generation time of the Langevin/score-based methods [19, 20], we did not have the computational resources to evaluate them on all datasets, and that's why certain table entries are blank.

For both MRI experiments, Table J.1 shows our method outperforming the competing methods in both the mean and covariance components of CFID (and thus the total CFID) for all sample sizes.

And, in the inpainting experiment, Table J.1 shows our method outperforming CoModGAN in both the mean and covariance components (and thus the total CFID) for all sample sizes.

For the inpainting experiment, Table J.1 shows our method outperforming the score-based approach in total CFID on the 3000- and 30 000-sample test sets but not on the 1000-sample test set. However, we now argue that the 1000-sample inpainting experiment is heavily affected by small-sample bias, and therefore untrustworthy. Looking at the mean component of CFID (i.e., $\mathrm{CFID}^1_{\mathsf{mean}}$, $\mathrm{CFID}^2_{\mathsf{mean}}$, and $\mathrm{CFID}^3_{\mathsf{mean}}$) across the inpainting experiments, we see that the values are relatively small and stable with sample size. But looking at the covariance component of CFID (i.e., $\mathrm{CFID}^1_{\mathsf{cov}}$, $\mathrm{CFID}^2_{\mathsf{cov}}$, and $\mathrm{CFID}^3_{\mathsf{cov}}$) across the inpainting experiments, we see that the values are large and heavily dependent on sample size. For the 1000-sample inpainting experiment, the total CFID is dominated by the covariance component and thus strongly affected by small-sample bias. Consequently, for the 1000-sample inpainting experiment, the total CFID is not trustworthy.

K Additional reconstruction plots

K.1 $R = 4$ MRI Reconstruction

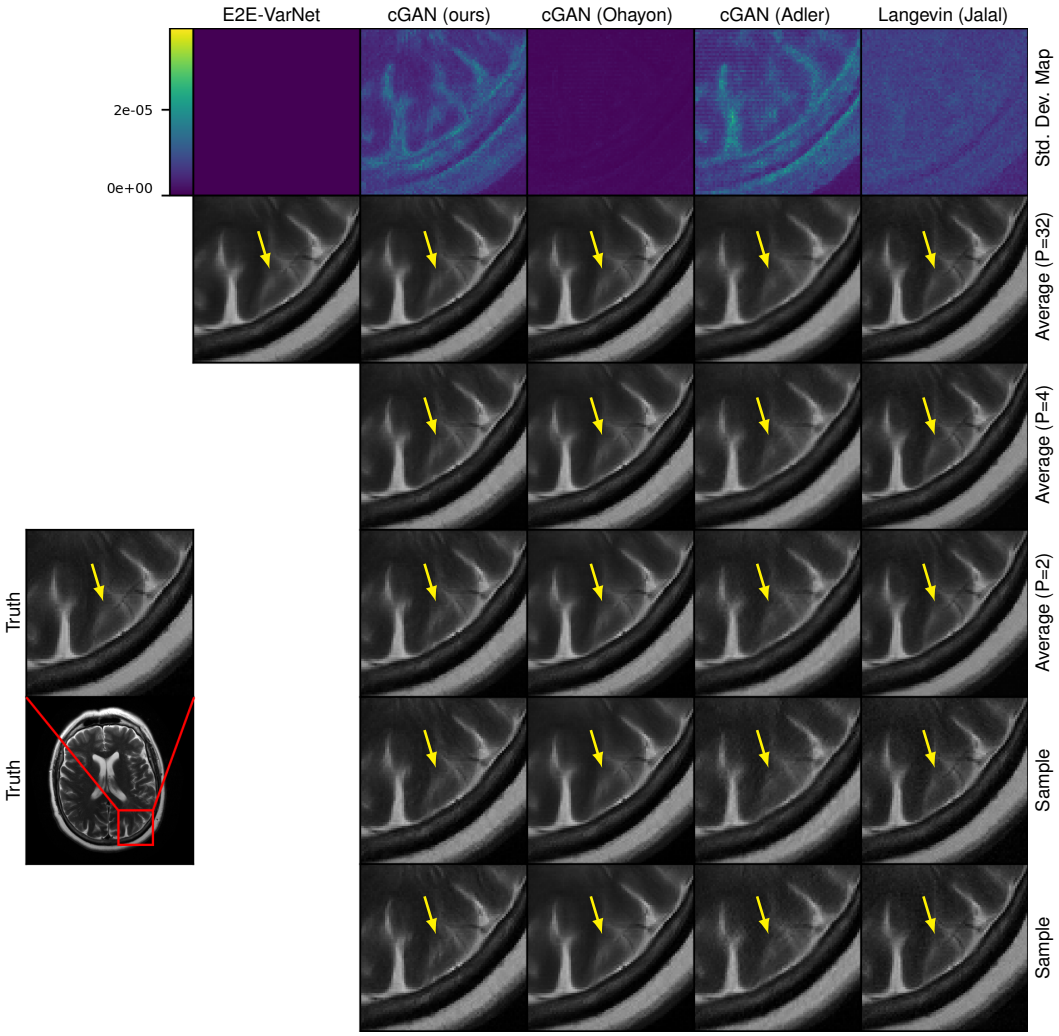

Figure K.1: Example $R = 4$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The arrows indicate regions of meaningful variation across posterior samples.

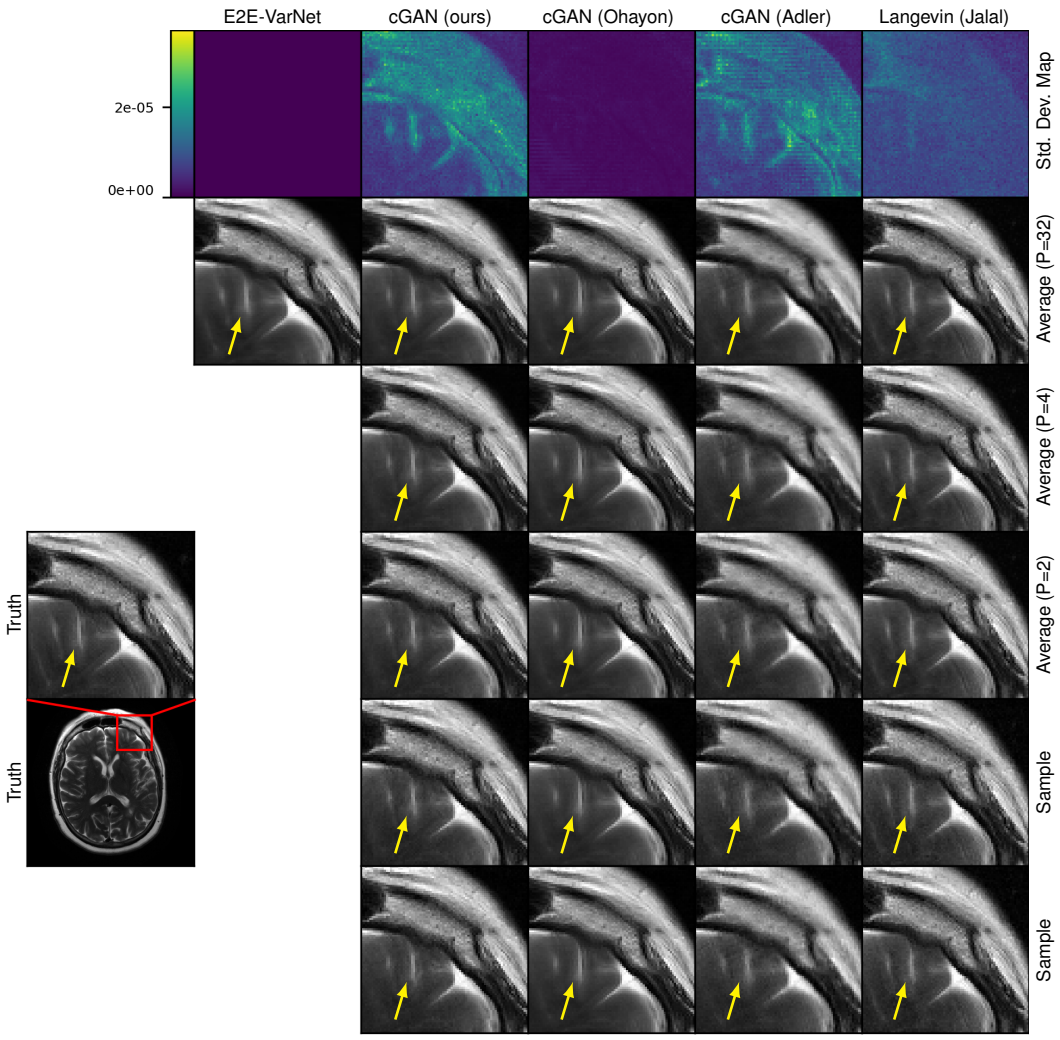

Figure K.2: Example $R = 4$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The arrows indicate regions of meaningful variation across posterior samples.

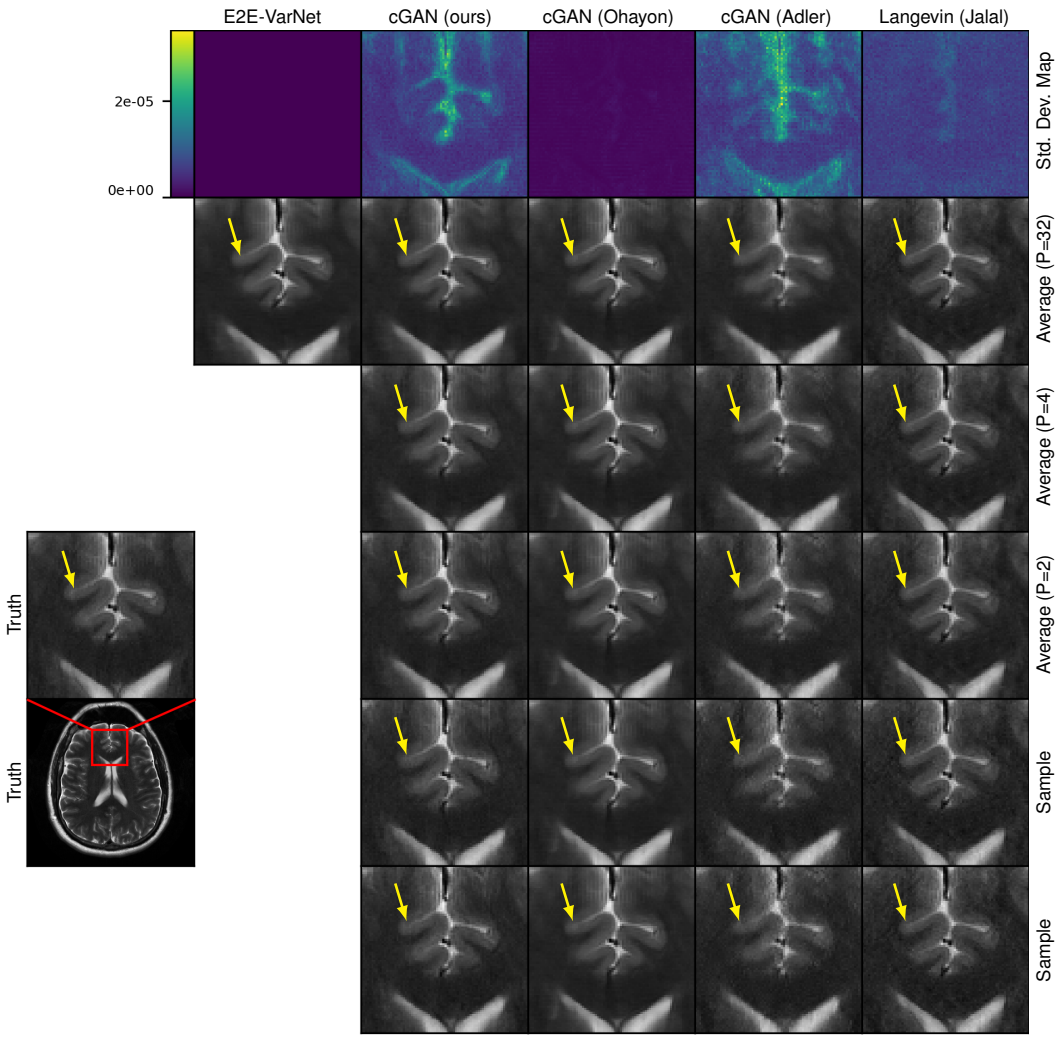

Figure K.3: Example $R = 4$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The arrows indicate regions of meaningful variation across posterior samples.

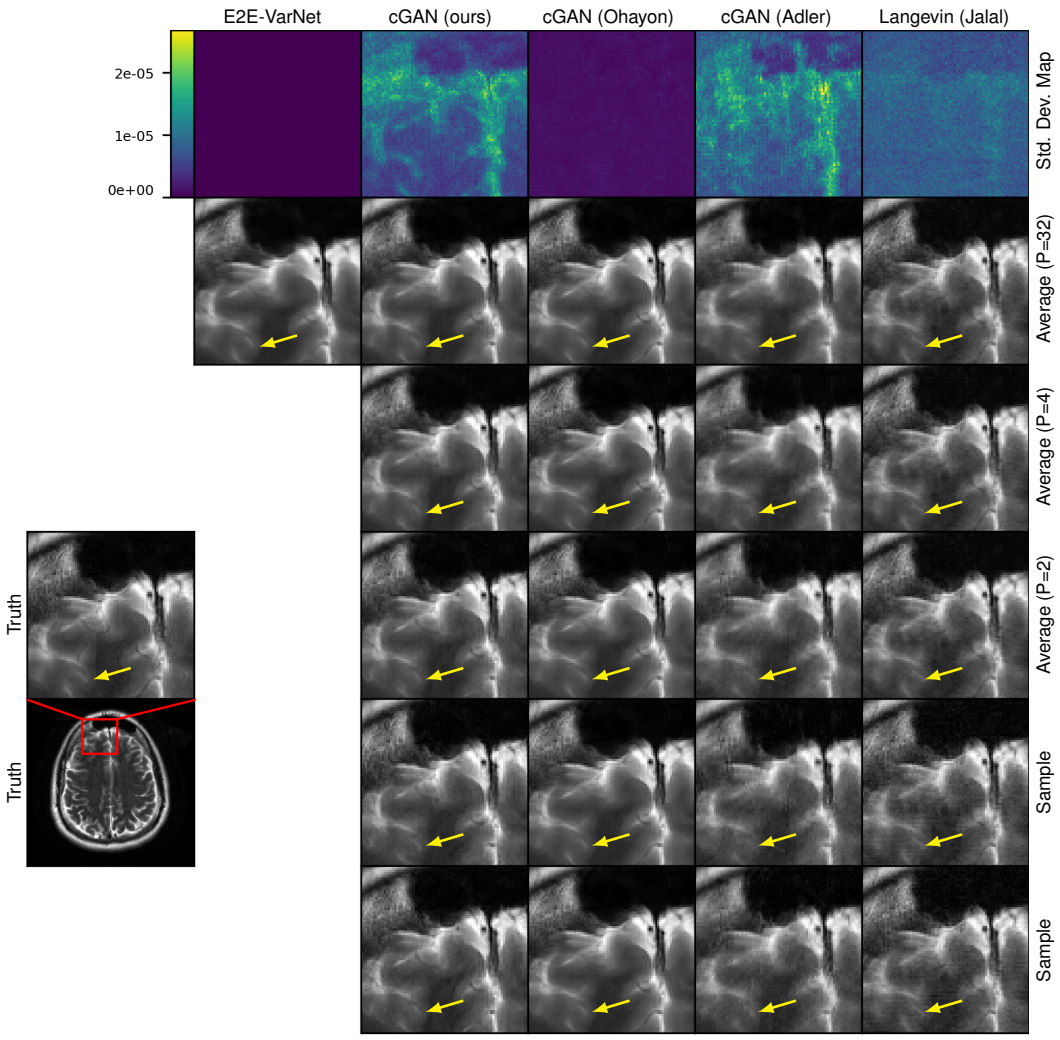

Figure K.4: Example $R = 4$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The arrows indicate regions of meaningful variation across posterior samples.

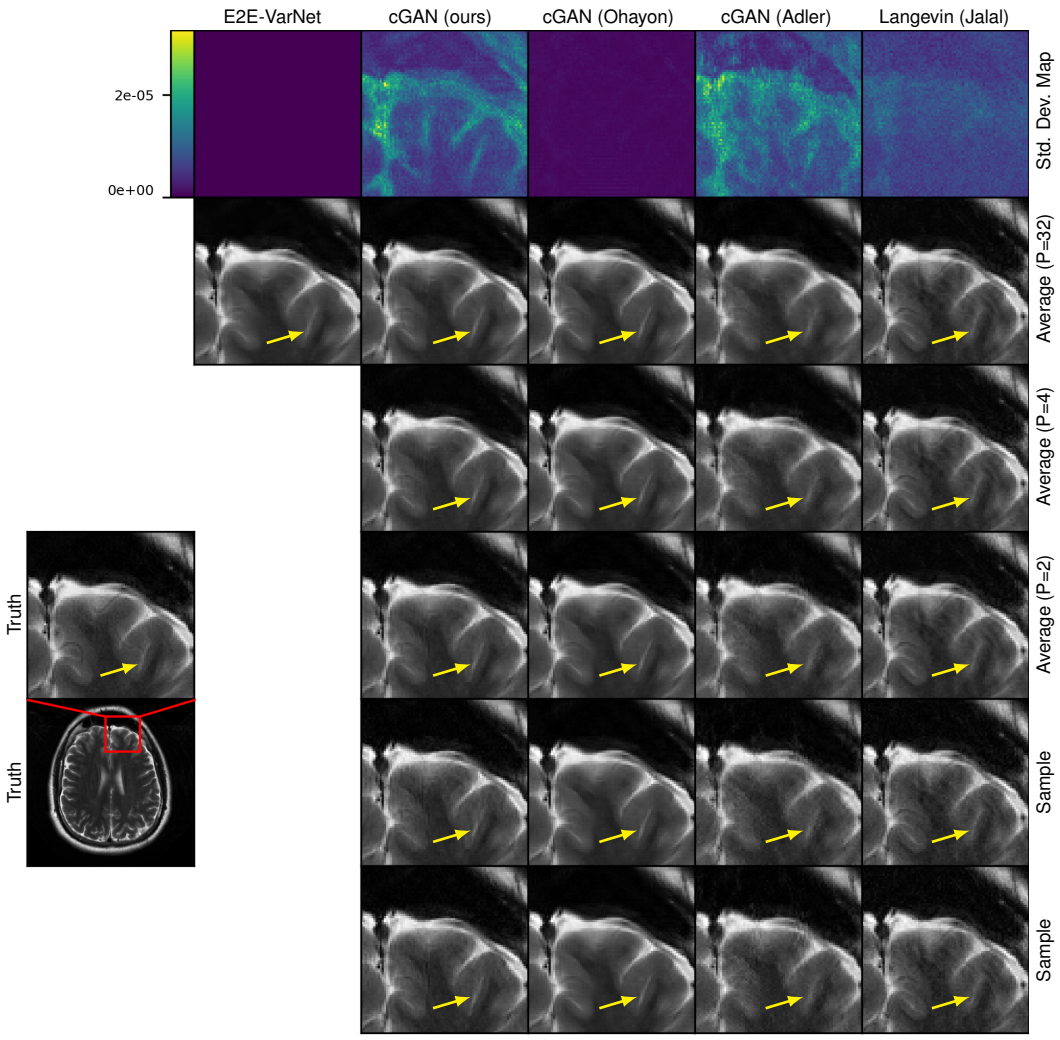

Figure K.5: Example $R = 4$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The arrows indicate regions of meaningful variation across posterior samples.

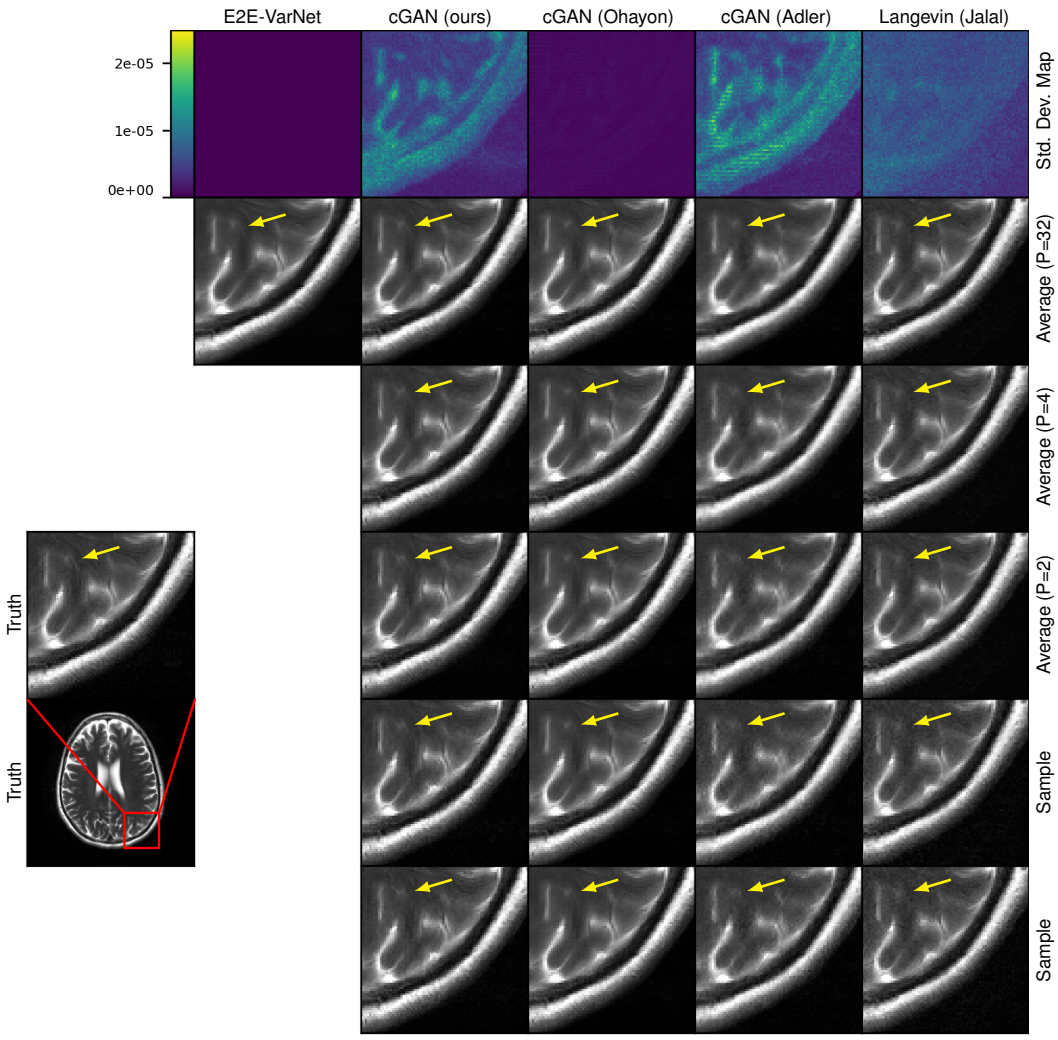

Figure K.6: Example $R = 4$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The arrows indicate regions of meaningful variation across posterior samples.

K.2 $R = 8$ MRI Reconstruction

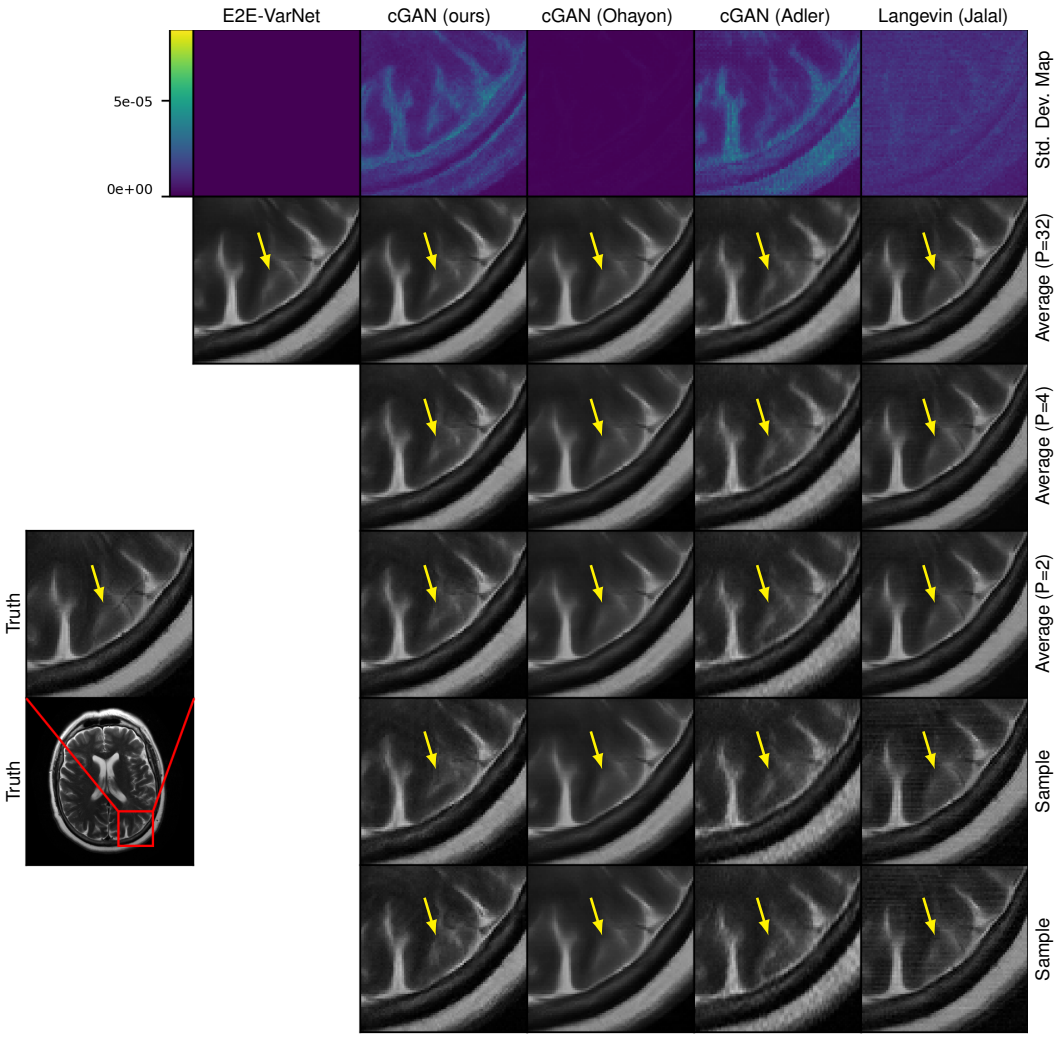

Figure K.7: Example $R = 8$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The arrows indicate regions of meaningful variation across posterior samples.

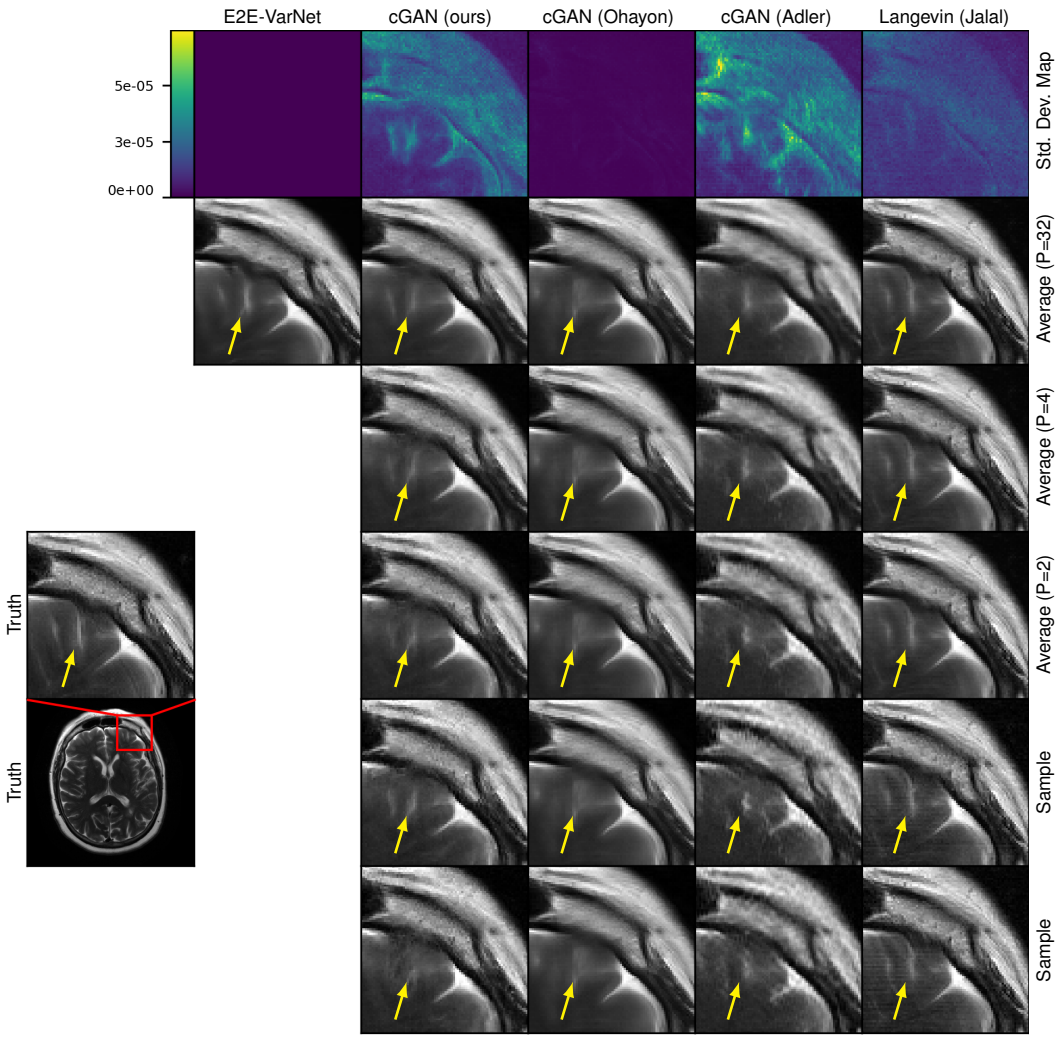

Figure K.8: Example $R = 8$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The arrows indicate regions of meaningful variation across posterior samples.

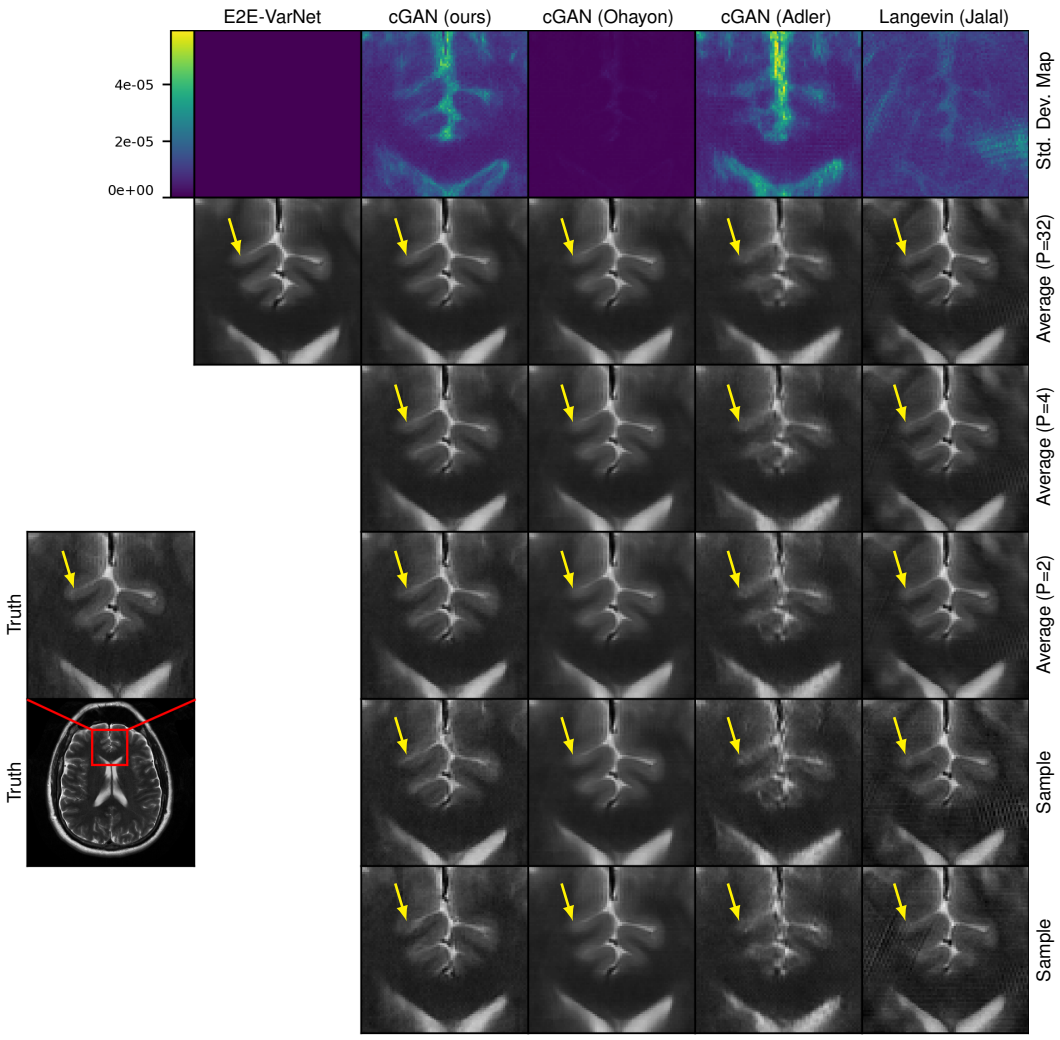

Figure K.9: Example $R = 8$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The arrows indicate regions of meaningful variation across posterior samples.

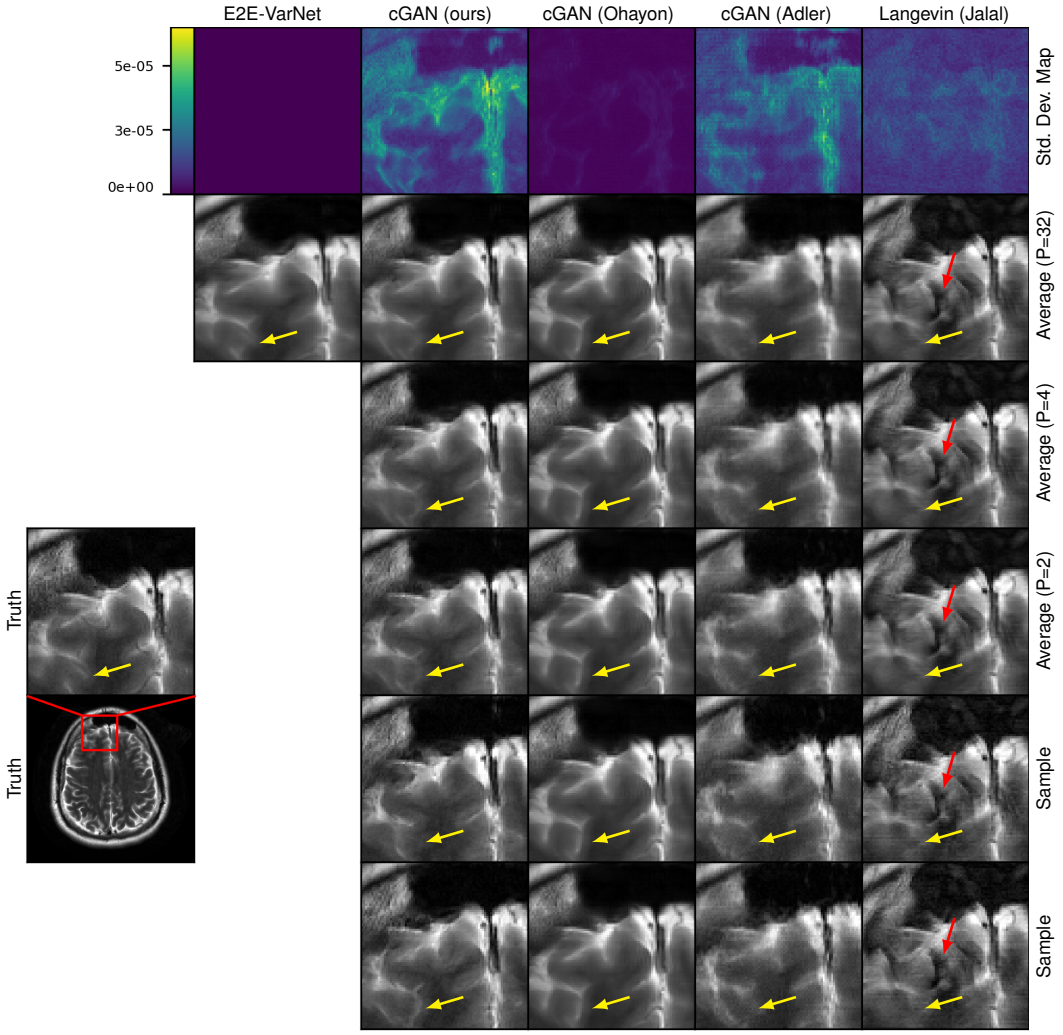

Figure K.10: Example $R = 8$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The yellow arrows indicate regions of meaningful variation across posterior samples. The red arrows show visible artifacts in the Langevin recovery.

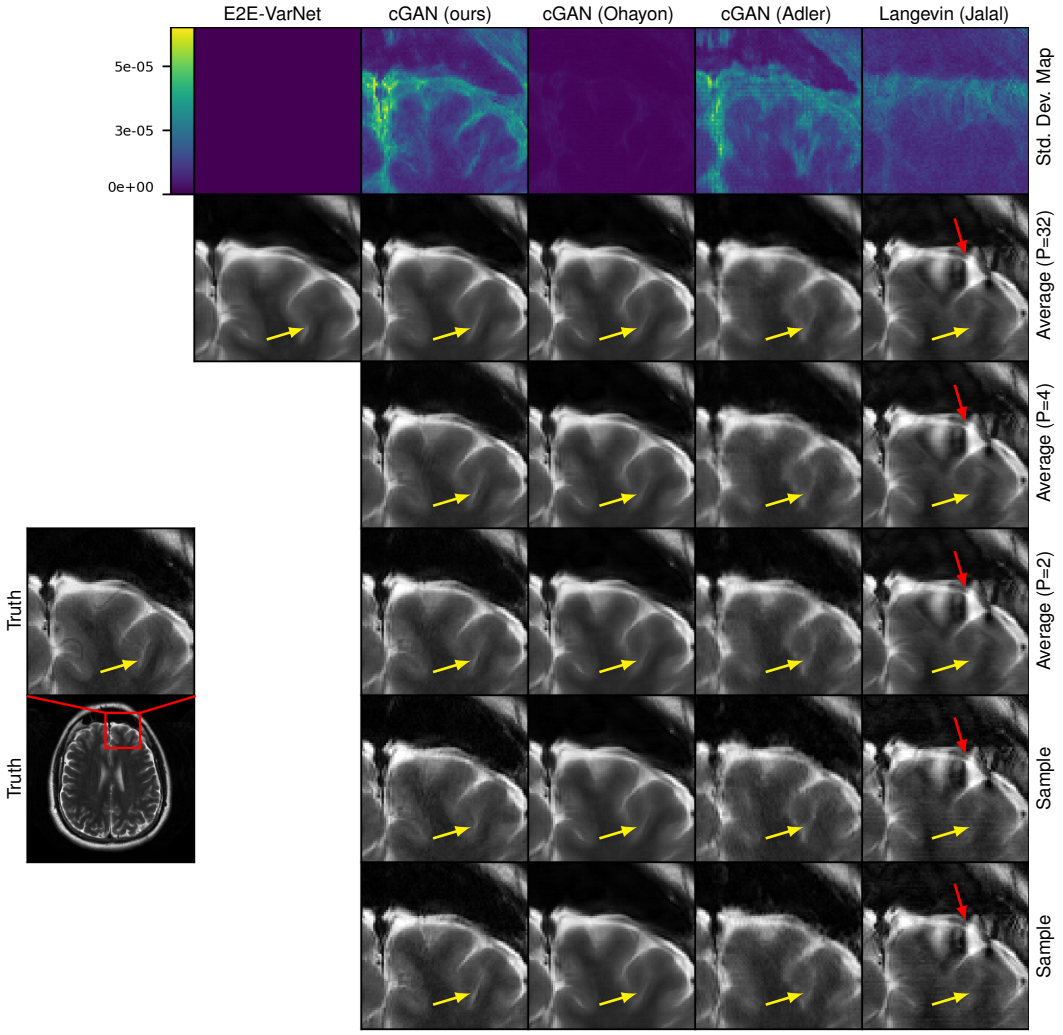

Figure K.11: Example $R = 8$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The yellow arrows indicate regions of meaningful variation across posterior samples. The red arrows show visible artifacts in the Langevin recovery.

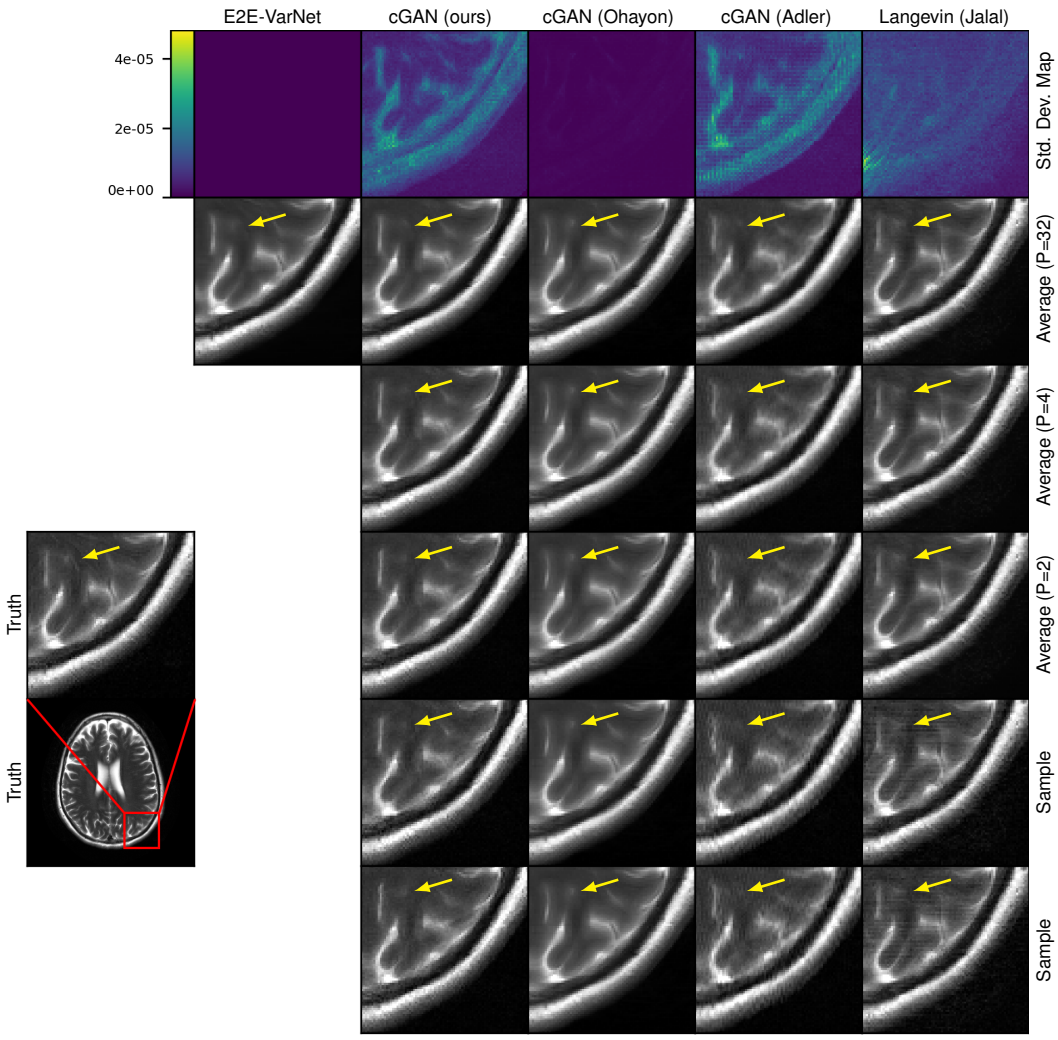

Figure K.12: Example $R = 8$ MRI reconstruction. Row one: pixel-wise SD with $P = 32$, Row two: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 32$, Row three: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 4$, Row four: $\widehat{\boldsymbol{x}}_{(P)}$ with $P = 2$, Rows five and six: posterior samples. The arrows indicate regions of meaningful variation across posterior samples.

K.3 Inpainting

Figure K.13: Example of inpainting a 128×128 square on a 256×256 resolution CelebA-HQ image.

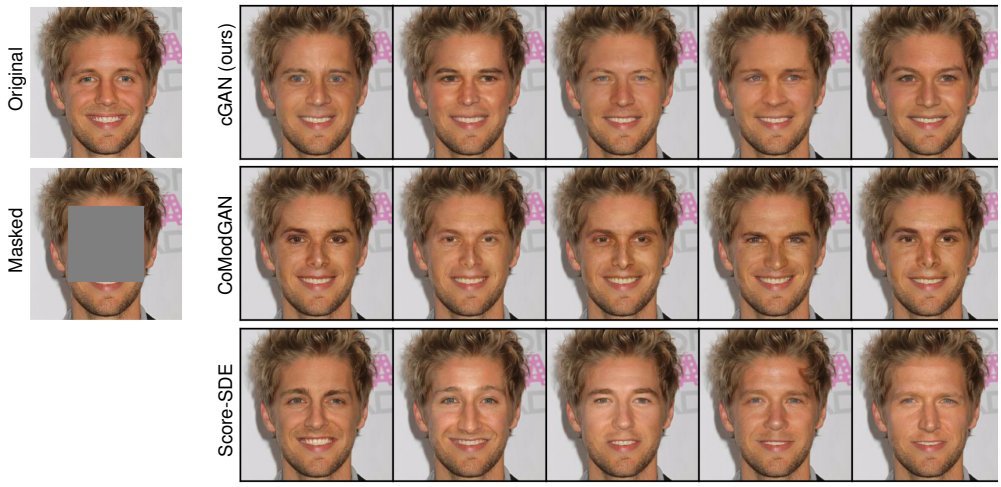

Figure K.14: Example of inpainting a 128×128 square on a 256×256 resolution CelebA-HQ image.

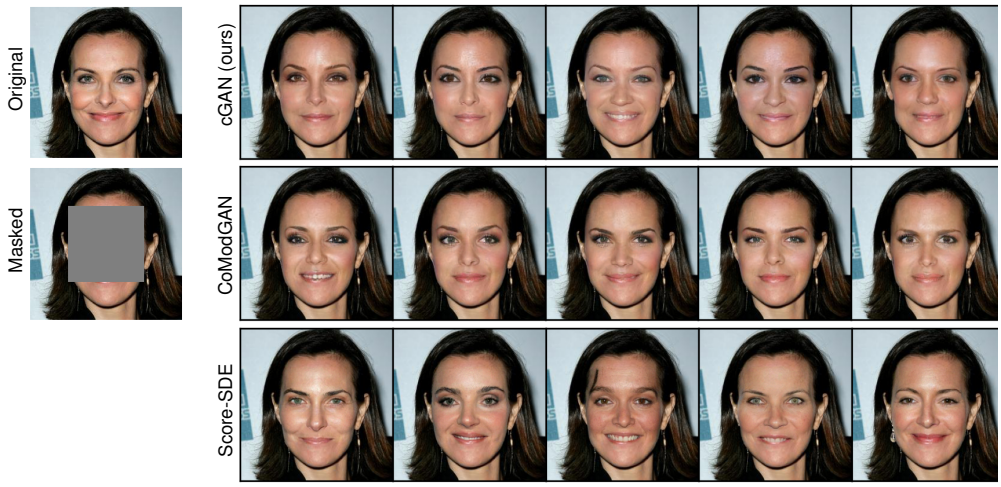

Figure K.15: Example of inpainting a 128×128 square on a 256×256 resolution CelebA-HQ image.

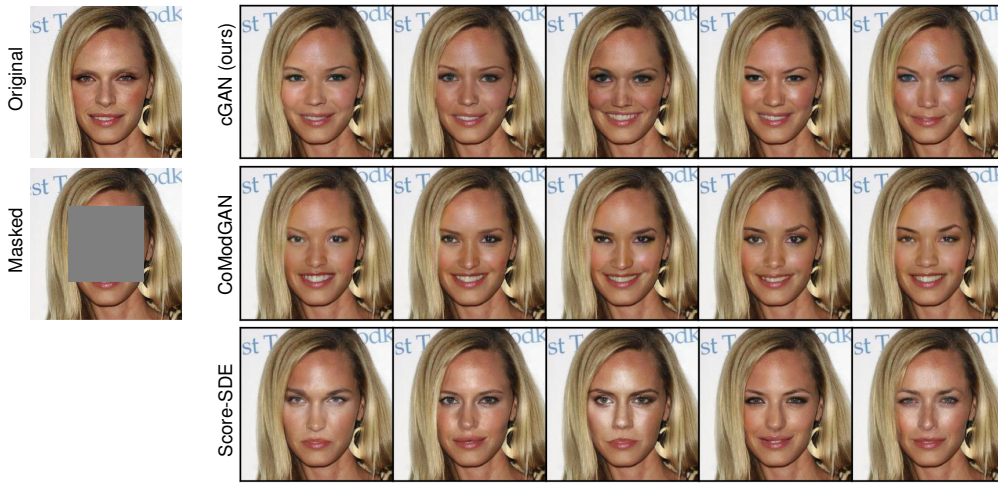

Figure K.16: Example of inpainting a 128×128 square on a 256×256 resolution CelebA-HQ image.

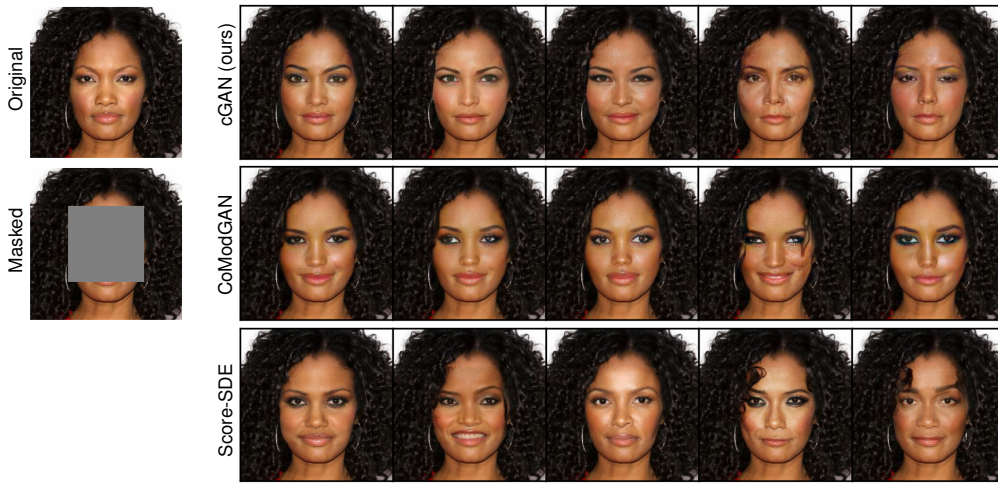

Figure K.17: Example of inpainting a 128×128 square on a 256×256 resolution CelebA-HQ image.

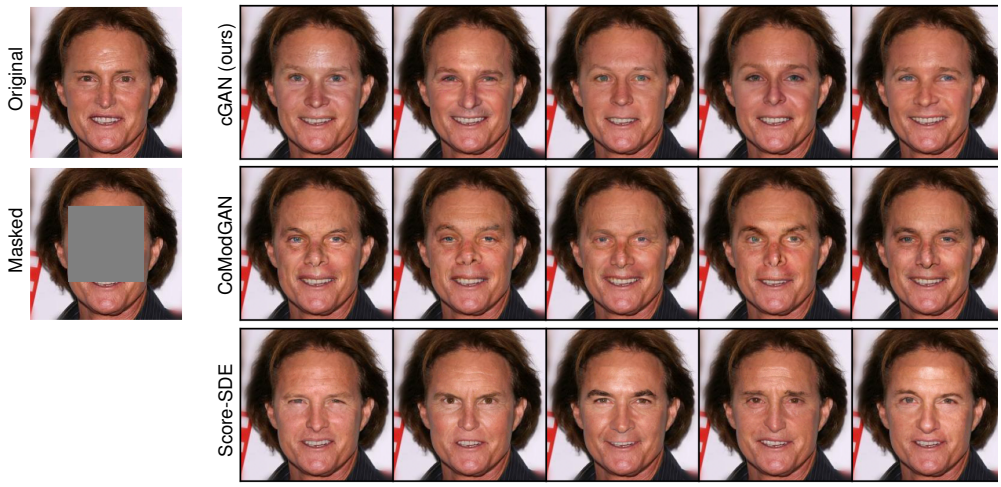

Figure K.18: Example of inpainting a 128×128 square on a 256×256 resolution CelebA-HQ image.