Bilevel Imputation Explained

1 Introduction

In the classic formulation of the imputation problem, the primary objective is to produce imputed data with the lowest possible error relative to the ground truth. Under the Missing At Random (MAR) assumption, a common approach is to synthetically mask observed values and train a model to reconstruct them. From a data-quality perspective, well-imputed data should enhance the generalizability of downstream models trained on the dataset. In this work, we adopt a setting where the imputation model is optimized to improve the performance of a supervised model—specifically, by minimizing its validation loss. We propose to intergrate the supervised validation loss directly into the objective of the imputation model through a bilevel optimization framework. This document provides a detailed explanation of the proposed method.

2 Problem Statement

To ground our approach, we begin by formally defining the imputation problem and its associated notation. We denote by $\widetilde{X} \in \mathbb{R}^{N\times d}$ the *unknown* ground truth data matrix, i.e., the complete data matrix without any missing values, where N is the number of data instances and d is the number of features. Let $\widehat{X} \in \mathbb{R}^{N\times d}$ denote the imputed data matrix produced by an imputation model, and $M \in \{0,1\}^{N\times d}$ be the corresponding binary mask matrix indicating observed entries: $M_{ij} = 1$ if the (i,j)-th entry is observed, and 0 otherwise. Define $\overline{M} := 1 - M$ to be the complement mask, indicating the missing entries.

We assume that the missing data distribution is stationary and, in our experiments, we primarily focus on the Missing At Random (MAR) setting. The goal of the imputation task is to generate imputations $\widehat{X}$ such that the discrepancy between the imputed values and the ground truth values at the missing locations is minimized. Specifically, we define the imputation loss as:

$$\mathcal{L}(\widehat{X}, \widetilde{X}, M) = \sum_{i=1}^{N} \frac{\langle \bar{m}^i,\, \ell(\hat{x}^i, \tilde{x}^i)\rangle}{\langle \bar{m}^i, \bar{m}^i\rangle},$$

where $\hat{x}^i$, $\tilde{x}^i \in \mathbb{R}^d$ are the i-th rows of $\widehat{X}$ and $\widetilde{X}$, respectively, $\bar{m}^i \in \{0,1\}^d$ is the i-th row of the complement mask $\overline{M}$, and $\ell(\cdot,\cdot)$ is an element-wise error function (e.g., absolute or squared error), and $\langle\cdot,\cdot\rangle$ indicates the standard dot product.

Note that in practice, the true data matrix $\widetilde{X}$ is not accessible, which makes the direct computation of $\mathcal{L}(\widehat{X},\widetilde{X},M)$ infeasible. Consequently, we adopt a surrogate objective by synthetically generating missing values in otherwise fully observed data, allowing us to evaluate imputation quality against known ground truth in controlled settings.

3 A Naive Formulation of the Bilevel Objective

Let the imputation model be denoted as the teacher T with parameters θ_T, and the predictor model as the student S with parameters θ_S. Consider a pair of data batches $(x_{\text{train}}, y_{\text{train}})$ and $(x_{\text{val}}, y_{\text{val}})$ corresponding to the training and validation sets, respectively. We define $T_x(x;\theta_T)$ as the set of features imputed by the teacher, and $T_y(x;\theta_T)$ as the label (in this context, the next-day $\text{PM}_{2.5}$ concentration) imputed by the teacher. The student's prediction on input x is denoted by $S(x;\theta_S)$. We impose the convention that if a feature x_j is not missing, then $\big[T_x(x;\theta_T)\big]_j = x_j$; similarly, if the label y is observed, then $T_y(x;\theta_T) = y$.

Given the imputed data $T_x(x_{\text{train}};\theta_T)$ produced by the teacher, this data is used to train the student model as follows:

$$\theta_S^* = \arg\min_{\theta_S} \Big[\underbrace{\text{MSE}\big(S\big(T_x(x_{\text{train}};\theta_T);\theta_S\big),\, T_y(x_{\text{train}};\theta_T)\big)}_{:=\mathcal{L}_{\text{train}}(\theta_T,\theta_S)} + \lambda\|\theta_S\|^2\Big].$$

Given a good teacher that is sufficiently capable of producing high-quality imputations, it is expected that the obtained θ_S^* would achieve a low loss on the validation set, i.e. $\text{MSE}\big(S\big(T_x(x_{\text{val}};\theta_T);\theta_S^*\big),\, y_{\text{val}}\big) := \mathcal{L}_{\text{val}}(\theta_T,\theta_S^*)$.

Notice that the optimal student parameters θ_S^* always depends on the teacher's parameters θ_T via the imputed data $T_x(x_{\text{train}};\theta_T)$. We can explicitly express the dependency as $\theta_S^*(\theta_T)$. As an immediate observation, the student loss on validation data $\mathcal{L}_{\text{val}}(\theta_T,\theta_S^*)$ is also a "function" of θ_T. Therefore, we could further optimize $\mathcal{L}_{\text{val}}$ with respect to θ_T. The overall optimization problem can thus be formulated as a bilevel optimization problem:

$$\min_{\theta_T} \mathcal{L}_{\text{val}}\big(\theta_T, \theta_S^*(\theta_T)\big) \tag{1}$$

$$\text{where } \theta_S^*(\theta_T) = \arg\min_{\theta_S}\Big[\mathcal{L}_{\text{train}}(\theta_T,\theta_S) + \lambda\|\theta_S\|^2\Big] \tag{2}$$

Intuitively, the teacher is optimized to generate imputed outputs that enhance the performance of the student model. However, this optimization framework has a significant flaw, where the teacher may introduce unrealistic artifacts that exploit the student's learning behavior.

4 Analysing the Artifact Problem

To understand the source of the artifact issue, we begin by analyzing the validation loss defined in Eq. 1:

$$\mathcal{L}_{\text{val}}(\theta_T, \theta_S^*(\theta_T)) = \text{MSE}\left(S\left(T_x(x_{\text{val}}; \theta_T); \theta_S^*(\theta_T)\right), \, y_{\text{val}}\right).$$

When computing the gradient of this loss with respect to the teacher's parameters θ_T, we obtain:

$$\frac{d\mathcal{L}_{\text{val}}}{d\theta_T} = \underbrace{\frac{\partial \mathcal{L}_{\text{val}}}{\partial \theta_T}\bigg|_{\theta_S=\theta_S^*(\theta_T)}}_{\text{(A)}} + \underbrace{\frac{\partial \mathcal{L}_{\text{val}}}{\partial \theta_S}\bigg|_{\theta_S=\theta_S^*(\theta_T)} \cdot \frac{\partial \theta_S^*(\theta_T)}{\partial \theta_T}}_{\text{(B)}}.$$

Here, term (A) represents the direct gradient of the validation loss with respect to the teacher's output $T_x(x_{\text{val}}; \theta_T)$. This allows the teacher to be trained by directly minimizing the student's validation loss. The pseudocode below illustrates an update process that corresponds to term (A):

```
# 1. Impute the validation data with teacher and predict with
 ↪  student.
imputed_x_val = teacher(x_val)
with torch.no_grad():
  for p in student.parameters():
    p.requires_grad = False  # Freeze student's parameters

y_pred   = student(imputed_x_val)
loss_val = criterion(y_pred, y_val) # MSE

# 2. Backprop only the teacher parameters
teacher_optimizer.zero_grad()
loss_val.backward()
teacher_optimizer.step()
```

As shown, term (A) enables the teacher to exploit adversarial signals from the student's validation loss. Specifically, the teacher can manipulate its imputed outputs $T_x(x_{\text{val}})$ to leak information about the true label y_{val}, thereby artificially lowering the validation error $\mathcal{L}_{\text{val}}$. In other words, if the teacher has access to y_{val}, it may encode this target directly into the imputed inputs. This undermines the integrity of the learning process, causing the teacher to generate unrealistic or distorted imputations that do not generalize beyond the current validation set.

5 Tackling the Artifact Problem

We propose to mitigate the artifact issue by removing the use of teacher-imputed data in the validation loss of the student (i.e., Eq. 1). Specifically, we omit imputing x_{val}, resulting in the following revised outer loss:

$$\mathcal{L}_{\text{outer}}\big(\theta_T, \theta_S^*(\theta_T)\big) = \text{MSE}\big(S(x_{\text{val}}; \theta_S^*(\theta_T)),\, y_{\text{val}}\big)$$

This modification requires the using a student that can infer over missing features directly, instead of relying on the teacher's imputed inputs. To simulate the missing data distribution during training, we introduce feature dropout on the teacher's output in the inner optimization objective (Eq. 2). The modified inner optimization becomes:

$$\begin{aligned}\theta_S^*(\theta_T) &= \arg\min_{\theta_S} \left[\mathcal{L}_{\text{inner}}(\theta_T, \theta_S) + \lambda\|\theta_S\|^2\right] \\ \text{with} \quad \mathcal{L}_{\text{inner}}(\theta_T, \theta_S) &= \text{MSE}\left(S\left(D_p(T_x(x_{\text{train}}; \theta_T)); \theta_S\right), T_y(x_{\text{train}}; \theta_T)\right)\end{aligned}$$

where D_p denotes the application of feature-level dropout with a drop rate of p. The gradient of the revised outer loss can be expressed as:

$$\frac{d\mathcal{L}_{\text{outer}}}{d\theta_T} = \underbrace{\frac{\partial \mathcal{L}_{\text{outer}}}{\partial \theta_T}\bigg|_{\theta_S = \theta_S^*(\theta_T)}}_{=0} + \frac{\partial \mathcal{L}_{\text{outer}}}{\partial \theta_S}\bigg|_{\theta_S = \theta_S^*(\theta_T)} \cdot \frac{\partial\, \theta_S^*(\theta_T)}{\partial \theta_T}.$$

We observe that the adversarial signal–i.e., the direct gradient flow from the outer loss to θ_T–is eliminated. As a result, the teacher no longer has a direct mechanism to leak label information through the imputed features, and is instead forced to learn more generalizable representations.

The final objective for training the teacher combines the outer loss with a Masked Autoencoder (MAE) loss:

$$\mathcal{L}_{\text{teach}}(\theta_T) = \mathcal{L}_{\text{outer}}(\theta_T, \theta_S^*(\theta_T)) + \mathcal{L}_{\text{mae}}(\theta_T)$$

Intuitively, the imputed data produced by the teacher in the proposed method can be viewed as a form of data augmentation, where feature dropout introduces variations of the incomplete inputs. Unlike full reconstruction, this approach encourages learning from partial information.

6 Derivation of the Teacher's Update Rule

In this section, we provide a detailed derivation of the outer gradient used to update the teacher model parameters, as discussed in Section 5. This derivation follows the approach introduced in the iMAML method[2].

Recall the bilevel optimization problem:

$$\begin{aligned}\mathcal{L}_{\text{outer}}(\theta_T, \theta_S^*) &= \text{MSE}\left(S(x_{\text{val}}; \theta_S^*(\theta_T)), y_{\text{val}}\right), \\ \theta_S^*(\theta_T) &= \arg\min_{\theta_S} \left[\mathcal{L}_{\text{inner}}(\theta_T, \theta_S) + \frac{\lambda}{2}\|\theta_S\|^2\right], \\ \mathcal{L}_{\text{inner}}(\theta_T, \theta_S) &= \text{MSE}\left(S(D_p(T_x(x_{\text{train}}; \theta_T)); \theta_S),\ T_y(x_{\text{train}}; \theta_T)\right),\end{aligned}$$

where D_p denotes a stochastic dropout operation with rate p applied to the features imputed by the teacher.

We now expand the gradient of the outer loss:

$$\begin{aligned}\frac{d\mathcal{L}_{\text{outer}}}{d\theta_T} &= \underbrace{\frac{\partial \mathcal{L}_{\text{outer}}}{\partial \theta_T}}_{=0} + \frac{\partial \mathcal{L}_{\text{outer}}}{\partial \theta_S} \cdot \frac{\partial \theta_S^*}{\partial \theta_T} \\ &= \frac{\partial \mathcal{L}_{\text{outer}}}{\partial \theta_S} \cdot \frac{\partial \theta_S^*}{\partial \theta_T}\,. \qquad (3)\end{aligned}$$

The first term vanishes by construction since the teacher is excluded from the validation loss. Hence, we focus on computing the second term. Since θ_S^* is defined as the minimum of the inner objective:

$$\theta_S^* = \arg\min_{\theta_S} \left[\mathcal{L}_{\text{inner}}(\theta_T, \theta_S) + \frac{\lambda}{2}\|\theta_S\|^2\right],$$

the stationary point condition yields:

$$\begin{aligned}&\frac{\partial}{\partial \theta_S}\left[\mathcal{L}_{\text{inner}}(\theta_S) + \lambda\|\theta_S\|^2\right]\Bigg|_{\theta_S=\theta_S^*} = 0 \\ \Rightarrow \quad &\frac{\partial}{\partial \theta_S}\mathcal{L}_{\text{inner}}(\theta_S^*) + \frac{\lambda}{2}\theta_S^* = 0 \\ \Rightarrow \quad &\theta_S^* = -\frac{1}{\lambda}\frac{\partial}{\partial \theta_S}\mathcal{L}_{\text{inner}}(\theta_S^*)\,.\end{aligned}$$

Taking the derivative with respect to θ_T, we obtain:

$$
\begin{aligned}
& \frac{d\theta_S^*}{d\theta_T} = -\frac{1}{\lambda}\left(\frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial\theta_T \partial\theta_S} + \frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial\theta_S^2} \cdot \frac{d\theta_S^*}{d\theta_T}\right) \\
\Rightarrow \quad & \left(I + \frac{1}{\lambda}\frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial\theta_S^2}\right) \cdot \frac{d\theta_S^*}{d\theta_T} = -\frac{1}{\lambda}\frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial\theta_T \partial\theta_S} \\
\Rightarrow \quad & \frac{d\theta_S^*}{d\theta_T} = -\left(\frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial\theta_S^2} + \lambda I\right)^{-1} \cdot \frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial\theta_T \partial\theta_S} \\
\Rightarrow \quad & \frac{d\theta_S^*}{d\theta_T} = -\left(\nabla^2_{\theta_S}\mathcal{L}_{\text{inner}} + \lambda I\right)^{-1} \cdot \left(\nabla_{\theta_T}\nabla_{\theta_S}\mathcal{L}_{\text{inner}}\right)^T . \qquad (4)
\end{aligned}
$$

Substituting Eq. (4) into Eq. (3), we obtain the final expression for the teacher's gradient:

$$\frac{d\mathcal{L}_{\text{outer}}}{d\theta_T} = -\left(\nabla_{\theta_S}\mathcal{L}_{\text{outer}}\right)^T \cdot \left(\nabla^2_{\theta_S}\mathcal{L}_{\text{inner}} + \lambda I\right)^{-1} \cdot \left(\nabla_{\theta_T}\nabla_{\theta_S}\mathcal{L}_{\text{inner}}\right)^T .$$

This expression highlights that updating the teacher parameters requires inverting a Hessian matrix with respect to the student parameters. In practice, this operation is computationally expensive. Approximation methods to avoid full matrix inversion are discussed in the following section.

7 Training Algorithm and Pseudocode

In this section, we provide a detailed description of the bilevel training procedure applied to the Masked Autoencoder. The algorithm is inspired by the implicit MAML method[2].

Initially, the teacher model (MAE) is pretrained using its reconstruction objective $\mathcal{L}_{\text{MAE}}$ over the training data. To ensure that the student model satisfies the stationarity condition

$$\frac{\partial}{\partial\theta_S}\left[\mathcal{L}_{\text{inner}}(\theta_S) + \lambda\|\theta_S\|^2\right]\Bigg|_{\theta_S=\theta_S^*} = 0,$$

we also pretrain the student using the inner loss $\mathcal{L}_{\text{inner}}$ on the imputed outputs of the teacher for a predefined number of steps.

During each bilevel training iteration, the optimal student parameter θ_S^* is approximated using a single-step gradient update:

$$\theta_S^* \approx \theta_S^{(t+1)} = \theta_S^{(t)} - \eta_S \cdot \nabla_{\theta_S}\mathcal{L}_{\text{inner}},$$

where η_S is the student learning rate.

The inverse Hessian term $\left(\nabla^2_{\theta_S}\mathcal{L}_{\text{inner}} + \lambda I\right)^{-1}$ is computationally expensive and numerically unstable to compute exactly. Therefore, we approximate this term using the Conjugate Gradient method [1, 3].

The full bilevel training pseudocode is outlined in Algorithm 1.

Algorithm 1 Bilevel Training Algorithm for Imputation Model

```
Input  : Training set with missing values ($x_{\text{train}}, m_{\text{train}}$), validation set with missing values
         ($x_{\text{val}}, m_{\text{val}}$); MAE masking ratio $p$; regularization coefficient $\lambda$; learning rates $\eta_S$, $\eta_T$;
         Number of CG iterations $K$
Output: $\theta_T^{(N)}$                                     ▷ Final teacher parameters for imputation inference
1  Initialize $\theta_T^{(0)}$ and $\theta_S^{(0)}$
2  for $t = 0$ to $N - 1$ do
3      Perform imputation on the training set:
           $\hat{x}_{\text{train}} = T_x(x_{\text{train}}, m_{\text{train}}; \theta_T^{(t)})$
           $\hat{y}_{\text{train}} = T_y(x_{\text{train}}, m_{\text{train}}; \theta_T^{(t)})$
4      Update student model on imputed data:
           $\theta_S^{(t+1)} = \theta_S^{(t)} - \eta_S \cdot \nabla_{\theta_S} \operatorname{MSE}(S(\hat{x}_{\text{train}}; \theta_S^{(t)}),\ \hat{y}_{\text{train}})$
5      Compute intermediate loss values:
           $\mathcal{L}_{\text{inner}} = \operatorname{MSE}(S(\hat{x}_{\text{train}}; \theta_S^{(t+1)}),\ \hat{y}_{\text{train}})$
           $\mathcal{L}_{\text{outer}} = \operatorname{MSE}(S(x_{\text{val}}; \theta_S^{(t+1)}),\ y_{\text{val}})$
6      Compute gradient components:
           $v = \nabla_{\theta_S} \mathcal{L}_{\text{outer}}, \quad H = \nabla^2_{\theta_S} \mathcal{L}_{\text{inner}}$
7      Use $K$-step Conjugate Gradient to approximate:
           $u_{\text{IG}} \approx (H + \lambda I)^{-1} v$
8      Compute the implicit bilevel gradient for teacher:
           $h_{\text{impl}}^{(t)} = -\nabla_{\theta_T} \left[ (\nabla_{\theta_S} \mathcal{L}_{\text{inner}})^\top u_{\text{IG}} \right]$
9      Compute the direct MAE gradient:
           $g_{\text{mae}}^{(t)} = \nabla_{\theta_T} \mathcal{L}_{\text{MAE}}$
10     Update teacher parameters:
           $\theta_T^{(t+1)} = \theta_T^{(t)} - \eta_T \cdot (h_{\text{impl}}^{(t)} + g_{\text{mae}}^{(t)})$
11 end
12 return $\theta_T^{(N)}$                                  ▷ Return final teacher model for inference
```

References

[1] J. L. Nazareth. “Conjugate gradient method”. In: *WIREs Computational Statistics* 1.3 (2009), pp. 348–353. DOI: https://doi.org/10.1002/wics.13. eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.13. URL: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.13.

[2] Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. *Meta-Learning with Implicit Gradients*. 2019. arXiv: 1909.04630 [cs.LG]. URL: https://arxiv.org/abs/1909.04630.

[3] Jonathan Richard Shewchuk. “An Introduction to the Conjugate Gradient Method Without the Agonizing Pain”. In: 1994. URL: https://api.semanticscholar.org/CorpusID:6491967.