Modify the influence function suitable for non-convex model.

Gradient Surgery for One-shot Unlearning on Generative Model

Seohui Bae [1] **Seoyoon Kim** [1] **Hyemin Jung** [1] **Woohyung Lim** [1]

Abstract

Recent regulation on right-to-be-forgotten emerges tons of interest in unlearning pre-trained machine learning models. While approximating a straightforward yet expensive approach of retrain-from-scratch, recent machine unlearning methods unlearn a sample by updating weights to remove its influence on the weight parameters. In this paper, we introduce a simple yet effective approach to remove a data influence on the deep generative model. Inspired by works in multi-task learning, we propose to manipulate gradients to regularize the interplay of influence among samples by projecting gradients onto the normal plane of the gradients to be retained. Our work is agnostic to statistics of the removal samples, outperforming existing baselines while providing theoretical analysis for the first time in unlearning a generative model.

1. Introduction

Suppose a user wants to get rid of his/her face image anywhere in your facial image generation application - including the database and the generative model on which it is trained. Is the expensive retrain-from-scratch the only solution for this kind of request? As the use of personal data has been increased in training the machine learning models for online service, meeting individual demand for privacy or the rapid change in the legislation of General Data Protection Registration (GDPR) is inevitable to ML service providers nowadays. This request on 'Right-To-Be-Forgotten (RTBF)' might be a one-time or in-series, scaling from a feature to a number of tasks, querying single instance to multiples. A straightforward solution for unlearning a single data might be to retrain a generative model from scratch without data of interest. This approach, however, is intractable in practice considering the grand size and complexity of the latest generative models (Rombach et al., 2022; Child, 2020) and the continual request for removal.

Unlearning, thereafter, aims to approximate this straightforward-yet-expensive solution of retrain-from-scratch time and computation efficiently. First-order data-influence-based approximate unlearning is currently considered the state-of-the-art approach to unlearning machine learning models in general. Grounded by the notion of data influence (Koh & Liang, 2017), a simple one-step Newton's update certifies sufficiently small bound between retrain-from-scratch (Guo et al., 2020). Nonetheless, those relaxations are infeasible to the non-convex deep neural networks (*e.g.* generative model) where the gap is not certifiably bounded and the process of computing the inverse of hessian is intractable. Several recent works also have affirmed that these relaxed alternatives perform poorly on deep neural networks (Golatkar et al., 2021; Liu et al., 2022) and even that on generative models have not been explored yet.

Contribution In this work, we propose a novel one-shot unlearning method for unlearning samples from pre-trained deep generative model. Relaxing the definition of influence function on parameters in machine unlearning (Koh & Liang, 2017; Basu et al., 2020), we focus on the influence of a single data on the *test loss* of the others and propose a simple and cost-effective method to minimize this interdependent influence to approximate retrain-from-scratch. We summarize our contributions as follows:

- We propose to annul the influence of samples on generations with simple gradient manipulation.
- Agnostic to removal statistics and thus applied to any removals such as a single data, a class, some data feature, etc.
- Grounded by a theoretical analysis bridging standard machine unlearning to generative model.

2. Gradient Surgery for One-shot Data Removals on Generative Model

Notations Let $D = \{\mathbf{x}_i\}_{i=1}^N \subseteq \mathcal{X}$ be the training data where $\mathbf{x}_i \in \mathcal{X}$ is input. Let $D_f \subseteq D$ be a subset of training data that is to be forgotten (*i.e.* forget set) and $D_r = D \setminus D_f$

[1]LG AI Research, Seoul, South Korea. Correspondence to: Seohui Bae <seohui.bae@lgresearch.ai>, Woohyung Lim <w.lim@lgresearch.ai>.

Accepted to the 1^{st} *Workshop on Generative AI and Law*, co-located with the *International Conference on Machine Learning*, Honolulu, Hawaii, USA. 2023.

be remaining training data of which information we want to retain. Recall that the goal of unlearning is to approximate the deep generative model retrained from scratch with only D_r, which we denote as f_{θ^*} parameterized by θ^*. Then, our goal is to unlearn $D_f \subseteq D$ from a converged pre-trained generator $f_{\hat{\theta}}$ by updating the parameter $\hat{\theta} \rightarrow \theta^-$, where θ^- represents the updated parameters obtained after unlearning.

Proposed method Given a generative model that models the distribution of training data $p(D)$, a successful unlearned model that unlearns D_f would be what approximates $p(D_r)$, the distribution of D_r, as if it had never seen D_f. The only case where the unlearned model generates samples similar to $x \in D_f$ is when $p(D_f)$ and $p(D_r)$ happen to be very close from the beginning. Under this goal, a straight-forward objective given the pre-trained model approximating $p(D)$ is to make the output of generation to *deviate from* $p(D_f)$, which could be simply formulated as the following:

$$\max_{\theta} \mathbb{E}_{(x,y)\sim D_f} \mathcal{L}(\theta, x, y) \tag{1}$$

where $\mathcal{L}$ denotes training loss (*e.g.* reconstruction loss). Meanwhile, assume we could *define* the influence of a single data on the weight parameter and generation result. Then, unlearning this data would be by simply updating the weight parameter in a direction of removing the data influence. Toward this, we start with defining the data influence on weight parameters and approximates to feasible form as introduced in Koh & Liang (2017):

Definition 2.1. Given upweighting z by some small ϵ and the new parameters $\hat{\theta}_{\epsilon,z} \overset{\text{def}}{=} \text{argmin}_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(z_i, \theta) + \epsilon\mathcal{L}(z, \theta)$, the influence of upweighting z on the parameter $\hat{\theta}$ is given by

$$I_{up,param}(z) \overset{\text{def}}{=} \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\Big|_{\epsilon=0} \overset{\text{def}}{=} -H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}) \tag{2}$$

where $H_{\hat{\theta}} = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 L(z_i, \hat{\theta})$ is the Hessian and is positive definite (PD) by assumption.

By forming a quadratic approximation to the empirical risk around $\hat{\theta}$, a data influence on the weight parameter is formulated as a single Newtons step (See details in Appendix of (Koh & Liang, 2017)), which is consistent with the objective we have mentioned in Equation 1. Although numerous works have verified that this data influence-based approach works well in shallow, discriminative models (Guo et al., 2020; Golatkar et al., 2020a;b), we cannot apply this directly to our generative model due to intractable computation and lack of guarantees on bounds. To address this problem, we re-purpose our objective to minimize the **data influence on generation**. Grounded by recent works (Basu et al., 2020; Sun et al., 2023), we find that we could enjoy this on generative model simply by diminishing the gradient conflict as follows:

Theorem 2.2. *Reducing the influence of samples $z \in D_f$ in training data with regard to test loss is formulated as:*

$$I'_{up,loss}(D_f, z') \rightarrow 0, \tag{3}$$

which is equivalent to

$$\nabla_\theta \mathcal{L}(z', \hat{\theta})^T \sum_{z\in D_f} \nabla_\theta \mathcal{L}(z, \hat{\theta}) \rightarrow 0 \tag{4}$$

where $z' \in D_r$ in our scenario.

Informally, we could achieve this by alleviating the conflict between two gradients $\nabla_\theta \mathcal{L}(z', \hat{\theta})$ and $\nabla_\theta \mathcal{L}(z, \hat{\theta})$, resulting in diminishing the inner product of two gradients. This reminds us of a classic approach of gradient manipulation techniques for conflicting gradients in multi-task learning scenario (Yu et al., 2020; Liu et al., 2021a; Guangyuan et al.). Specifically, we project a gradient of forget sample $x_f \in D_f$ onto normal plane of a set of retain samples $x_r \in D_r$ to meet $\mathcal{I}_{up,loss}(x_f, x_r) = 0$. This orthogonal projection manipulates the original gradient of forget sample $\mathbf{g}_f = \nabla\mathcal{L}_f$ to the weight parameter to which sufficiently unlearns a sample $x_f \in D_f$: $\mathbf{g}_f = \mathbf{g}_f - \frac{\mathbf{g}_f \cdot \mathbf{g}_r}{\|\mathbf{g}_r^2\|}\mathbf{g}_r$. Then, the unlearned model θ^- is obtained after the following gradient update: $\theta^- = \hat{\theta} - \eta\mathbf{g}_f$.

3. Experiments

We verify our idea under numerous data removal requests. Note that measuring and evaluating a generative model to unlearn *a single data* is non-trivial. Even comparing pre-trained generative models trained *with* a particular data over *without* simply by looking at the output of training (*e.g.* generated image, weight) is intractable in case of a deep generative model to the best of our knowledge (van den Burg & Williams, 2021). To make the problem verifiable, in this work, we experiment to unlearn a group of samples sharing similar statistics in the training data - either belonging to a particular class or that has a distinctive semantic feature. In this case, one can evaluate the output of the generation by measuring the number of samples including that class or a semantic feature; a successfully unlearned model would generate nearly zero number of samples having these features. Although we are not able to cover unlearning a single data in this work, note that in essence, our method could successfully approximate the generative model trained without a single data seamlessly, and we look forward to exploring and adjusting a feasible evaluation on this scenario in the near future.

3.1. Experimental Setup

Scenarios We unlearn either a whole class or some notable feature from a group of samples. In the experiment, we use a subset of MNIST (Alsaafin & Elnagar, 2017) with samples

Table 1. Performance of Class/Feature Unlearning VAE on MNIST138 (*left columns*) and CelebA (*right column*) Each experiments are three times repeated. (*) indicates erroneous evaluation by a pre-trained feature classifier. **Bold** indicates the best score.

Metric	MNIST138(Class: 1)				CelebA(Feature: Male)			
	Privacy	Utility		Cost	Privacy	Utility		Cost
	fratio(↓)	*IS*(↑)	*FID*(↓)	*Time*(s)(↓)	*fratio*(↓)	*IS*(↑)	*FID*(↓)	*Time*(s)(↓)
Before	0.343(0.027)	2.053(0.029)	0.030(0.003)	218.6	0.394(0.119)	1.812(0.044)	29.81(0.341)	3×10^4
Grad.Ascnt.	0.264(0.141)	2.029(0.018)	0.127(0.059)	**1.010**	- (*)	**1.311**(0.076)	30.93(1.215)	**97.31**
Moon et al. (2023)	0.344(0.019)	2.048(0.021)	**0.031**(0.002)	166.2	1.000(0.000)	1.000(0.000)	**15.81**(9.831)	8×10^4
Ours	**0.153**(0.057)	**2.192**(0.076)	0.092(0.030)	13.12	**0.150**(0.098)	1.254(0.013)	34.24(0.698)	613.2

of classes 1,3,8 and 64x64 CelebA (Liu et al., 2015) to train and unlearn vanilla VAE (Kingma & Welling, 2013).

Evaluation We evaluate our method under the following three criteria: a privacy guarantee, utility guarantee, and cost. Privacy guarantee includes feature ratio (*fratio*), a ratio of images including the target feature (See details in Appendix A). Utility guarantee includes Frechet Inception Distance (*FID*), a widely used measure for generation quality. Cost includes a total execution time (*Time*) which should be shorter than retrain-from-scratch. A successfully unlearned model would show near-zero on feature ratio, the same IS, FID score as the initial pre-trained model (BEFORE), and the lowest possible execution time. Given the legal impact and the goal of unlearning, note that guaranteeing privacy is prioritized the highest.

Figure 1. Unlearning groups of class 1 samples from VAE pre-trained on MNIST138 (*left*: original, *right*: unlearned) Note that images of class 1 do not appear in generation result.

3.2. Result on Pre-trained Generative Model

Quantitative Result We run the proposed method on pre-trained VAE to remove unlearning group D_f (*e.g.* class 1 or male, respectively) and evaluate them as follows (Table 3) Starting from the pre-trained model (BEFORE) our method unlearns the target D_f with a large decrease on *fratio* by 65% to 70% while keeping the time cost of unlearning $\leq 5\%$ of retrain-from-scratch. All the while, our method still keeps a decent utility performance. Comparing the baselines, our method shows the best in privacy - the prioritized metric - through all experiments. Note that the feature ratio of gradient ascent in the CelebA experiment (feature ratio-CelebA-Grad.Ascnt) was omitted because the generated samples are turned out to be noisy images and thus the evaluation result of pre-trained classifier cannot be accepted. Also, note that although baselines show better performance in terms of utility and cost, they don't show near-best score on privacy guarantee.

Qualitative Result We further validate our method by comparing the generated images before and after the proposed unlearning algorithm. As in Figure 3.1, no class 1 samples are observed after unlearning class 1, meaning that our method successfully meets the request of unlearning class 1, which aligns with the quantitative result where the ratio of samples with class 1 is reduced from 34.3% to $\leq$ 15% as in Table 3. The output of image generation is fair where 3 and 8 are decently distinguishable through one's eyes, although it is certain that some examples show some minor damaged features, which are in the same line as a decrease in IS and an increase in FID score. Note that the ultimate goal of unlearning is to meet the privacy guarantee while preserving the utility of pre-training, which are remained as our next future work.

4. Conclusion

In this work, we introduce a novel theoretically sounded unlearning method for the generative method. Inspired by the influence of the sample on the others, we suggest a simple and effective gradient surgery to unlearn a given set of samples on a pre-trained generative model and outperform the existing baselines. Although we don't experiment to unlearn single data due to a lack of ground evaluation on the uniqueness of the particular data, we leave it as future work emphasizing that our method could also be applied to this scenario. Furthermore, it would be interesting to verify our ideas on various privacy-sensitive datasets. Nonetheless, our work implies the possibility of unlearning a pre-trained generative model, laying the groundwork for privacy handling in generative AI.

References

Alsaafin, A. and Elnagar, A. A minimal subset of features using feature selection for handwritten digit recognition. *Journal of Intelligent Learning Systems and Applications*, 9(4):55–68, 2017.

Basu, S., You, X., and Feizi, S. On second-order group influence functions for black-box predictions. In *International Conference on Machine Learning*, pp. 715–724. PMLR, 2020.

Bishop, C. Exact calculation of the hessian matrix for the multilayer perceptron, 1992.

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159. IEEE, 2021.

Child, R. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

Fu, S., He, F., and Tao, D. Knowledge removal in sampling-based bayesian inference. *arXiv preprint arXiv:2203.12964*, 2022.

Golatkar, A., Achille, A., and Soatto, S. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312, 2020a.

Golatkar, A., Achille, A., and Soatto, S. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pp. 383–398. Springer, 2020b.

Golatkar, A., Achille, A., Ravichandran, A., Polito, M., and Soatto, S. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 792–801, 2021.

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., and Shet, V. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

Guangyuan, S., Li, Q., Zhang, W., Chen, J., and Wu, X.-M. Recon: Reducing conflicting gradients from the root for multi-task learning. In *The Eleventh International Conference on Learning Representations*.

Guo, C., Goldstein, T., Hannun, A., and Van Der Maaten, L. Certified data removal from machine learning models. In *International Conference on Machine Learning*, pp. 3832–3842. PMLR, 2020.

Gupta, V., Jung, C., Neel, S., Roth, A., Sharifi-Malvajerdi, S., and Waites, C. Adaptive machine unlearning. *Advances in Neural Information Processing Systems*, 34: 16319–16330, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Liu, B., Liu, X., Jin, X., Stone, P., and Liu, Q. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021a.

Liu, B., Liu, Q., and Stone, P. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pp. 243–254. PMLR, 2022.

Liu, G., Ma, X., Yang, Y., Wang, C., and Liu, J. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pp. 1–10. IEEE, 2021b.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Moon, S., Cho, S., and Kim, D. Feature unlearning for generative models via implicit feedback. *arXiv preprint arXiv:2303.05699*, 2023.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Sun, Z., Mu, Y., and Hua, G. Regularizing second-order influences for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20166–20175, 2023.

van den Burg, G. and Williams, C. On memorization in probabilistic deep generative models. *Advances in Neural Information Processing Systems*, 34:27916–27928, 2021.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33: 5824–5836, 2020.

Zhang, Z., Zhou, Y., Zhao, X., Che, T., and Lyu, L. Prompt certified machine unlearning with randomized gradient smoothing and quantization. *Advances in Neural Information Processing Systems*, 35:13433–13455, 2022.

A. Experimental Details

A.1. Setup

Architecture In this experiment, we use vanilla VAE (Kingma & Welling, 2013) with encoders of either stack of linear(for MNIST experiment) or convolutional(for CelebA experiment) layers. Although we verify our result on VAE, note that our method can be applied to any variational inference based generative model such as (Kingma et al., 2021; Higgins et al., 2017).

Baseline We compare our experimental results with the following two baselines. One is a recently published, first and the only unlearning work on generative model (Moon et al., 2023) (*FU*) to unlearn by feeding a surrogate model with projected latent vectors. We reproduce FU and follow the hyperparameter details (*e.g.* unlearning epochs 200 for MNIST) as in the original paper. The other is a straight-forward baseline (*Grad.Ascnt.*) which updates the gradient in a direction of maximizing the reconstruction loss on forget, which is equivalent to meeting *e.g.* Objective 1 without gradient surgery. Note that we keep the same step size when unlearning with these three different methods (including ours) for fair comparison.

Training details We use Adam optimizer with learning rate 5e-04 for MNIST experiment and 1e-05 for CelebA experiment. We update the parameter only once (1 epoch) for removals, thus named our title 'one-shot unlearning'. All experiments are three times repeated.

A.2. How to Evaluate Feature Ratio

We first prepare a classification model that classifies the image having a target feature from the remains. In order to obtain a highly accurate classifier, we search for the best classifier which shows over 95% accuracy. In the experiment, we use AllCNN (Springenberg et al., 2014) to classify class 1 over the other in MNIST with 1,3,8 (MNIST381), and ResNet18 (He et al., 2016) to classify male over female on CelebA. After unlearning, we generate 10000 samples from the generator and feed the sample to the pre-trained classifier. Assuming that the classifier classifies the image well, the prediction result would the probability that the generated output contains the features to be unlearned.

B. Definitions and Proof for Theoretical Analysis

In Koh & Liang (2017) and Basu et al. (2020), an influence of sample z on weight parameter is defined as the product of its gradient and inverse of hessian. Moreover, an influence of sample z to *test loss* of sample z' defined in as following:

Definition B.1. (Equation 2 from Koh & Liang (2017)) Suppose up-weighting a converged parameter $\hat{\theta}$ by small ϵ, which gives us new parameters $\hat{\theta}_{\epsilon,z} \overset{\text{def}}{=} \text{argmin}_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(z_i,\theta) + \epsilon\mathcal{L}(z,\theta)$. The influence of up-weighting z on the loss at an arbitrary point z' against has a closed-form expression:

$$\begin{aligned} \mathcal{I}_{up,loss}(z,z') &\overset{\text{def}}{=} \left.\frac{d\mathcal{L}(z',\hat{\theta}_{\epsilon,z})}{d\epsilon}\right|_{\epsilon=0} \\ &= \nabla_\theta \mathcal{L}(z',\hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta \mathcal{L}(z,\hat{\theta}) \end{aligned} \tag{5}$$

where $H_{\hat{\theta}} \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 \mathcal{L}(z_i,\hat{\theta})$ is the Hessian and is positive definite (PD) by assumption on convex and Lipschitz continuity of loss $\mathcal{L}$.

Theorem B.2. *(Theorem 2.2 from Section 2) Reducing the influence of samples $z \in D_f$ in training data with regard to test loss is formulated as:*

$$I'_{up,loss}(D_f, z') \to 0, \tag{6}$$

which is equivalent to

$$\nabla_\theta \mathcal{L}(z',\hat{\theta})^T \sum_{z\in D_f} \nabla_\theta \mathcal{L}(z,\hat{\theta}) \to 0 \tag{7}$$

where $z' \in D_r$ in our scenario.

Proof. The second-order influence of D_f, $\mathcal{I}^{(2)}_{up,param}$, is formulated as sum of first-order influence $\mathcal{I}^{(1)}_{up,param}$ and $\mathcal{I}^{'}_{up,param}$, which captures the dependency of the terms in $\mathcal{O}(\epsilon^2)$ on the group influence is defined as following:

$$\mathcal{I}^{'}_{up,param}(D_f, z') = \mathcal{A}H^{-1}_{\hat{\theta}} \sum_{z \in D_f} \nabla_\theta \mathcal{L}(z, \hat{\theta}) \tag{8}$$

where $\mathcal{A} = \frac{p}{1-p}(I - (\nabla^2 L(\theta^*))^{-1} \frac{1}{|\mathcal{U}|} \sum_{z \in \mathcal{U}} \nabla^2 l(h_{\theta^*}(z)))$ (from Basu et al. (2020)).

The influence of samples in D_f on the test loss of z' can be formulated as:

$$\mathcal{I}_{up,loss}(D_f, z') = \nabla_\theta \mathcal{L}(z, \hat{\theta})^T \mathcal{I}_{up,param}(D_f) \tag{9}$$

which can be equivalently applied to all orders of $\mathcal{I}$ including $\mathcal{I}^{(1)}, \mathcal{I}^{(2)}, \mathcal{I}^{'}$.

Then, $\mathcal{I}^{'}_{up,loss}(D_f, z') = 0$ is now reduced to

$$\nabla_\theta \mathcal{L}(z, \hat{\theta})^T \mathcal{A}H^{-1}_{\hat{\theta}} \sum_{z \in D_f} \nabla_\theta \mathcal{L}(z, \hat{\theta}) = 0 \tag{10}$$

which satisfies the right-hand side of Theorem 2.2 where $\mathcal{A}$ and $H^{-1}_{\hat{\theta}}$ are negligible. □