**Novelty:** We study the problem of learning differential graphs from functional data compared to existing literature that focused on scalar data. While some of the tools we used exist in the literature, we carefully needed to improve a number of these tools to tackle challenges that arise from the infinite-dimensional functional data observed at large number of vertices. For example, while fPCA has been used in [12], they made a strong assumption that there is finite number of nonzero eigenvalues (Condition 2 in [12]), which restricts functional data to lie in a finite-dimensional space. We relax this assumption — see Assumption 3.2 and Theorem 3.1 — thus handling the truly infinite dimensional problem. Moreover, we carefully analyze the bias due to dimensionality reduction, so that we can gain consistency in a high-dimensional setting with sparsity imposed only on the differential graph, rather than separate graphs.

**Loss function:** The design of the loss function $L(\Delta)$ in equation (2.7) is based on [9], where in order to construct a consistent M-estimator, we want the true parameter value $\Delta^M$ to minimize the population loss $\mathbb{E}\left[L(\Delta)\right]$. For a differentiable and convex loss function, this is equivalent to selecting $L$ such that $\mathbb{E}\left[\nabla L(\Delta^M)\right] = 0$. Since $\Delta^M = \left(\Sigma^{X,M}\right)^{-1} - \left(\Sigma^{Y,M}\right)^{-1}$, it satisfies $\Sigma^{X,M}\Delta^M\Sigma^{Y,M} - (\Sigma^{Y,M} - \Sigma^{X,M}) = 0$. By this observation, a choice for $\nabla L(\Delta)$ is $\nabla L(\Delta) = S^{X,M}\Delta S^{Y,M} - (S^{Y,M} - S^{X,M})$ so that $\mathbb{E}\left[\nabla L(\Delta^M)\right] = \Sigma^{X,M}\Delta^M\Sigma^{Y,M} - (\Sigma^{Y,M} - \Sigma^{X,M}) = 0$, and from this choice of $\nabla L(\Delta)$, we get $L(\Delta)$ in equation (2.7) by using properties of differential of trace function. The chosen loss is quadratic (see equation (B.10) in supplement) and leads to an efficient algorithm. Such loss has been used in [15, 17], see also [21]. We will provide additional discussion in the final version.

**R2: 1.** Due to space constraints, we have only included necessary background related to presentation of our ideas. We will include additional background in the supplement of the final version. **2.** Differential graphs have been studied in a scalar setting, while there is no existing literature for the functional data setting. This makes finding a reasonable competitor difficult. Nonetheless we include two competitors. The first competitor is based on [12] — which proposes a method for estimating a single functional graph — and estimates two separate graphs before finding the differential graph. As pointed in reviews and also in [12], choosing tuning parameters for this method is challenging. Here we adopt a heuristic AIC and BIC criterion. Since each edge corresponds to an $M \times M$ block in precision matrix, the degrees of freedom is defined as the the number of edges included in the graph times $M^2$. One advantage of the FuDGE procedure is that we do not need to select multiple hyperparameters but can select a single penalty parameter for the entire problem. In addition, when both underlying graphs are dense, but the difference graph is sparse, the direct difference estimation can perform much better than the separate estimation procedure. The second competitor is to ignore the functional nature of data and directly apply scalar differential graph estimators at various fixed time points to obtain several differential graphs. The final differential graph is then an aggregate of the multiple graphs; we choose majority vote. There are at least two problems with this approach. First, in order to implement this approach, we have to have curves X and Y observed at same time points, and all sample curves need to share the same observation time grids. Both of these requirements make this competitor applicable only in simulations. Second, even if the above requirements are satisfied, there is loss of information by ignoring the functional nature of data. This is demonstrated in simulations. Thus, preserving the entire functional nature of the data is critical for graph estimation. Of course, if we know in advance that the structural information is completely captured at each time point, this procedure estimates fewer parameters and can perform better, as we considered in Supplement D. Many differential estimators can be used for estimation from one time point and here we compare against the state-of-the-art. Our main argument is to stress the strength of preserving the functional nature of the data, while different methods for scalar graphs should not make much difference. This is also related to the discussion of the novelty of the procedure in that our proposed procedure can outperform more naive procedures in the settings for which it is designed. **4.** After performing fPCA, the proximal gradient descent method converges in $O\left(L/\epsilon\right)$ iterations, where $L$ is the largest eigenvalue of $S^{X,M} \otimes S^{Y,M}$ and $\epsilon$ is error tolerance. Each iteration takes $O((pM)^3)$ operations. We will include complexity analysis in final version. **5.** We consider data of a similar size as in [21]. The differential graph we are estimating is represented by $(pM) \times (pM)$ matrix, which contains many more parameters than sample size. Moreover, we are dealing functional data, which is intrinsically an infinite-dimensional object. **6.** Please see point 2. In the final version, we will include for EEG data out of sample prediction and compare against methods that learn precision matrices separately. **7.** Note that the three models tried in the simulations actually represent very different structures in the true underlying graphs. Model 1 is a generally sparse model, but there are hub nodes, which means that some nodes can be densely connected. Model 2 is a very sparse model while model 3 is a very dense model. The differential structures are all sparse, since we are trying to demonstrate that one benefit of direct estimation is that we only require sparsity in differential graph rather than individual graphs.

**R3:** Please see the top of the response for discussion of novelty and the loss function. **3.** We have provided only the necessary background for the paper due to space constraints and we will carefully check notation for consistency in the final version.

**R1:** The code is available on github and a link will be included in the final version.

**R4:** See 2 and 6 in R2. We will include discussion of scientific findings in the final version, comparing to [12].