We would like to thank the reviewers for their overall positive assessment of our work, and for their detailed comments.

Reviewer 1: SparseMap as an appropriate baseline We agree. We will add it to sec. 5.1 and 5.2, similar to Figure R1 below. A potential drawback compared to our method is the possibility of zero gradients. $O(|E|)$ time complexity for shortest path You are correct, this requires integer costs. For general costs, we will replace it with $O(|V|^2)$, for Dijkstra's algorithm. benefits compared to more general frameworks [...] further explained. We will detail further the benefits: guarantee of non-zero gradients and ease of implementation (simple averages of solutions from the same solver).

Reviewer 2: On the tasks of gradient estimation (GE) and loss functions (LF), and additional references suggested Thank you for the additional references, we will include them with discussion. We tried our best to distinguish GE from LF throughout the paper but we will further clarify, following this perspective. We consider this work to provide advances for both of these tasks: Our perturbed optimizers can be used as intermediate layers, with any loss, and are well-defined without requiring a loss (GE); they also naturally yield a FY loss (LF) at the last layer, for which gradient estimation is even easier. We focused mostly in our experiments on cases showcasing both our contributions, with perturbed optimizers being used in a FY loss, but not exclusively, as you point out. We will also include (Fig R1) (in addition to Figure 3), showing that on the 21 dataset experiment, for a given loss (ℓ_2), our GE method with perturbation is on par or better than the GE method of Vlastelica et al. in terms of performance, while there is consistent performance between ℓ_2 and FY losses using our GE method. about "blackbox loss of Vlastelica. et al" but *there is no such thing* We agree, and will change the designation to "Gradient estimation for loss (eqref)." comparing the YF-loss to Vlastelica. et al. is a bit of apples to oranges (LF) to (GE). We argue that it is fair to compare GE and LF if the end task is the same, especially when replicating an experiment. However, we agree that comparing different GE methods that are generic is interesting and will add Figure R1 in this respect. comparison of YF-loss to the classical max-margin loss We will also include it in experiments The curves reported in Figure 6 are *wildly* inconsistent [...] 1) The labels for bb-loss and squared loss are switched The labels were indeed accidentally swapped between bb-loss and squared loss in the left part of Figure 6, thanks for catching it. We will correct this. 2) The metric reported is (slightly) different to the one Vlastelica et al. Following this paper, we reported the proportion of paths with exactly "the optimal costs" with all methods. In order to display the evolution of this metric (and cost ratio to optimal) along epochs, we ran an experiment using the code provided by the authors, with the same parameters. We report directly those figures. We found that another statistic "below 1% accuracy" was close to the numbers reported in this paper, but this is not one of the metric that we considered in our experiment. value of M for the ranking experiment $M = 10$ in the experiment on perturbed GE for the ℓ_2 loss. Prior work being added Thank you for these references, we will cite them and include some discussion.

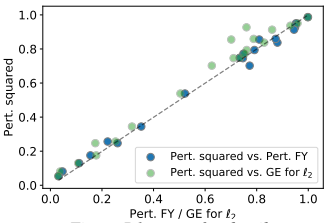

Figure R1: zoom for details

Reviewer 3: running time scales linearly with M? We will add running times and specify the hardware setup. Regarding the dependency in M, when running these on GPU, by the parallelizable nature of our technique, the running time is almost independent of M (up to $M = 1000$). Further, in many cases (as discussed l.210-212, 250-251), the value of M used in learning tasks to minimize the FY loss is very small (of order 1 to 10). Permutation learning with the Sinkhorn algorithm We will include a comparison with this benchmark for the ranking experiments, as in Figure R1. The phrase "the computational graph is broken" on L186 is not quite right. Thank you for pointing it out, we will add this remark.

Reviewer 4: Comments about clarity of the paper Regarding contributions, our propositions are novel, except Prop.3.1, as referenced, and Prop.2.1 (a few lines from classical results). If the paper is accepted, we will use the ninth page to describe section 2 and 3 in more ample details. We made efforts in the abstract and introduction to explain the application to learning tasks, and will use this space to detail it further, early in the paper. Comparison to differentiation of convex programs and finite differences We would like to clarify that we mainly focus on problems where the Jacobian is almost everywhere null. Hence finite-differencing or implicit differentiation of convex functions might not be useful if the function is not smoothed. Our approach provides automatically a smoothing of the function and an unbiased stochastic estimate of the corresponding Jacobian - very easy to implement and fast to compute. It is also very generic and does not require problem-specific adaptation. One can smooth a general convex optimization problem by penalizing the constraints with a log-barrier Thank you for the reference, we will include it. This could raise some algorithmic issues, requiring ad hoc solvers, instead of the existing ones. It is also not always tractable, for problems with efficient solvers but exponentially many facets in the polytope, a barrier approach is impossible. Our method does not suffer from this, and only requires to use the original solver, without a new, often slower method. A notable exception is optimal transport, which we mention, for which the entropic regularization yields an algorithmic advantage compared to the LP. Role of the convex regularizer We will make this point clearer: this formulation leads to many of the important properties of the perturbed optimizer (see Prop 2.2, 2.3, 4.1), including that y^*_ε is one-to-one, implying that estimating θ from the observed model is possible. We will also further address the interesting phenomenon that you describe: the FY loss need not to be computed in order to be minimized. We perceive this as an advantage, since the gradients are very simple to compute, and it performs very well on several learning tasks (see Section 5), but deserves more explanation.