We would like to thank all reviewers for your helpful suggestions.

Response to Reviewer #1

We are sorry about the confusing descriptions in this submission, in the updated version we will check every symbol to make sure they are clear. In what follows we answer your questions to help clarify the notations and experiments.

1. "motivation not clear": G-SSL and specifically label propagation used to be an important semi-supervised learning algorithm and it is still not outdated today: for image/text data we can apply the deep feature to construct a graph and propagate the labels to unlabeled data. `Neo4j`, a industrial level machine learning library even contains a build-in module for label propagation and it has already been applied in social network [Speriosu], Pharmacy [Zhang], etc. Furthermore, G-SSL algorithms are still widely used in social network or recommender systems, where malicious users can easily create fake account or inject fake information into nodes. Lastly, the clean math of G-SSL provides provides a good starting point to generalize adversarial machine learning to other SSL methods, including deep SSL. We will try to include more real-world examples in the revised version.
2. "$\boldsymbol{X}_l$ and $\boldsymbol{X}_u$": following the standard notations, we use $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ to represent the feature matrix for the whole data, where each row of $\boldsymbol{X}$ is a data point. The feature matrix is composed of labeled part and unlabeled part. We simply set $\boldsymbol{X} = [\boldsymbol{X}_l; \boldsymbol{X}_u]$, i.e. the first n_l rows of $\boldsymbol{X}$ are labeled data and the following n_u rows are unlabeled data, and $n = n_l + n_u$. With this in mind, it might be easier to understand $\boldsymbol{X}_l + \Delta_x$—this is the feature of labeled data after perturbation by matrix $\Delta_x \in \mathbb{R}^{n_l\times d}$, which means the feature perturbation can be done to all labeled instances. Moreover, by applying group sparsity constraints on Δ_x (consider each row as a group), our algorithm can conduct perturbation only to a small fraction of instances to greatly change the results.
3. "subscriptions i, j": each row of $\boldsymbol{X}$ is a data point. $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the i-th and j-th row of the matrix $\boldsymbol{X}$ so i and j are actually the index of data ($i, j \in \{1, 2, \ldots, n\}$), and they are iterated within the whole dataset (including both labeled and unlabeled data). However, the constraint in Eq(1) indicates that $\hat{\boldsymbol{y}}$ are fixed for the labeled part, so the free variables in (1) are the unlabeled part of $\hat{\boldsymbol{y}}$.
4. "Datasets not explained": we will add more descriptions of datasets in the next version. Specifically, `mnist` and `rcv1` are widely used datasets to test non-deep learning models such as SVM/logitstic regression/tree based models, `cadata/E2006` are regression datasets for predicting the house price/stock volatility. All of them are publically available at libsvm repository.
5. "Scalability of our algorithm": please note that the computation overhead mainly lies in generating the kernel matrix $\boldsymbol{S}$ in Eq(3), which is $\mathcal{O}(n^2 d)$, computing this matrix for 20,000 nodes is certainly doable. For large data, one can construct a k-NN graph efficiently by efficient similarity search tools, such as Faiss (by Facebook). $\boldsymbol{S}$ will be sparse so the computations will be efficient even for very large datasets.
6. "Dimension of MNIST17 should be 784 rather than 780": 4 in 784 pixels are zero in every image, so we remove these four dimensions in the experiment. Removing them will not affect the final results.

(Speriosu) Twitter polarity classification with label propagation over lexical links and the follower graph, in *EMNLP* 2011.
(Zhang) Label Propagation Prediction of Drug-Drug Interactions Based on Clinical Side Effects, in *Scientific Reports* 2015.

Response to Reviewer #2

Thanks for your feedback! We will correct typos and revise our submission according to your suggestions.

Response to Reviewer #3

Regarding the scope of the paper: The main text of our paper is mainly on poisoning label propagation method. We further show how to poison another widely used SSL method – manifold regularization in Appendix 4.2.

Regarding the datasets: the datasets used in the experiment are all real world datasets (downloaded from libsvm data repository) and are widely used for benchmarking classification and regression models. We will provide more details about these datasets in the revised version of the paper.

Response to Reviewer #4

Regarding the novelty of theory: Our main theoretical contribution lies in the efficient trust region solver in the nonconvex case. To our knowledge, prior work is limited to trust region with cubic regularization [Carmon and Duchi], which is different from our setting (a ℓ_2-norm constraint), and their work cannot be directly applied to our data poisoning problem. Furthermore, earlier trust region solvers (e.g., Dogleg or Steihaug's method) can only get an approximate solution or local minimizer, while our algorithm is guaranteed to return the global minimizer.

Regarding the novelty of algorithm: Finding good "label fliping" operations via probabilistic method is novel. We agree that this method has no theoretical guarantee, however we show empirically that this epsilon-greedy solver successfully avoids some sub-optimal solutions, and beats the greedy method by a significant margin.

Regarding a unified framework: For completeness, we also include manifold regularization method in the appendix. It is certainly preferable to design a framework that include all semi-supervised learning approaches. Regarding to this limitation, we do not intend to solve it completely in this submission and we appeal for follow-up works to give a better answer.

(Carmon and Duchi) Gradient Descent Efficiently Finds the Cubic-Regularized Non-Convex Newton Step, in *ArXiv* 2016.