First, we would like to thank all reviewers for their very positive comments on the theory presented in the paper. This is a theoretical paper indeed, introducing a new concept, mathematically principled and studied. In order to be applicable in practice, we show how to compute it and how to quickly approximate it, with code available (on the anonymous github link provided in the paper). We also check experimentally that our estimator behaves correctly. We believe this is already a nice set of contributions.

In addition, we also propose a certain number of possible uses and extensions of this concept, showing it could be useful in many different ways. We consider it is out of the scope of this paper to actually run such applications, which would be difficult to include in the paper anyway for space reasons without sacrificing ideas in the theoretical section.

However, we do understand the main criticisms about: (a) the lack of insights brought by the large-scale experiment on remote sensing image registration in Section 6, and (b) the lack of comparison to the *perceptual loss*. For (b), we propose to add such a comparison on nearest neighbor retrieval in Section 6. We notice that the *perceptual loss* sometimes performs reasonably well, but often not. For instance, we show below the closest neighbors to a structured residential area image, for the *perceptual loss* (first row: does not make sense) and for our similarity measure (second row: similar areas).

To tackle (a), we propose to show how the similarity experimental computations in Section 6 can be used to **solve the initial problem**, by explicitly turning similarity statistics into a **quantification of the self-denoising effect**, as follows. Let us denote by y_i the true (unknown) label for input $\mathbf{x}_i$, by $\widetilde{y}_i$ the noisy label given in the dataset, and by $\widehat{y}_i = f_\theta(\mathbf{x}_i)$ the label predicted by the network. We will denote the (unknown) noise by $\varepsilon_i = \widetilde{y}_i - y_i$ and assume it is centered and i.i.d., with finite variance σ_ε. The training criterion is $E(\theta) = \sum_j ||\widehat{y}_j - \widetilde{y}_j||^2$. At convergence, the training leads to a local optimum of the energy landscape: $\nabla_\theta E = 0$, that is, $\sum_j (\widehat{y}_j - \widetilde{y}_j) \nabla_\theta \widehat{y}_j = 0$. Let's choose any sample i and multiply by $\nabla_\theta \widehat{y}_i$: using $k^I_\theta(\mathbf{x}_i, \mathbf{x}_j) = \nabla_\theta \widehat{y}_i . \nabla_\theta \widehat{y}_j$, we get: $\sum_j (\widehat{y}_j - \widetilde{y}_j)\, k^I_\theta(\mathbf{x}_j, \mathbf{x}_i) = 0$.

Let us denote by $k^{IN}_\theta(\mathbf{x}_j, \mathbf{x}_i) = k^I_\theta(\mathbf{x}_j, \mathbf{x}_i) \big(\sum_j k^I_\theta(\mathbf{x}_j, \mathbf{x}_i) \big)^{-1}$ the normalized kernel, and by $\mathbb{E}_k[a] = \sum_j a_j\, k^{IN}_\theta(\mathbf{x}_j, \mathbf{x}_i)$ the mean of value a in the neighborhood of i, that is, the weighted average of the a_j with weights $k^I_\theta(\mathbf{x}_j, \mathbf{x}_i)$ normalized to sum up to 1. This is actually a Parzen window estimator. Then the previous property can be rewritten as $\mathbb{E}_k[\widehat{y}] = \mathbb{E}_k[\widetilde{y}]$. As $\mathbb{E}_k[\widetilde{y}] = \mathbb{E}_k[y] + \mathbb{E}_k[\varepsilon]$, this yields: $\widehat{y}_i - \mathbb{E}_k[y] = \mathbb{E}_k[\varepsilon] + (\widehat{y}_i - \mathbb{E}_k[\widehat{y}])$

i.e. the difference between the predicted $\widehat{y}_i$ and the average of the **true labels** in the neighborhood of i is equal to the average of the noise in the neighborhood of i, up to the deviation of the prediction $\widehat{y}_i$ from the average prediction in its neighborhood. **We want to bound the error** $\|\widehat{y}_i - \mathbb{E}_k[y]\|$ **without knowing** neither **the true labels** y nor the noise ε. One can show that $\mathbb{E}_k[\varepsilon] \propto \mathrm{var}_\varepsilon(\mathbb{E}_k[\varepsilon])^{1/2} = \sigma_\varepsilon \, \|k^{IN}_\theta\|_{L2}$. The **denoising factor** is thus $\|k^{IN}_\theta\|_{L2}$, which is between $1/\sqrt{N}$ and 1, depending on the neighborhood quality. It is $1/\sqrt{N}$ when all N data points are identical, i.e. all satisfying $k^C_\theta(\mathbf{x}_i, \mathbf{x}_j) = 1$. On the other extreme, this factor is 1 when all points are independent: $k^I_\theta(\mathbf{x}_i, \mathbf{x}_j) = 0 \quad \forall i \neq j$. This way we extend *noise2noise*[11] to real datasets with non-identical inputs. **In our remote sensing experiment**, we estimate this way a denoising factor of 0.02, consistent across all training rounds and inputs ($\pm 10\%$), implying that each training round contributed equally to denoising the labels. This is confirmed by Fig. 2, which shows the error steadily decreasing, on a control test where true labels are known. The shift $(\widehat{y}_i - \mathbb{E}_k[\widehat{y}])$ on the other hand can be directly estimated given the network prediction. In our case, it is 4.4px on average, which is close to the observed median error for the last round in Fig. 2. It is largely input-dependent, with variance 3.2px, which is reflected by the spread distribution of errors in Fig. 2. This input-dependent shift thus provides a hint about prediction reliability.

It is also possible to bound $(\widehat{y}_i - \mathbb{E}_k[\widehat{y}]) = \mathbb{E}_k[\widehat{y}_i - \widehat{y}]$ using only similarity information (without predictions $\widehat{y}$). **Theorem 1** implies that the application: $\frac{\nabla_\theta f_\theta(\mathbf{x})}{\|\nabla_\theta f_\theta(\mathbf{x})\|} \mapsto f_\theta(\mathbf{x})$ is well-defined, and it can actually be shown to be differentiable and Lipschitz with a network-dependent constant. Thus $\|f_\theta(\mathbf{x}) - f_\theta(\mathbf{x}')\| \leqslant C \left\| \frac{\nabla_\theta f_\theta(\mathbf{x})}{\|\nabla_\theta f_\theta(\mathbf{x})\|} - \frac{\nabla_\theta f_\theta(\mathbf{x}')}{\|\nabla_\theta f_\theta(\mathbf{x}')\|} \right\| = \sqrt{2} C \sqrt{1 - k^C_\theta(\mathbf{x}, \mathbf{x}')}$, yielding $\|\widehat{y}_i - \widehat{y}_j\| \leqslant \sqrt{2} C \sqrt{1 - k^C_\theta(\mathbf{x}_i, \mathbf{x}_j)}$ and thus $\mathbb{E}_k[\widehat{y}_i - \widehat{y}] \leqslant \sqrt{2} C \, \mathbb{E}_k \Big[\sqrt{1 - k^C_\theta(\mathbf{x}_i, \cdot)} \Big]$.

Other: thank you for the very relevant literature, and the nice application suggestion to GAN / cycle-consistency. We can postpone the paragraph *Dynamics of learning* to the appendix to make place for the section above if needed.