DON’T FEAR THE UNLABELLED:
SAFE SEMI-SUPERVISED LEARNING VIA DEBIASING

Hugo Schmutz∗† Olivier Humbert†‡ Pierre-Alexandre Mattei∗
hugo.schmutz@inria.fr

ABSTRACT

Semi-supervised learning (SSL) provides an effective means of leveraging unlabelled data to improve a model’s performance. Even though the domain has received a considerable amount of attention in the past years, most methods present the common drawback of lacking theoretical guarantees. Our starting point is to notice that the estimate of the risk that most discriminative SSL methods minimise is biased, even asymptotically. This bias impedes the use of standard statistical learning theory and can hurt empirical performance. We propose a simple way of removing the bias. Our debiasing approach is straightforward to implement and applicable to most deep SSL methods. We provide simple theoretical guarantees on the trustworthiness of these modified methods, without having to rely on the strong assumptions on the data distribution that SSL theory usually requires. In particular, we provide generalisation error bounds for the proposed methods. We evaluate debiased versions of different existing SSL methods, such as the Pseudo-label method and Fixmatch, and show that debiasing can compete with classic deep SSL techniques in various settings by providing better calibrated models. Additionally, we provide a theoretical explanation of the intuition of the popular SSL methods. An implementation of a debiased version of Fixmatch is available at https://github.com/HugoSchmutz/DeFixmatch

1 INTRODUCTION

The promise of semi-supervised learning (SSL) is to be able to learn powerful predictive models using partially labelled data. In turn, this would allow machine learning to be less dependent on the often costly and sometimes dangerously biased task of labelling data. Early SSL approaches—e.g. Scudder’s (1965) untaught pattern recognition machine—simply replaced unknown labels with predictions made by some estimate of the predictive model and used the obtained pseudo-labels to refine their initial estimate. Other more complex branches of SSL have been explored since, notably using generative models (from McLachlan, 1977 to Kingma et al., 2014) or graphs (notably following Zhu et al., 2003). Deep neural networks, which are state-of-the-art supervised predictors, have been trained successfully using SSL. Somewhat surprisingly, the main ingredient of their success is still the notion of pseudo-labels (or one of its variants), combined with systematic use of data augmentation (e.g. Xie et al., 2019; Sohn et al., 2020; Rizve et al., 2021).

An obvious SSL baseline is simply throwing away the unlabelled data. We will call such a baseline the complete case, following the missing data literature (e.g. Tsiatis, 2006). As reported in van Engelen & Hoos (2020), the main risk of SSL is the potential degradation caused by the introduction of unlabelled data. Indeed, semi-supervised learning outperforms the complete case baseline only in specific cases (Singh et al., 2008; Scholkopf et al., 2012; Li & Zhou, 2014). This degradation risk for generative models has been analysed in Chapelle et al. (2006, Chapter 4). To overcome this issue, previous works introduced the notion of safe semi-supervised learning for techniques which never reduce predictive performance by introducing unlabelled data (Li & Zhou, 2014; Guo et al., 2020). Our loose definition of safeness is as follows: an SSL algorithm is safe if it has theoretical guarantees that are similar or stronger to the complete case baseline. The “theoretical” part of the definition

∗ Université Côte d’Azur, Inria, Maasai, LJAD, CNRS, Nice, France
† Université Côte d’Azur, TIRO-MATOs, UMR E 4320 CEA, Nice, France
‡ Centre Antoine Lacassagne, Nice, France
is motivated by the fact that any empirical assessment of generalisation performances of an SSL algorithm is jeopardised by the scarcity of labels. “Similar or stronger guarantees” can be understood in a broad sense since there are many kinds of theoretical guarantees (e.g. the two methods may be both consistent, have similar generalisation bounds, be both asymptotically normal with related asymptotic variances). Unfortunately, popular deep SSL techniques generally do not benefit from theoretical guarantees without strong and essentially untestable assumptions on the data distribution (Mey & Loog, 2022) such as the smoothness assumption (small perturbations on the features $x$ do not cause large modification in the labels, $p(y|pert(x)) \approx p(y|x)$) or the cluster assumption (data points are distributed on discrete clusters and points in the same cluster are likely to share the same label).

Most semi-supervised methods rely on these distributional assumptions to ensure performance in entropy minimisation, pseudo-labelling and consistency-based methods. However, no proof is given that guarantees the effectiveness of state-of-the-art methods (Tarvainen & Valpola, 2017; Miyato et al., 2018; Sohn et al., 2020; Pham et al., 2021). To illustrate that SSL requires specific assumptions, we show in a toy example that pseudo-labelling can fail. To do so, we draw samples from two uniform distributions with a small overlap. Both supervised and semi-supervised neural networks are trained using the same labelled dataset. While the supervised algorithm learns perfectly the true distribution of $p(1|x)$, the semi-supervised learning methods (both entropy minimisation and pseudo-label) underestimate $p(1|x)$ for $x \in [1, 3]$ (see Figure 1). We also test our proposed method (DeSSL) on this dataset and show that the unbiased version of each SSL technique learns the true distribution accurately. See Appendix A for the results with Entropy Minimisation. Beyond this toy example, a recent benchmark (Wang et al., 2022) of the last SSL methods demonstrates that no one method is empirically better than the others. Therefore, the scarcity of labels brings out the need for competitive methods that benefit from theoretical guarantees. The main motivation of this work is to show that these competitive methods can be easily modified to benefit from theoretical guarantees without performance degradation.

1.1 Contributions

Rather than relying on the strong geometric assumptions usually used in SSL theory, we use the _missing completely at random (MCAR)_ assumption, a standard assumption from the missing data literature (see e.g. Little & Rubin, 2019) and often implicitly made in most SSL works. With this only assumption on the data distribution, we propose a new safe SSL method derived from simply debiasing common SSL risk estimates. Our main contributions are:

- We introduce debiased SSL (DeSSL), a safe method that can be applied to most deep SSL algorithms without assumptions on the data distribution;

- We propose a theoretical explanation of the intuition of popular SSL methods. We provide theoretical guarantees on the safeness of using DeSSL both on consistency, calibration and asymptotic normality. We also provide a generalisation error bound;

- We show how simple it is to apply DeSSL to the most popular methods such as Pseudo-label and Fixmatch, and show empirically that DeSSL leads to models that are never worse than their classical counterparts, generally better calibrated and sometimes much more accurate.
2 SEMI-SUPERVISED LEARNING

The ultimate objective of most of the learning frameworks is to minimise a risk $R$, defined as the expectation of a particular loss function $L$ over a data distribution $p(x, y)$, on a set of models $f_\theta(x)$, parametrised by $\theta \in \Theta$. The distribution $p(x, y)$ being unknown, we generally minimise a Monte Carlo approximation of the risk, the empirical risk $\hat{R}(\theta)$ computed on a sample of $n$ i.i.d points drawn from $p(x, y)$. $\hat{R}(\theta)$ is an unbiased and consistent estimate of $R(\theta)$ under mild assumptions. Its unbiased nature is one of the basic properties that is used for the development of traditional learning theory and asymptotic statistics (van der Vaart 1998; Shalev-Shwartz & Ben-David 2014).

Semi-supervised learning leverages both labelled and unlabelled data to improve the model’s performance and generalisation. Further information on the distribution $p(x)$ provides a better understanding of the distributions $p(x, y)$ and also $p(y|x)$. Indeed, $p(x)$ may contain information on $p(y|x)$ (Schölkopf et al. 2012; Goodfellow et al. 2016, Chapter 7.6, van Engelen & Hoos 2020).

In the following, we have access to $n$ samples drawn from the distribution $p(x, y)$ where some of the labels are missing. We introduce a new binary random variable $r \sim \mathcal{B}(\pi)$ that governs whether or not a data point is labelled ($r = 0$ missing, $r = 1$ observed, and $\pi \in (0, 1)$ is the probability of being labelled). The labelled (respectively the unlabelled) datapoints are indexed by the set $\mathcal{L}$ (respectively $\mathcal{U}$), $\mathcal{L} \cup \mathcal{U} = \{1, \ldots, n\}$. We note $n_l$ the number of labelled and $n_u$ the number of unlabelled datapoints.

The MCAR assumption states that the missingness of a label $y$ is independent of its features and the value of the label: $p(x, y, r) = p(x, y)p(r)$. This is the case when neither features nor labels carry information about the potential missingness of the labels. This description of semi-supervised learning as a missing data problem has already been done in multiple works (see e.g. Seeger 2000; Ahflock & McLachlan 2019). Moreover, the MCAR assumption is implicitly made in most of the SSL works to design the experiments, indeed, missing labels are drawn completely as random in datasets such as MNIST, CIFAR or SVHN (Tarvainen & Valpola 2017; Miyato et al., 2018; Xie et al., 2019; Sohn et al. 2020). Moreover, the definition of safety in the introduction is not straightforward without the MCAR assumption as the complete case is not an unbiased estimator of the risk in non-MCAR settings (see e.g. Liu & Goldberg 2020 Section 3.1).

2.1 COMPLETE CASE: THROWING THE UNLABELLED DATA AWAY

In missing data theory, the complete case is the learning scheme that only uses fully observed instances, namely labelled data. The natural estimator of the risk is then simply the empirical risk computed on the labelled data. Fortunately, in the MCAR setting, the complete case risk estimate keeps the same good properties of the traditional supervised one: it is unbiased and converges point wisely to $\hat{R}(\theta)$. Therefore, traditional learning theory holds for the complete case under MCAR. While these observations are hardly new (see e.g. Liu & Goldberg 2020), they can be seen as particular cases of the theory that we develop below. The risk to minimise is

$$\hat{R}_{CC}(\theta) = \frac{1}{n_l} \sum_{i \in \mathcal{L}} L(\theta; x_i, y_i).$$

(1)

2.2 INCORPORATING UNLABELLED DATA

A major drawback of the complete case framework is that a lot of data ends up not being exploited. A class of SSL approaches, mainly inductive methods with respect to the taxonomy of van Engelen & Hoos 2020, generally aim to minimise a modified estimator of the risk by including unlabelled data. Therefore, the optimisation problem generally becomes finding $\hat{\theta}$ that minimises the SSL risk,

$$\hat{R}_{SSL}(\theta) = \frac{1}{n_l} \sum_{i \in \mathcal{L}} L(\theta; x_i, y_i) + \frac{\lambda}{n_u} \sum_{i \in \mathcal{U}} H(\theta; x_i).$$

(2)

where $H$ is a term that does not depend on the labels and $\lambda$ is a scalar weight which balances the labelled and unlabelled terms. In the literature, $H$ can generally be seen as a surrogate of $L$. Indeed, it looks like the intuitive choices of $H$ are equal or equivalent to a form of expectation of $L$ on a distribution given by the model.
2.3 Some examples of surrogates

A recent overview of the recent SSL techniques has been proposed by van Engelen & Hoos (2020). In this work, we focus on methods suited for a discriminative probabilistic model \( p_\theta(y|x) \) that approximates the conditional \( p(y|x) \). We categorised methods into two distinct sections, entropy and consistency-based.

**Entropy-based methods**  Entropy-based methods aim to minimise a term of entropy of the predictions computed on unlabelled data. Thus, they encourage the model to be confident on unlabelled data, implicitly using the cluster assumption. Entropy-based methods can all be described as an expectation of \( L \) under a distribution \( \pi_x \) computed at the datapoint \( x \):

\[
H(\theta; x) = \mathbb{E}_{\pi_x(\hat{x}, \hat{y})}[L(\theta; \hat{x}, \hat{y})].
\]

For instance, Grandvalet & Bengio (2004) simply use the Shannon entropy as \( H(\theta; x) \) which can be rewritten as equation 3 with \( \pi_x(\hat{x}, \hat{y}) = \delta_{\hat{x}}(\hat{x})p_\theta(y|\hat{x}) \), where \( \delta_{\hat{x}} \) is the dirac distribution in \( \hat{x} \). Also, pseudo-label methods, which consist in picking the class with the maximum predicted probability as a pseudo-label for the unlabelled data (Scudder [1965]), can also be described as Equation 3. See Appendix B for complete description of the entropy-based literature (Berthelot et al., 2019; 2020; Xie et al., 2019; Sohn et al., 2020; Rizve et al., 2021) and further details.

**Consistency-based methods**  Another range of SSL methods minimises a consistency objective that encourages invariant prediction for perturbations either on the data or on the model in order to enforce stability on model predictions. These methods rely on the smoothness assumption. In this category, we cite II-model (Sajjadi et al., 2016), temporal ensembling (Laine & Aila [2017]), Mean-teacher (Tarvainen & Valpola [2017]), virtual adversarial training (VAT, Miyato et al. [2018]) and interpolation consistent training (ICT, Verma et al. [2019]). These objectives \( H \) are equivalent to an expectation of \( L \) (see Appendix B). The general form of the unsupervised objective can be written as

\[
C_1\mathbb{E}_{\pi_x(\hat{x}, \hat{y})}[L(\theta; \hat{x}, \hat{y})] \leq H(\theta; x) = \text{Div}(f_\theta(x, .), pert(f_\theta(x, .))) \leq C_2\mathbb{E}_{\pi_x(\hat{x}, \hat{y})}[L(\theta; \hat{x}, \hat{y})],
\]

where \( f_\theta \) is the predictions of the model, the \( \text{Div} \) is a non-negative function that measures the divergence between two distributions, \( \hat{\theta} \) is a fixed copy of the current parameter \( \theta \) (the gradient is not propagated through \( \hat{\theta} \)), \( pert \) is a perturbation applied to the model or the data and \( 0 \leq C_1 \leq C_2 \).

Previous works also remarked that \( H \) is an expectation of \( L \) for entropy-minimisation and pseudo-label (Zhu et al., 2022) [Aminnian et al., 2022]. We describe a more general framework covering further methods and provide with our theory an intuition on the choice of \( H \).

2.4 Safe semi-supervised learning

The main risk of SSL is the potential degradation caused by the introduction of unlabelled data when distributional assumptions are not satisfied (Singh et al., 2008; Schölkopf et al., 2012; Li & Zhou, 2014), specifically in settings where the MCAR assumption does not hold anymore (Oliver et al., 2018; Guo et al., 2020). Additionally, in Zhu et al. (2022), the authors show disparate impacts of pseudo-labelling on the different sub-classes of the population. As remarked by Oliver et al. (2018), SSL performances are enabled by leveraging large validation sets which is not suited for real-world applications. To mitigate these problems, previous works introduced the notion safe semi-supervised learning for techniques which never reduce learning performance by introducing unlabelled data (Li & Zhou, 2014; Kawakita & Takeuchi [2014], Li et al., 2016; Gan et al., 2017; Trapp et al., 2017; Guo et al., 2020). While the spirit of the definition of safe given by these works is the same, there are different ways of formalising it. To be more precise, we listed in Appendix C the theoretical guarantees given by these different works. In our work, we propose to call safe an SSL algorithm that has theoretical guarantees that are similar to or stronger than those of the complete case baseline.

Even though the methods presented in Section 2.3 produce good performances in a variety of SSL benchmarks, they generally do not benefit from theoretical guarantees, even elementary. Moreover, Schölkopf et al. (2012) identify settings on the causal relationship between the features \( x \) and the target \( y \) where SSL may systematically fail, even if classic SSL assumptions hold. Our example of Figure 1 also shows that classic SSL may fail to generalise in a very benign setting with a large number of labelled data. Presented methods minimise a biased version of the risk under the MCAR...
assumption and therefore classical learning theory cannot be applied anymore, as we argue more precisely in Appendix C.2. Learning over a biased estimate of the risk is not necessarily unsafe but it is difficult to provide theoretical guarantees on such methods even if some works try to do so with strong assumptions on the data distribution (Mey & Loog 2022, Sections 4 and 5). Additionally, the choice of $H$ can be confusing as seen in the literature. For instance, Grandvalet & Bengio (2004) and Corduneanu & Jaakkola (2003) perform respectively entropy and mutual information minimisation whereas Pereyra et al. (2017) and Krause et al. (2010) perform maximisation of the same quantities.

Some other SSL methods have theoretical guarantees, unfortunately, so far these methods come with either strong assumptions or important computational burdens. Li & Zhou (2014) introduced a safe semi-supervised SVM and showed that the accuracy of their method is never worse than SVMs trained with only labelled data with the assumption that the true model is accessible. However, if the distributional assumptions are not satisfied, no improvement or degeneration is expected. Sakai et al. (2017) proposed an unbiased estimate of the risk for binary classification by including unlabelled data. The key idea is to use unlabelled data to better evaluate on the one hand the risk of positive class samples and on the other the risk of negative samples. They provided theoretical guarantees on its variance and a generalisation error bound. The method is designed only for binary classification and has not been tested in a deep-learning setting. It has been extended to ordinal regression in follow-up work (Tsuchiya et al., 2021). In the context of kernel machines, Liu & Goldberg (2020) used an unbiased estimate of risk, like ours, for a specific choice of $H$. Guo et al. (2020) proposed DS$^4$L, a safe method that needs to approximately solve a bi-level optimisation problem. In particular, the method is designed for a different setting, not under the MCAR assumption, where there is a class mismatch between labelled and unlabelled data. The resolution of the optimisation problem provides a solution not worse than the complete case but comes with approximations. They provide a generalisation error bound. Sokolovska et al. (2008) proposed a method with asymptotic guarantees using strong assumptions such that the feature space is finite and the marginal probability distribution of $x$ is fully known. Fox-Roberts & Rosten (2014) proposed an unbiased estimator in the generative setting applicable to a large range of models and they prove that this estimator has a lower variance than the one of the complete case.

3 DeSSL: Unbiased semi-supervised learning

To overcome the issues introduced by the second term in the approximation of the risk for the semi-supervised learning approach, we propose DeSSL, an unbiased version of the SSL estimator using labelled data to annul the bias. The idea here is to retrieve the properties of classical learning theory. Fortunately, we will see that the proposed method can eventually have better properties than the complete case, in particular with regard to the variance of the estimate. The proposed DeSSL objective is

$$\hat{R}_{\text{DeSSL}}(\theta) = \frac{1}{n_l} \sum_{i \in L} L(\theta; x_i, y_i) + \frac{\lambda}{n_u} \sum_{i \in U} H(\theta; x_i) - \frac{\lambda}{n_l} \sum_{i \in L} H(\theta; x_i).$$

(5)

Under the MCAR assumption, this estimator is unbiased for any value of the parameter $\lambda$. For proof of this result see Appendix D. We prove the optimality of debiasing with the labelled dataset in Appendix E. Intuitively, for entropy-based methods, $H$ should be applied only on unlabelled data to enforce the confidence of the model only on unlabelled datapoints. Whereas, for consistency-based methods, $H$ can be applied to any subset of data points. Our theory and proposed method remain the same whether $H$ is applied to all the available data or not (see Appendix K).

3.1 Does the DeSSL risk estimator make sense?

The most intuitive interpretation is that by debiasing the risk estimator, we get back to the basics of learning theory. This way of debiasing is closely related to the method of control variates (Owen, 2013, Chapter 8) which is a common variance reduction technique. The idea is to add an additional term to a Monte-Carlo estimator with a null expectation in order to reduce the variance of the estimator without modifying the expectation. Here, DeSSL can also be interpreted as a control variate on the risk’s gradient itself and should improve the optimisation scheme. This idea is close to the optimisation schemes introduced by Johnson & Zhang (2013) and Defazio et al. (2014) which reduce the variance of the gradients’ estimate to improve optimisation performance. As a matter of fact, we study the gradient’s variance reduction of DeSSL in Appendix E.1.
Another interesting way to interpret DeSSL is as a constrained optimisation problem. Indeed, minimising $\hat{R}_{DeSSL}$ is equivalent to minimising the Lagrangian of the following optimisation problem:

$$\min_{\theta} \quad \hat{R}_{CC}(\theta)$$

$$\text{s.t.} \quad \frac{1}{n_u} \sum_{i \in U} H(\theta; x_i) = \frac{1}{n_l} \sum_{i \in L} H(\theta; x_i).$$

(6)

The idea of this optimisation problem is to minimise the complete case risk estimator by assessing that some properties represented by $H$ are on average equal for the labelled data and the unlabelled data. For example, if we consider entropy-minimisation, this program encourages the model to have the same confidence on the unlabelled examples as on the labelled ones.

The debiasing term of our objective will penalise the confidence of the model on the labelled data. Pereyra et al. [2017] show that penalising the entropy in a supervised context for supervised models improves on the state-of-the-art on common benchmarks. This comforts us in the idea of debiasing using labelled data in the case of entropy-minimisation. Moreover, the debiasing term in pseudo-label is similar to plausibility inference (Barndorff-Nielsen, 1976). Intuitively we understand the benefits of debiasing the estimator with labelled data to penalise the confidence of the model on these datapoints. But the debiasing can be performed on any subset of the training data (labelled or unlabelled). However, in regards to the variance of the estimator, we can prove that debiasing with only the labelled data or the whole dataset are both optimal and equivalent (see Appendix E).

Our objective also resembles doubly-robust risk estimates used for SSL in the context of kernel machines by Liu & Goldberg [2020] and for deep learning by Hu et al. [2022]. In both cases, their focus is quite different, as they consider weaker conditions than MCAR, but very specific choices of $H$.

3.2 **IS $\hat{R}_{DeSSL}(\theta)$ AN ACCURATE RISK ESTIMATE?**

Because of the connections between our debiased estimate and variance reduction techniques, we have a natural interest in the variance of the estimate. Having a lower-variance estimate of the risk would mean estimating it more accurately, leading to better models. Similarly to traditional control variates (Owen, 2013), the variance can be computed, and optimised in $\lambda$:

**Theorem 3.1.** The function $\lambda \mapsto \sqrt{\mathbb{V}(\hat{R}_{DeSSL}(\theta) | r)}$ reaches its minimum for:

$$\lambda_{opt} = \frac{n_u}{n} \frac{\text{Cov}(L(\theta; x, y), H(\theta; x))}{\mathbb{V}(H(\theta; x))},$$

(7)

and at $\lambda_{opt}$:

$$\mathbb{V}(\hat{R}_{DeSSL}(\theta) | r)_{\lambda_{opt}} = \left(1 - \frac{\rho_{L,H}}{n} \right) \mathbb{V}(\hat{R}_{CC}(\theta) | r) \leq \mathbb{V}(\hat{R}_{CC}(\theta) | r),$$

(8)

where $\rho_{L,H} = \text{Covr}(L(\theta; x, y), H(\theta; x))$.

Additionally, we have a variance reduction regime $\sqrt{\mathbb{V}(\hat{R}_{DeSSL}(\theta) | r)} \leq \sqrt{\mathbb{V}(\hat{R}_{CC}(\theta) | r)}$ for all $\lambda$ between 0 and $2\lambda_{opt}$. A proof of this theorem is available as Appendix E. This theorem provides a formal justification to the heuristic idea that $H$ should be a surrogate of $L$. Indeed, DeSSL is a more accurate risk estimate when $H$ is strongly positively correlated with $L$, which is likely to be the case when $H$ is equal or equivalent to an expectation of $L$. Then, choosing $\lambda$ positive is a coherent choice. We demonstrate in Appendix E that $L$ and $H$ are positively correlated when $L$ is the negative likelihood and $H$ is the entropy. Finally, we experimentally validate this theorem and the unbiasedness of our estimator in Appendix E. Other SSL methods have variance reduction guarantees, see Fox-Roberts & Rosten [2014] and Sakai et al. [2017]. In a purely supervised context, Chen et al. [2020] show that the effectiveness of data augmentation techniques lies partially in the variance reduction of the risk estimate. A natural application of this theorem would be to tune $\lambda$ automatically by estimating $\lambda_{opt}$. In our case however, the estimation of $\text{Cov}(L(\theta; x, y), H(\theta; x))$ with few labels led to unsatisfactory results. However, we estimate it more accurately using the test set (which is of course impossible in practice) on different datasets and methods to provide intuition on the order of $\lambda_{opt}$ and the range of the variance reduction regime in Appendix M.2.
3.3 Calibration, Consistency and Asymptotic Normality

The calibration of a model is its capacity of predicting probability estimates that are representative of the true distribution. This property is determinant in real-world applications when we need reliable predictions. In the sense of scoring rules (Gneiting & Raftery, 2007) theory, we prove DeSSL is as well-calibrated as the complete case. See Theorem 4.1 and proof in Appendix G.

We say that \( \hat{\theta} \) is consistent if \( d(\hat{\theta}, \theta^*) \xrightarrow{\mathbb{P}} 0 \) when \( n \to \infty \), where \( d \) is a distance on \( \Theta \). The properties of \( \hat{\theta} \) depend on the behaviours of the functions \( L \) and \( H \). We will thus use the following standard assumptions.

Assumption 3.2. The minimum \( \theta^* \) of \( R \) is well-separated: \( \inf_{\theta \in \Theta : d(\theta, \theta^*) \geq \delta} R(\theta) > R(\theta^*) \).

Assumption 3.3. The uniform weak law of large numbers holds for both \( L \) and \( H \).

Theorem 3.4. Under the MCAR assumption, Assumption 3.2 and Assumption 3.3, \( \hat{\theta} = \arg \min R_{\text{DeSSL}} \) is consistent.

For proof of this theorem see Appendix G. This theorem is a simple application of van der Vaart’s (1998) Theorem 5.7 proving the consistency of an M-estimator. The result holds for the complete case, with \( \lambda = 0 \) which proves that the complete case is a solid baseline under the MCAR assumption.

Coupling of \( n_l \) and \( n_u \) under the MCAR assumption Under the MCAR assumption, \( n_l \) and \( n_u \) are random variables. We have \( r \sim \mathcal{B}(\pi) \) (i.e. any \( x \) has the probability \( \pi \) of being labelled). Then, with \( n \) growing to infinity, we have \( \frac{n_l}{n} = \frac{n_l}{n_l + n_u} \to \pi \). Therefore, both \( n_l \) and \( n_u \) grow to infinity and \( \frac{n_l}{n_u} \to \frac{1}{\pi} \). This implies \( n_u = \mathcal{O}(n_l) \) when \( n \) goes to infinity.

Going further, we prove the asymptotic normality of \( \hat{\theta}_{\text{DeSSL}} \) by simplifying a bit the learning objective. The idea is to replace \( n_l \) with \( \pi n \) to remove the dependence between samples. However, this modification is equivalent to our objective by modifying \( \lambda \). We also show that the asymptotic variance can be optimised with respect to \( \lambda \). We define the cross-covariance matrix between random vectors \( \nabla L(\theta; x, y) \) and \( \nabla H(\theta; x) \) as \( K_\theta(i, j) = \text{Cov}(\nabla L(\theta; x, y)_i, \nabla H(\theta; x)_j) \).

Theorem 3.5. Suppose \( L \) and \( H \) are smooth functions in \( \mathcal{C}^2(\Theta, \mathbb{R}) \). Assume \( R(\theta) \) admits a second-order Taylor expansion at \( \theta^* \) with a non-singular second order derivative \( V_{\theta^*} \). Under the MCAR assumption, \( \hat{\theta}_{\text{DeSSL}} \) is asymptotically normal and the trace of the covariance can be minimised.

Indeed, \( \text{Tr}(\Sigma_{\text{DeSSL}}) \) reaches its minimum at

\[
\lambda_{\text{opt}} = (1 - \pi) \frac{\text{Tr}(V_{\theta^*}^{-1} K_\theta V_{\theta^*}^{-1})}{\text{Tr}(V_{\theta^*}^{-1} E \left[ \nabla H(\theta^*; x) \nabla H(\theta^*; x)^T \right] V_{\theta^*}^{-1})}, \tag{9}
\]

and at \( \lambda_{\text{opt}} \), we have \( \text{Tr}(\Sigma_{\text{DeSSL}}) - \text{Tr}(\Sigma_{\text{CC}}) \leq 0 \).

See Appendix I for proof of this result. The Theorem shows there exists a range of \( \lambda \) for which DeSSL is a better model’s parameters than the complete case. In this context, we have a variance reduction regime on the risk estimate but even more importantly on the parameters estimate.

3.4 Rademacher Complexity and Generalisation Bounds

In this section, we prove an upper bound for the generalisation error of DeSSL. To simplify, we use the same modification as for the asymptotic variance. The unbiasedness of \( \hat{R}_{\text{DeSSL}} \) can directly be used to derive generalisation bounds based on the Rademacher complexity (Bartlett & Mendelson, 2002), defined in our case as

\[
R_n = \mathbb{E}_{(\varepsilon_i)_{i \leq n}} \left[ \sup_{\theta \in \Theta} \left( \frac{1}{n \pi} \sum_{i \leq L} \varepsilon_i L(\theta; x_i, y_i) - \frac{\lambda}{n \pi} \sum_{i \leq L} \varepsilon_i H(\theta; x_i) + \frac{\lambda}{n(1 - \pi)} \sum_{i \leq L} \varepsilon_i H(\theta; x_i) \right) \right], \tag{10}
\]

where \( \varepsilon_i \) are i.i.d. Rademacher variables independent of the data. In the particular case of \( \lambda = 0 \), we recover the standard Rademacher complexity of the complete case.

Theorem 3.6. We assume that labels are MCAR and that both \( L \) and \( H \) are bounded. Then, there exists a constant \( \kappa > 0 \), that depends on \( \lambda \), \( L \), \( H \), and the ratio of observed labels, such that, with probability at least \( 1 - \delta \), for all \( \theta \in \Theta \),

\[
R(\theta) \leq \hat{R}_{\text{DeSSL}}(\theta) + 2R_n + \kappa \sqrt{\frac{\log(4/\delta)}{n}}. \tag{11}
\]
The proof follows Shalev-Shwartz & Ben-David (2014, Chapter 26), and is available in Appendix J.

4 Experiments

We evaluate the performance of DeSSL against different classic methods. In particular, we perform experiments with varying $\lambda$ on different datasets MNIST, DermaMNIST, CIFAR-10, CIFAR-100 (Krizhevsky, 2009) and five small datasets of MedMNIST (Yang et al., 2021, 2023) with a fixed $\lambda$. The results of these experiments are reported below. In our figures, the error bars represent the size of the 95% confidence interval (CI). Finally, we modified the implementation of Fixmatch (Sohn et al., 2020) and compare it with its debiased version on CIFAR-10, CIFAR-100 and on STL10 (Coates et al., 2011). Finally, we show how simple it is to debias an existing implementation, by demonstrating it on the consistency-based models benchmarked by (Oliver et al., 2018), namely VAT, II-model and MeanTeacher on CIFAR-10 and SVHN (Netzer et al., 2011). We observe similar performances between the debiased and biased versions for the different methods, both in terms of cross-entropy and accuracy. See Appendix O.

4.1 Pseudo Label

We compare PseudoLabel and DePseudoLabel on CIFAR-10 and MNIST. We test the influence of the hyperparameters $\lambda$ and report the accuracy, the cross-entropy and the expected calibration error (ECE, Guo et al., 2017) at the epoch of best validation accuracy using 10% of $n_l$ as the validation set.

MNIST is an advantageous dataset for SSL since classes are well-separated. We train a LeNet-like architecture using $n_l = 1000$ labelled data on 10 different splits of the training dataset into a labelled and unlabelled set. Models are then evaluated using the standard 10,000 test samples. Results are reported in Figure 2 and Appendix L. In this example SSL and DeSSL have almost the same accuracy for all $\lambda$, however, DeSSL seems to be always better calibrated.

CIFAR-10 We train a CNN-13 from Tarvainen & Valpola (2017) on 5 different splits. We use $n_l = 4000$ and use the rest of the dataset as unlabelled. Models are then evaluated using the standard test samples. Results are reported in Figure 3. We report the ECE in Appendix M. The performance of both methods on CIFAR-100 with $n_l = 10000$ are reported in Appendix M. We observe DeSSL provides both a better cross-entropy and ECE with the same accuracy for small $\lambda$. For larger $\lambda$, DeSSL performs better in all the reported metrics. We performed a paired Student’s t-test to ensure that our results are significant and reported the p-values in Appendix M. The p-values indicate that for $\lambda$ close to 10, DeSSL is often significantly better in all the metrics. Moreover, DeSSL for large $\lambda$ provides a better cross-entropy and ECE than the complete case whereas SSL never does.

MedMNIST is a large-scale MNIST-like collection of biomedical images. We selected the five smallest 2D datasets of the collection, for these datasets it is likely that the cluster assumption no longer holds. Results are reported in Appendix L. We show DePseudoLabel competes with PseudoLabel in terms of accuracy and even success when PseudoLabel’s accuracy is less than the complete case. Moreover, DePseudoLabel is always better in terms of cross-entropy, so calibration, whereas PseudoLabel is always worse than the complete case. We focus on the unbalanced dataset DermaMNIST with a majority class and we compare the accuracy per class of the complete case, PseudoLabel and DePseudoLabel, see Figure 4. In this context, PseudoLabel fails to learn correctly the minority classes as the majority class is over-represented in the pseudo labels. DePseudoLabel outperforms both the Complete Case and PseudoLabel in term of balanced accuracy.

Figure 2: The influence of $\lambda$ on Pseudo-label and DePseudo-label for a Lenet trained on MNIST with $n_l = 1000$: (Left) Mean test accuracy; (Right) Mean test cross-entropy, with 95% CI.

Figure 3: Influence of $\lambda$ on Pseudo-label and DePseudo-label for a CNN trained on CIFAR with $n_l = 4000$: (Left) Mean test accuracy; (Right) Mean test cross-entropy, with 95% CI.
4.2 Fixmatch (Sohn et al., 2020)

The efficiency of the debiasing method lies in the correlation of the labelled objective and the unlabelled objective as shown in Theorem 3.1. So, we debiased a version of Fixmatch that includes strong augmentations in the labelled objective. See Appendix N for further details. We also compare our results to the original Fixmatch (Fixmatch∗ in Table 1). While the modified version is slightly worse than the original Fixmatch, debiasing it works. For this experiment, we use \( n_l = 4000 \) on 5 different folds for CIFAR-10 and \( n_l = 10000 \) on one fold for CIFAR-100. First, we report that a strong complete case baseline using data augmentation reaches 87.27% accuracy on CIFAR-10 (resp. 62.62% on CIFAR-100). A paired Student’s t-test ensures that our results are significantly better on CIFAR-10 (p-value in Appendix N). Then, we observe that the debiasing method improves both the accuracy and cross-entropy of this modified version of Fixmatch. Inspired by Zhu et al. (2022), we analyse individual classes on CIFAR-10 and show that our method improved performance on “poor” classes more equally than the biased version. Pseudo-label-based methods with a fixed selection threshold draw more pseudo-labels in the “easy” subpopulations. For instance, in the toy dataset (Figure 6), PseudoLabel will always draw samples of the class blue in the overlapping area. DeSSL prevents the method to overfit and be overconfident on the “easy” classes with the debiasing term. Indeed, DeFixmatch improves Fixmatch by 1.57% (resp. 1.94%) overall but by 4.91% on the worst class (resp. 8.00%). We report in Appendix N.1 the accuracy per class and in Appendix N.2 the results on STL10.

Table 1: Test accuracy, worst class accuracy and cross-entropy of Complete Case, Fixmatch and DeFixmatch on 5 folds of CIFAR-10 and one fold of CIFAR-100. Fixmatch∗ is the original version of Fixmatch, that uses strong augmentations only on the unlabelled.

<table>
<thead>
<tr>
<th></th>
<th>CIFAR-10 (( n_l = 4000 ))</th>
<th>CIFAR-100 (( n_l = 10000 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete Case</td>
<td>Fixmatch</td>
<td>Fixmatch∗</td>
</tr>
<tr>
<td>Accuracy</td>
<td>87.27 ± 0.25</td>
<td>93.87 ± 0.13</td>
</tr>
<tr>
<td>Worst class accuracy</td>
<td>70.08 ± 0.93</td>
<td>82.25 ± 2.27</td>
</tr>
<tr>
<td>Cross entropy</td>
<td>0.60 ± 0.01</td>
<td>0.27 ± 0.01</td>
</tr>
<tr>
<td>Brier score</td>
<td>0.214 ± 0.005</td>
<td>0.101 ± 0.003</td>
</tr>
</tbody>
</table>

5 Conclusion

Motivated by the remarks of van Engelen & Hoos (2020) and Oliver et al. (2018) on the missingness of theoretical guarantees in SSL, we proposed a simple modification of SSL frameworks. We consider frameworks based on the inclusion of unlabelled data in the computation of the risk estimator and debias them using labelled data. We show theoretically that this debiasing comes with several theoretical guarantees. We demonstrate these theoretical results experimentally on several common SSL datasets. DeSSL shows competitive performance in terms of accuracy compared to its biased version but improves significantly the calibration. There are several future directions open to us. Our experiments suggest that DeSSL will perform as well as SSL when classical SSL assumptions are true (such as the cluster assumption), for instance with MNIST in Figure 2. Still, for MNIST, DeSSL does not outperform in terms of accuracy but does in terms of cross-entropy and expected calibration error. However, when these assumptions do not hold anymore DeSSL will outperform its SSL counterparts. We showed that \( \lambda_{\text{opt}} \) exists (Theorem 5.1) and therefore our formula provides guidelines for the optimisation of \( \lambda \). Finally, an interesting improvement would be to go beyond the MCAR assumption by considering settings with a distribution mismatch between labelled and unlabelled data (Guo et al., 2020; Cao et al., 2021; Hu et al., 2022).
ACKNOWLEDGEMENTS

This work has been supported by the French government, through the 3IA Côte d’Azur, Investment in the Future, project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

We thank Jes Frellsen for suggesting the interpretation of DeSSL as a constrained optimisation problem. We are also grateful to the OPAL infrastructure from Université Côte d’Azur for providing resources and support.

REFERENCES


Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.


A TOY EXAMPLE

We trained a 4-layer neural network (1/20/100/20/1) with ReLU activation function using 25,000 labelled and 25,000 unlabelled points drawn from two 1D uniform laws with an overlap. We used $\lambda = 1$ and a confidence threshold for Pseudo-label $\tau = 0.70$. We optimised the model’s weights using a stochastic gradient descent (SGD) optimiser with a learning rate of 0.1.

Figure 5: Data histogram

Figure 6: 4-layer neural net trained using SSL methods on a 1D dataset drawn from two uniform laws. (Top-left) Posterior probabilities $p(1|x)$ of the same model trained following either complete case (only labelled data), Pseudo-label or our DePseudo-label. (Top-right) Same for EntMin and DeEntMin (Bottom-left) Training cross-entropy for Pseudo-label and DePseudo-label (Bottom-right) Training cross-entropy for EntMin and DeEntMin
B DETAILS ON SURROGATES AND MORE EXAMPLES

We provide in this appendix further details on our classification of SSL methods between entropy-based and consistency-based (see Section 2.3). We detail a general framework for both of these methods’ classes. We also show how popular SSL methods are related to our framework.

B.1 Entropy-based

We class as entropy-based, methods that aim to minimise a term of entropy such as Grandvalet & Bengio (2004) which minimises Shannon’s entropy or pseudo-label which is a form of entropy, see Remark E.5. These methods encourage the model to be confident on unlabelled data, implicitly using the cluster assumption. We recall those entropy-based methods can all be described as an expectation of $L$ under a distribution $\pi_x$ computed at the datapoint $x$:

$$H(\theta; x) = \mathbb{E}_{\pi_x(\tilde{x}, \tilde{y})}[L(\theta; \tilde{x}, \tilde{y})].$$  \hspace{1cm} (12)

**Pseudo-label:** As presented in the core article, the unsupervised objective of pseudo-label can be written as an expectation of $L$ on the distribution $\pi_x(\tilde{x}, \tilde{y}) = \delta_x(\tilde{x})\delta_y(y|x\tilde{x})$. The Pseudo-label unsupervised term can be written as an entropy in term of Rényi entropy, $H_\infty(x) = -\max_y \log p(y|x)$. For this reason, we class Pseudo-label methods as entropy-based methods. Recently, Lee (2013) encouraged the pseudo-labels method for deep semi-supervised learning. Then, Rizve et al. (2021) recently improved the pseudo-label selection by introducing an uncertainty-aware mechanism on the confidence of the model concerning the predicted probabilities. Pham et al. (2021) reaches state-of-the-art on the Imagenet challenge using pseudo-labels on a large dataset of additional images.

B.2 Pseudo-label and Data Augmentation

Recently, several methods based on data augmentation have been proposed and proven to perform well on a large spectrum of SSL tasks. The idea is to have a model resilient to strong data-augmentation of the input (Berthelot et al., 2019; 2020; Sohn et al., 2020; Xie et al., 2019; Zhang et al., 2021a). These methods rely both on the cluster assumption and the smoothness assumption and are at the border between entropy-based and consistency-based methods. The idea is to have the same prediction for an input and an augmented version of it. For instance, in Sohn et al. (2020), we first compute pseudo-labels predicted using a weakly-augmented version of $x$ (flip-and-shift data augmentation) and then minimise the likelihood with the predictions of the model on a strongly-augmented version of $x$. In Xie et al. (2019), the method is a little bit different as we minimise the cross-entropy between the prediction of the model on $x$ and the predictions of an augmented version. In both cases, the unsupervised part of the risk estimator can be reformulated as Equation 12.

**Fixmatch:** In Fixmatch, Sohn et al. (2020), the unsupervised objective can be written as:

$$H(\theta; x) = \mathbb{1}\max_y p_{\hat{y}}(y|x_1) > \tau]L(\theta; x_2, \arg\max_y p_{\hat{y}}(y|x_1))$$ \hspace{1cm} (13)

where $\hat{\theta}$ is a fixed copy of the current parameters $\theta$ indicating that the gradient is not propagated through it, $x_1$ is a weakly-augmented version of $x$ and $x_2$ a strongly-augmented one. Therefore, we write $H$ as an expectation of $L$ on the distribution $\pi_x(\tilde{x}, \tilde{y}) = \delta_{x_2}(\tilde{x})\delta_{\arg\max_y p_{\hat{y}}(y|x_1)}(\tilde{y})\mathbb{1}\max_y p_{\hat{y}}(y|x_1) > \tau]$.

**UDA:** In UDA, Xie et al. (2019), the unsupervised objective can be written as:

$$H(\theta; x) = \sum_yp_{\hat{y}}(y|x)L(\theta; x_1, y)$$ \hspace{1cm} (14)

where $\hat{\theta}$ is a fixed copy of the current parameters $\theta$ indicating that the gradient is not propagated through it and $x_1$ is an augmented version of $x$. Therefore, we write $H$ as an expectation of $L$ on the distribution $\pi_x(\tilde{x}, \tilde{y}) = \delta_{x_1}(\tilde{x})p_{\hat{y}}(\tilde{y}|\tilde{x})$.
Consistency-based methods aim to smooth the decision function of the models or have more stable predictions. These objectives $H$ are not directly a form of expectation of $L$ but are equivalent to an expectation of $L$. For all the following methods we can write the unsupervised objective $H$ such that:

$$C_1 E_{\pi_x(\hat{x}, \hat{y})}[L(\theta; \hat{x}, \hat{y})] \leq H(\theta; x) \leq C_2 E_{\pi_x(\tilde{x}, \tilde{y})}[L(\theta; \tilde{x}, \tilde{y})],$$

with $0 \leq C_1 \leq C_2$.

Indeed, consistency-based methods minimise an unsupervised objective that is a divergence between the model predictions and a modified version of the input (data augmentation) or a perturbation of the model. Using the fact that all norms are equivalent in a finite-dimensional space such as the space of the labels, we have the equivalence between a consistency-based $H$ and an expectation of $L$.

**VAT** The virtual adversarial training method proposed by [Miyato et al., 2018] generates the most impactful perturbation $r_{adv}$ to add to $x$. The objective is to train a model robust to input perturbations. This method is closely related to adversarial training introduced by [Goodfellow et al., 2014].

$$H(\theta; x) = \text{Div}(f_{\hat{\theta}}(x, .), f_{\theta}(x + r_{adv}, .))$$

where the $\text{Div}$ is a non-negative function that measures the divergence between two distributions, the cross-entropy or the KL divergence for instance. If the divergence function is the cross-entropy, it is straightforward to write the unlabelled objective as Equation[3] If the objective function is the KL divergence, we can write the objective as

$$H(\theta; x) = E_{\pi_x(\tilde{x}, \tilde{y})}[L(\theta; \tilde{x}, \tilde{y})] - E_{\pi_x(\tilde{x}, \tilde{y})}[L(\theta; \tilde{x}, \tilde{y})]$$

with $\pi_x(\tilde{x}, \tilde{y}) = \delta_{\tilde{x}}(\tilde{x})p_{\tilde{y}}(y|x)$. Therefore, variation of $H$ with respect to $\theta$ are the same as $E_{\pi_x(\tilde{x}+r,y)}[L(\theta; \tilde{x}, \tilde{y})]$. VAT is also a method between consistency-based and entropy-based methods as long as we use the KL-divergence or the cross-entropy as the measure of divergence.

**Mean-Teacher** A different form of pseudo-labelling is the Mean-Teacher approach proposed by [Tarvainen & Valpola, 2017] where pseudo-labels are generated by a teacher model for a student model. The parameters of the student model are updated, while the teacher’s are a moving average of the student’s parameters from the previous training steps. The idea is to have a more stable pseudo-labelling using the teacher than in the classic Pseudo-label. Final predictions are made by the student model. A generic form of the unsupervised part of the risk estimator is then

$$H(\theta; x) = \sum_y (p_{\theta}(y|x) - p_{\hat{\theta}}(y|x))^2,$$

where $\hat{\theta}$ are the fixed parameters of the teacher.

**II-Model** The II-Models are intrinsically stochastic models (for example a model with dropout) encouraged to make consistent predictions through several passes of the same $x$ in the model. The SSL loss is using the stochastic behaviour of the model where the model $f_{\theta}$ and penalises different predictions for the same $x$ [Sajjadi et al., 2016]. Let’s note $f_{\theta}(x, .)_1$ and $f_{\theta}(x, .)_2$ two passes of $x$ through the model $f_{\theta}$. A generic form of the unsupervised part of the risk estimator is then

$$H(\theta; x) = \text{Div}(f_{\theta}(x, .)_1, f_{\theta}(x, .)_2),$$

where $\text{Div}$ is a measure of divergence between two distributions (often the Kullback-Leibler divergence).
Temporal ensembling  Temporal ensembling (Laine & Aila, 2017) is a form of II-Model where we compare the current prediction of the model on the input $x$ with an accumulation of the previous passes through the model. Then, the training is faster as the network is evaluated only once per input on each epoch and the perturbation is expected to be less noisy than for II-models.

ICT  Interpolation consistency training (Verma et al., 2019) is an SSL method based on the mixup operation (Zhang et al., 2017). The model trained is then consistent to predictions at interpolations. The unsupervised term of the objective is then computed on two terms:

$$H(\theta; x_1, x_2) = \text{Div} \left( f_\theta(\alpha x_1 + (1-\alpha)x_2, .), \alpha f_\hat{\theta}(x_1, .) + (1 - \alpha) f_\hat{\theta}(x_2, .) \right), \quad (18)$$

with $\alpha$ drawn with from a distribution $B(a, a)$. With the exact same transformation, we will be able to show that this objective is equivalent to a form of expectation of $L$. 

C  SAFE SEMI-SUPERVISED LEARNING

C.1  ON THE DEFINITION OF SAFE SEMI-SUPERVISED LEARNING

Previous works introduced the notion safe semi-supervised learning for techniques which never reduce learning performance by introducing unlabelled data (Li & Zhou 2014, Kawakita & Takeuchi 2014, Li et al. 2016, Trapp et al. 2017, Guo et al. 2020). While the spirit of the definition of safe given by these works is the same, there are different ways of formalising it. In the following, we list the theoretical guarantees given on the performance of SSL methods by each of these works:

- Li & Zhou (2014) prove that the accuracy of their method, namely S4VM, is never worse than the complete case in a transductive setting under the low-density assumption.
- Kawakita & Takeuchi (2014) prove that the model learned is asymptotically normal and has a lower asymptotic variance without assumption on the data distribution but only when $n_l \leq n_u$.
- Li et al. (2016) prove that their method builds a model which is better for a chosen measure of performance in the range of top-k precision, $F_\beta$ score or AUC than the complete case under either the cluster assumption, the low-density assumption or the manifold assumption.
- Trapp et al. (2017) prove that the Sum-Product network learned has a lower train loss than the one with the complete case.
- Guo et al. (2020) prove that the empirical risk of the SSL model on the labelled train set is lower or equal than the one of the complete case with no assumption on the data distribution nor the missingness mechanism. Additionally, they prove generalisation error bounds.

Some other works proposed theoretical guarantees in the use of their SSL methods without calling the method safe. In regard to the definition and the safe SSL methods listed above, we can categorise the following method as safe. Among them, we cite the following:

- Sokolovska et al. (2008) prove that the model learned is asymptotically normal and has a lower asymptotic variance with the assumptions that the feature space is finite and $n_u \to \infty$.
- Fox-Roberts & Rosten (2014) propose an unbiased estimator of the likelihood in a generative setting and prove that the estimator is unbiased and has a lower variance than its complete case baseline with no assumption on the data distribution.
- Loog (2015) proposes a method for which the log-likelihood of the SSL model on the labelled train set is lower or equal to the one of the complete case with the gaussian distribution assumption of Linear Discriminant analysis.
- Sakai et al. (2017) prove that their risk estimator is unbiased and has a lower variance than the complete case baseline. Additionally, they prove generalisation error bounds and the bounds decrease with respect to the number of unlabelled data without assumption on the data distribution but with strong assumption on the chosen loss function.

C.2  ON THE SEMI-SUPERVISED BIAS

We provide in this appendix a further explanation of the risk induced by the SSL bias as introduced in Section 2.4.

Presented methods minimise a biased version of the risk under the MCAR assumption and therefore classical learning theory does not apply anymore,

$$\mathbb{E}[\hat{R}_{SSL}(\theta)] = \mathbb{E}[L(\theta; x, y)] + \lambda \mathbb{E}[H(\theta; x, y)] \neq R(\theta).$$

(19)

Learning over a biased estimate of the risk is not necessarily unsafe but it is difficult to provide theoretical guarantees on such methods even if some works try to do so with strong assumptions on the data distribution (Mey & Loog 2022, Section 4 and 5, Zhang et al. 2021b). Previous works proposed generalisation error bounds of SSL methods under strong assumptions on the data distribution or the true model. We refer to the survey by Mey & Loog (2022). More recently, Wei et al. (2021) proves
an upper bound for training deep models with the pseudo-label method under strong assumptions. Under soft assumptions, [Aminian et al., 2022] provides an error bound showing that the choice of $H$ is crucial to provide good performances.

Indeed, the unbiased nature of the risk estimate is crucial in the development of learning theory. This bias on the risk estimate may look like one of a regularisation, such as a ridge regularisation. However, SSL and regularisation are intrinsically different for several reasons:

- Regularisers have a vanishing impact in the limit of infinite data whereas SSL usually do not in the proposed methods, see Equation [19]. A solution would be to choose $\lambda$ with respect of the number of data points and make it vanish when $n$ goes to infinity. However, in most works, the choice of $\lambda$ is independent of the number of $n$ or $n_f$ (Oliver et al., 2018; Sohn et al., 2020).

- One of the main advantages of regularisation is to turn the learning problem into a “more convex” problem, see Shalev-Shwartz & Ben-David (2014, Chapter 13). Indeed, ridge regularisation will often turn a convex problem into a strongly-convex problem. However, SSL faces the danger to turn the learning problem into non-convex as previously noted by Sokolovska et al. (2008).

- The objective of a regulariser is to bias the risk towards optimum with smooth decision functions whereas entropy-based SSL will lead to sharp decision functions.

- Regularisation usually does not depend on the data whereas $H$ does in the SSL framework.

An entropy bias has been actually used by [Pereyra et al., 2017] as a regulariser but as entropy maximisation which should have the exact opposite effect of the SSL method introduced by [Grandvalet & Bengio, 2004], the entropy minimisation.
D Proof that $\hat{R}_{\text{DeSSL}}(\theta)$ is unbiased under MCAR

Theorem D.1. Under the MCAR assumption, $\hat{R}_{\text{DeSSL}}(\theta)$ is an unbiased estimator of $R(\theta)$.

As a consequence of the theorem, under the MCAR assumption, $\hat{R}_{\text{CC}}(\theta)$ is also unbiased as a special case of $\hat{R}_{\text{DeSSL}}(\theta)$ for $\lambda = 0$

Proof: We first recall that the DeSSL risk estimator $\hat{R}_{\text{DeSSL}}(\theta)$ is defined for any $\lambda$ by

$$\hat{R}_{\text{DeSSL}}(\theta) = \frac{1}{n_l} \sum_{i \in L} L(\theta; x_i, y_i) + \frac{\lambda}{n_u} \sum_{i \in U} H(\theta; x_i) - \frac{\lambda}{n_l} \sum_{i \in L} H(\theta; x_i) = \sum_{i=1}^{n} \left( \frac{r_i}{n_l} L(\theta; x_i, y_i) + \lambda \left( \frac{1 - r_i}{n_u} - \frac{r_i}{n_l} \right) H(\theta; x_i) \right).$$

By the law of total expectation:

$$E[\hat{R}_{\text{DeSSL}}(\theta)] = E_r \left[ E_{x,y}[\hat{R}_{\text{DeSSL}}(\theta)|r] \right].$$

As far as we are under the MCAR assumption, the data $(x, y)$ and the missingness variable $r$ are independent thus, $E_r \left[ E_{x,y}[\hat{R}_{\text{DeSSL}}(\theta)|r] \right] = E_r \left[ E_{x,y}[\hat{R}_{\text{DeSSL}}(\theta)] \right].$

We focus on $E_{x,y}[\hat{R}_{\text{DeSSL}}(\theta)]$. First, we replace $\hat{R}_{\text{DeSSL}}(\theta)$ by its definition and then use the linearity of the expectation. Then,

$$E_{x,y}[\hat{R}_{\text{DeSSL}}(\theta)] = E \left[ \frac{1}{n_l} \sum_{i \in L} L(\theta; x_i, y_i) + \frac{\lambda}{n_u} \sum_{i \in U} H(\theta; x_i) - \frac{\lambda}{n_l} \sum_{i \in L} H(\theta; x_i) \right] \text{ by definition}$$

$$= \frac{1}{n_l} \sum_{i \in L} E \left[ L(\theta; x_i, y_i) \right] + \frac{\lambda}{n_u} \sum_{i \in U} E \left[ H(\theta; x_i) \right] - \frac{\lambda}{n_l} \sum_{i \in L} E \left[ H(\theta; x_i) \right] \text{ by linearity}$$

The couples $(x_i, y_i)$ are i.i.d. samples following the same distribution. Then, we have

$$E_{x,y}[\hat{R}_{\text{DeSSL}}(\theta)|r] = \frac{1}{n_l} \sum_{i \in L} E \left[ L(\theta; x, y) \right] + \frac{\lambda}{n_u} \sum_{i \in U} E \left[ H(\theta; x) \right] - \frac{\lambda}{n_l} \sum_{i \in L} E \left[ H(\theta; x) \right] \text{ i.i.d samples}$$

$$= E \left[ L(\theta; x, y) \right]$$

$$= R(\theta).$$

Finally, we have the results that, $\hat{R}_{\text{DeSSL}}(\theta)$ is unbiased as $R(\theta)$ is a constant,

$$E[\hat{R}_{\text{DeSSL}}(\theta)] = E \left[ E_{x,y}[\hat{R}_{\text{DeSSL}}(\theta)|r] \right] = E_r \left[ R(\theta) \right] = R(\theta).$$
E Proof and Comments about Theorem 3.1

Theorem 3.1 The function $\lambda \mapsto \mathbb{V}(\hat{R}_{\text{DeSSL}}(\theta)|r)$ reaches its minimum for:

$$\lambda_{\text{opt}} = \frac{n_u}{n} \frac{\text{Cov}(L(\theta; x, y), H(\theta; x))}{\mathbb{V}(H(\theta; x))}$$  \hspace{2cm} (22)

and

$$\mathbb{V}(\hat{R}_{\text{DeSSL}}(\theta)|r)_{\lambda_{\text{opt}}} = \left(1 - \frac{n_u}{n} \rho_{L,H}^2 \right) \mathbb{V}(\hat{R}_{\text{CC}}(\theta)) \leq \mathbb{V}(\hat{R}_{\text{CC}}(\theta)), \hspace{2cm} (23)$$

where $\rho_{L,H} = \text{Corr}(L(\theta; x, y), H(\theta; x))$.

Proof: For any $\lambda \in \mathbb{R}$, we want to compute the variance:

$$\mathbb{V}(\hat{R}_{\text{DeSSL}}(\theta)|r).$$

Under the MCAR assumption, $x$ and $y$ are both jointly independent of $r$. Also, the couples $(x_i, y_i, r_i)$ are independent. Therefore, we have

$$\mathbb{V}(\hat{R}_{\text{DeSSL}}(\theta)|r) = \sum_{i=1}^{n} \mathbb{V}(x_i, y_i)_{\sim p(x,y)|r} \left( \frac{r_i}{n_l} L(\theta, x_i, y_i) + \lambda \left( \frac{1-r_i}{n_u} - \frac{r_i}{n_l} \right) H(\theta, x_i) \right) \text{ i.i.d samples}$$

$$= \sum_{i=1}^{n} \mathbb{V}(x_i, y_i)_{\sim p(x,y)} \left( \frac{r_i}{n_l} L(\theta, x, y) + \lambda \left( \frac{1-r_i}{n_u} - \frac{r_i}{n_l} \right) H(\theta, x) \right) \text{ (x, y) and r independent}$$

Using the fact that the couples $(x_i, y_i)$ are i.i.d. samples following the same distribution, we have

$$\mathbb{V}(\hat{R}_{\text{DeSSL}}(\theta)|r) = \sum_{i=1}^{n} \mathbb{V}(L(\theta, x, y)) + \lambda^2 \left( \frac{1-r_i}{n_u} - \frac{r_i}{n_l} \right)^2 \mathbb{V}(H(\theta, x)) \text{ using covariance}$$

$$+ 2\lambda \frac{r_i}{n_l} \left( \frac{1-r_i}{n_u} - \frac{r_i}{n_l} \right) \text{Cov}(L(\theta, x, y), H(\theta, x))$$

Now, we remark that the variable $r$ is binary and therefore $r^2 = r, (1-r)^2 = 1-r$ and $r(1-r) = 0$. Using that and simplifying, we have

$$\mathbb{V}(\hat{R}_{\text{DeSSL}}(\theta)|r) = \sum_{i=1}^{n} \frac{r_i}{n_l} \mathbb{V}(L(\theta, x, y)) + \lambda^2 \frac{(1-r_i)n_u^2 + r_in_l^2}{n_l^2 n_u^2} \mathbb{V}(H(\theta, x))$$

$$- 2\lambda \frac{r_i}{n_l} \text{Cov}(L(\theta, x, y), H(\theta, x))$$

Finally, by summing and simplifying the expression (note that $n_l + n_u = n$), we compute the expression variance,

$$\mathbb{V}(\hat{R}_{\text{DeSSL}}(\theta)|r) = \frac{1}{n_l} \mathbb{V}(L(\theta, x, y)) + \lambda^2 \frac{n}{n_l n_u} \mathbb{V}(H(\theta, x)) - 2\lambda \frac{n}{n_l} \text{Cov}(L(\theta, x, y), H(\theta, x))$$

So $\mathbb{V}(\hat{R}_{\text{DeSSL}}(\theta)|r)$ is a quadratic function in $\lambda$ and reaches its minimum for $\lambda_{\text{opt}}$ such that:

$$\lambda_{\text{opt}} = \frac{n_u}{n} \frac{\text{Cov}(L(\theta, x, y), H(\theta, x))}{\mathbb{V}(H(\theta, x))}$$
And, at \( \lambda_{\text{opt}} \), the variance of \( \hat{R}_{\text{DeSSL}}(\theta)||r) \) becomes

\[
\mathbb{V}(\hat{R}_{\text{DeSSL}}(\theta)||r) = \frac{1}{n_l} \mathbb{V}(L(\theta, x, y)) \left( 1 - \frac{n_u}{n} \text{Cov}(L(\theta, x, y), H(\theta, x)) \right) = \left( 1 - \frac{n_u}{n} \right) \mathbb{V}(L(\theta, x, y))
\]

Remark E.1. If \( H \) is perfectly correlated with \( L (\rho_{L, H} = 1) \), then the variance of the DeSSL estimator is equal to the variance of the estimator with no missing labels.

Remark E.2. **Is it possible to estimate \( \lambda_{\text{opt}} \) in practice?** The data distribution \( p(x, y) \) being unknown, the computation of \( \lambda_{\text{opt}} \) is not possible directly. Therefore, we need to use an estimator of the covariance \( \text{Cov}(L(\theta, x, y), H(\theta, x)) \) and the variance \( \mathbb{V}(H(\theta, x)) \) (See Equation [24]). Also, we have to be careful not to introduce a new bias with the computation of \( \lambda_{\text{opt}} \), indeed, if we compute it using the training set, \( \lambda_{\text{opt}} \) becomes dependent on \( x \) and \( y \) and therefore \( \hat{R}_{\text{DeSSL}}(\theta)||r) \) becomes biased. A solution would be to use a validation dataset for its computation. Another approach is to compute it using the splitting method (Avramidis & Wilson, 1993). Moreover, the computation of \( \lambda_{\text{opt}} \) is tiresome and time-consuming in practice as it has to be updated for every different value of \( \theta \), so at each gradient step.

\[
\hat{\lambda}_{\text{opt}} = \frac{1}{n} \sum_{i \in L}(L(\theta; x_i, y_i) - \bar{L}(\theta))(H(\theta; x_i) - \bar{H}(\theta))
\]

where \( H(\theta) = \frac{1}{n} \sum_{i=1}^{n} H(\theta; x_i) \) and \( L(\theta) = \frac{1}{n} \sum_{i \in L} L(\theta; x_i, y_i) \)

Remark E.3. **About the sign of \( \lambda \)** The theorem still has a quantitative merit when it comes to choosing \( \lambda \) by telling that the sign of \( \lambda \) is positive when \( H \) and \( L \) are positively correlated which will generally be the case with the examples mentioned in the article. For instance, concerning the entropy minimisation technique, the following proposition proves that the log-likelihood is negatively correlated with its entropy and therefore it justifies the choice of \( \lambda > 0 \) in the entropy minimisation.

**Proposition E.4.** The log-likelihood of the true distribution \( \log p(y|x) \) is negatively correlated with its entropy \( H_{\tilde{g}}(p(\tilde{y}|x)) \):

\[
\text{Cov}(\log p(y|x), H_{\tilde{g}}(p(\tilde{y}|x))) < 0
\]

**Proof.**

\[
\text{Cov}(\log p(y|x), H_{\tilde{g}}(p(\tilde{y}|x))) = \mathbb{E}_{x,y} [\log p(y|x)H_{\tilde{g}}(p(\tilde{y}|x))] - \mathbb{E}_{x,y} [\log p(y|x)] \mathbb{E}_{x} [H_{\tilde{g}}(p(\tilde{y}|x))] = -\mathbb{E}_{x,y} [\log p(y|x)] \mathbb{E}_{\tilde{g}|x} [\log p(\tilde{y}|x)] + \mathbb{E}_{x,y} [\log p(y|x)] \mathbb{E}_{\tilde{g}|x} [\log p(\tilde{y}|x)]
\]

By the law of total expectation, we have that \( \mathbb{E}_{\tilde{g}|x} [\log p(\tilde{y}|x)] = \mathbb{E}_{\tilde{g}} [\log p(\tilde{y}|x)] \), then

\[
\text{Cov}(\log p(y|x), H_{\tilde{g}}(p(\tilde{y}|x))) = -\mathbb{E}_{x,y} [\log p(y|x)] \mathbb{E}_{\tilde{g}|x} [\log p(\tilde{y}|x)] + \mathbb{E}_{x,y} [\log p(y|x)]^2
\]

On the other hand, also with the law of total expectation, \( \mathbb{E}_{x,y} [\log p(y|x)] \mathbb{E}_{\tilde{g}|x} [\log p(\tilde{y}|x)] \) = \( \mathbb{E}_{x} [\mathbb{E}_{y|x} [\log p(\tilde{y}|x)] \mathbb{E}_{\tilde{g}|x} [\log p(\tilde{y}|x)]] \), so

\[
\mathbb{E}_{x,y} [\log p(y|x)] \mathbb{E}_{\tilde{g}|x} [\log p(\tilde{y}|x)] \geq \mathbb{E}_{x} [\mathbb{E}_{y|x} [\log p(y|x)]^2] \geq \mathbb{E}_{x,y} [\log p(y|x)]^2
\]

Jensen’s inequality total expectation law.
Finally, we have the results, 
\[
\text{Cov}(\log p(y|x), \mathbb{H}_\theta(p(\hat{y}|x))) \leq \mathbb{E}_{x,y}[\log p(y|x)]^2 - \mathbb{E}_{x,y}[\log p(y|x)]^2 \leq 0
\]

\[\text{Remark E.5.}\] We can also see the Pseudo-label as a form of entropy. Indeed, modulo the confidence selection on the predicted probability, the Pseudo-label objective is the inverse of the Rényi min-entropy:
\[
\mathbb{H}_\infty(x) = -\max_y \log p(y|x)
\]

### E.1 On Risk Estimation Quality

We train CNN-13 on CIFAR-10 using only 4,000 labelled data and then split the test dataset into labelled and unlabelled data to estimate the PseudoLabel (PL) and DePseudoLabel (DePL) risks that we compared to the oracle risk estimate using all the test set. For this experiment, we split 100 times the test set into 40 labelled data and 9960 unlabelled data in order to estimate the variance of the risk estimator. We compute \( \lambda_{\text{opt}} \) using the entire test set. This experiment illustrates that DePL is unbiased for any value of \( \lambda \) (Figure 7 Left) and its variance can be optimised in \( \lambda \) and at \( \lambda_{\text{opt}} \), we have \( \mathbb{V}(\hat{R}_{\text{DePL}}(\theta)|r) \leq \mathbb{V}(\hat{R}_{\text{CC}}(\theta)|r) \). (Figure 7 Right)

![Figure 7: (Left) Risk estimate value for PseudoLabel (PL) and DePseudoLabel (DePL) compared to the true value of the risk. (Right) The influence of \( \lambda \) on the ratio \( \mathbb{V}(\hat{R}_{\text{DePL}}(\theta)|r)/\mathbb{V}(\hat{R}_{\text{CC}}(\theta)|r) \)](image)

The gradient of DeSSL is also an unbiased estimator of the risk’s gradient, as the Complete Case. We compare the quality of these two estimators under the same setting. We compare the variance of the gradient of DePL risk to the complete case and show that DePL reduces the variance considerably (see Figure 8). Variance reduction techniques applied on the gradients have shown promising results in the amelioration of optimisation algorithms [Defazio et al., 2014; Johnson & Zhang, 2013].

![Figure 8: The influence of \( \lambda \) on the ratio \( \mathbb{V}(\nabla \hat{R}_{\text{DePL}}(\theta)|r)/\mathbb{V}(\nabla \hat{R}_{\text{CC}}(\theta)|r) \)](image)
F  WHY DEBIASING WITH THE LABELLED DATASET?

We remark that the debiasing can be performed with any subset of the training data, labelled and unlabelled. The choice of debiasing only with the labelled data can be explained both intuitively and computationally in regard to the Theorem 3.1. Intuitively, the debiasing term penalises the confidence on the labelled datapoints and then prevents the overfitting on the train dataset. As remarked in section 3.1 [Pereyra et al. (2017)] showed that penalising low entropy models acts as a strong regulariser in supervised settings. This comforts the idea of penalising low entropy on the labelled dataset, i.e. debiasing the entropy minimisation with the labelled dataset. Considering Pseudo-Label-based methods, the objective for the labelled data is to predict the correct labels with moderate confidence. This is also similar to the concept of plausibility inference described by [Barndorff-Nielsen (1976)].

In regard to Theorem 3.1, we show that the optimum choice of subset for debiasing is either only the labelled data or the whole dataset and both are equivalent.

We consider a subset $\mathcal{A}$ of the training set. We defined $a$ as follow:

$$a_i = \begin{cases} 
1/|\mathcal{A}| & \text{if } x_i \in \mathcal{A} \\
0 & \text{otherwise} 
\end{cases}$$

The unbiased estimator is then:

$$\hat{\mathcal{R}}_{DeSSL,\mathcal{A}}(\theta) = \frac{1}{n_l} \sum_{i \in \mathcal{L}} L(\theta; x_i, y_i) + \frac{\lambda}{n_u} \sum_{i \in \mathcal{U}} H(\theta; x_i) - \frac{1}{n} \sum_{i=1}^{n} a_i H(\theta; x_i). \quad (26)$$

We compute the variance of this quantity as in the proof of Theorem 3.1 and show that:

$$\mathbb{V}(\hat{\mathcal{R}}_{DeSSL,\mathcal{A}}(\theta)|r) = \sum_{i=1}^{n} \frac{r_i}{n_u^2} \mathbb{V}(L(\theta, x, y)) + \lambda^2 \left( \frac{1-r_i}{n_u} - a_i \right)^2 \mathbb{V}(H(\theta, x))$$

$$- 2\frac{r_i a_i}{n_l} \mathbb{Cov}(L(\theta, x, y), H(\theta, x)) \quad (27)$$

Suppose that no labelled datapoints are in $\mathcal{A}$. Then, the last term of the variance is null. Hence, having no labelled datapoints in $\mathcal{A}$ leads to a variance increase. We also remark that debiasing with the entire dataset is equivalent that debiasing with the labelled datapoints. Indeed

$$\hat{\mathcal{R}}_{DeSSL}(\theta) = \frac{1}{n_l} \sum_{i \in \mathcal{L}} L(\theta; x_i, y_i) + \frac{\lambda}{n_u} \sum_{i \in \mathcal{U}} H(\theta; x_i) - \frac{1}{n} \sum_{i=1}^{n} H(\theta; x_i)$$

$$= \frac{1}{n_l} \sum_{i \in \mathcal{L}} L(\theta; x_i, y_i) + \frac{\lambda}{n_u} \sum_{i \in \mathcal{U}} H(\theta; x_i) - \frac{\lambda}{n} \sum_{i \in \mathcal{L}} H(\theta; x_i) - \frac{\lambda}{n} \sum_{i \in \mathcal{L}} H(\theta; x_i)$$

$$= \frac{1}{n_l} \sum_{i \in \mathcal{L}} L(\theta; x_i, y_i) + \frac{\lambda n_l}{nn_u} \sum_{i \in \mathcal{U}} H(\theta; x_i) - \frac{\lambda n_l}{nn_u} \sum_{i \in \mathcal{L}} H(\theta; x_i),$$

which is equivalent to debiasing with only the labelled dataset by replacing $\lambda$ by $\lambda n_l / n$.

At this point, we can still sample a random subset composed of $l$ labelled datapoints and $u$ unlabelled datapoints. Therefore $a_i = 1/(l+u)1\{x_i \in \mathcal{A}\}$, we show in the following that the optimum choice of the couples $(l, u)$ are $(n_l, 0)$ and $(n_l, n_u)$, so only the labelled or the whole dataset.

We sample $l$ labelled and $u$ unlabelled datapoints to debiased the estimator, by simplifying the term in the sum of Equation 27 as follow:

$$\left( \frac{1-r_i}{n_u} - a_i \right)^2 = \begin{cases} 
\left( \frac{1}{n_u} - \frac{1}{l+u} \right)^2 & \text{if } x_i \in \mathcal{A} \text{ and } r_i = 0 \\
\left( \frac{1}{l+u} \right)^2 & \text{if } x_i \in \mathcal{A} \text{ and } r_i = 1 \\
\frac{1}{n_u^2} & \text{if } x_i \notin \mathcal{A} \text{ and } r_i = 0 \\
0 & \text{if } x_i \notin \mathcal{A} \text{ and } r_i = 1 
\end{cases}$$
Then, by summing the term and simplifying, we get:

\[
\mathbb{V}(\hat{R}_{DeSSL}(\theta | r)) = \frac{1}{n_l} \mathbb{V}(L(\theta, x, y)) + \lambda^2 \left[ u \left( \frac{1}{n_u} - \frac{1}{l + u} \right) + \frac{l}{(l + u)^2} + \frac{n_u - u}{n_u^2} \right] \mathbb{V}(H(\theta, x)) \\
- 2\lambda \frac{l}{n_l(l + u)} \mathbb{Cov}(L(\theta, x, y), H(\theta, x)) \\
= \frac{1}{n_l} \mathbb{V}(L(\theta, x, y)) + \lambda^2 \frac{n_l}{n_u} \frac{n_u - u + l}{l + u} \mathbb{V}(H(\theta, x)) \\
- 2\lambda \frac{l}{n_l(l + u)} \mathbb{Cov}(L(\theta, x, y), H(\theta, x))
\]

We want to minimise \(\mathbb{V}(\hat{R}_{DeSSL}(\theta | r))\) with respect to \((\lambda, l, u)\). \(\mathbb{V}(\hat{R}_{DeSSL}(\theta | r))\) reaches is minimum in \(\lambda\) at

\[
\lambda_{opt} = \frac{n_u}{n_l} \frac{l}{n_u - u + l} \mathbb{Cov}(L(\theta, x, y), H(\theta, x)) / \mathbb{V}(H(\theta, x)).
\]

Then,

\[
\mathbb{V}(\hat{R}_{DeSSL}(\theta | r)) = \frac{1}{n_l} \mathbb{V}(L(\theta, x, y)) - \frac{n_u}{n_l} \frac{l^2}{(n_u - u + l)(l + u)} \mathbb{Cov}(L(\theta, x, y), H(\theta, x))^2 / \mathbb{V}(H(\theta, x)).
\]

We now want to minimise with respect to \(0 \leq u \leq n_u\) and \(1 \leq l \leq n_l\). We can easily show that the \((n_u - u + l)(l + u)\) reaches its minimum for \(u = 0\) or \(u = n_u\) and for both value:

\[
\mathbb{V}(\hat{R}_{DeSSL}(\theta | r)) = \frac{1}{n_l} \mathbb{V}(L(\theta, x, y)) - \frac{n_u}{n_l} \frac{l}{n_u + l} \mathbb{Cov}(L(\theta, x, y), H(\theta, x))^2 / \mathbb{V}(H(\theta, x)).
\]

Then \(l/(n_u + l)\) is an increasing function, then reaches its maximum at \(l = n_l\). So finally, the optimal choices for the couple \((n_l, 0)\) and \((n_l, n_u)\). We showed that these couples are equivalent.
The calibration of a model is its capacity of predicting probability estimates that are representative of the true distribution. This property is determinant in real-world applications when we need reliable predictions. A scoring rule $S$ is a function assigning a score to the predictive distribution $p_\theta(y|x)$ relative to the event $y|x \sim p(y|x)$, $S(p_\theta, (x, y))$, where $p(x, y)$ is the true distribution (see e.g. [Gneiting & Raftery, 2007]). A scoring rule measures both the accuracy and the quality of predictive uncertainty, meaning that better calibration is rewarded. The expected scoring rule is

$$\mathbb{E}_p[S(p_\theta, (x, y))].$$

A proper scoring rule is defined as a scoring rule such that $S(p_\theta, p) \leq S(p, p)$ (Gneiting & Raftery, 2007). The motivation behind having proper scoring rules comes from the following: suppose that the true data distribution $p$ is accessible by our set of models. Then, the scoring rule encourages to predict $p_\theta = p$. The opposite of a proper scoring rule can then be used to train a model to encourage the calibration of predictive uncertainty: $L(\theta; x, y) = -S(p_\theta, (x, y))$. The most common losses used to train models are proper scorings rules such as log-likelihood.

**Theorem G.1.** If $S(p_\theta, (x, y)) = -L(\theta; x, y)$ is a proper scoring rule, then $S'(p_\theta, (x, y, r)) = -\left(\frac{\tau n}{n_1} L(\theta; x, y) + \lambda n \left(\frac{1-r}{n_u} - \frac{r}{n_l}\right) H(\theta; x)\right)$ is also a proper scoring rule.

The proof follows directly from unbiasedness and the MCAR assumption. The main interpretation of this theorem is that we can expect DeSSL to be as well-calibrated as the complete case.

**Proof.** The scoring rule considered in our SSL framework is:

$$S'(p_\theta, (x, y, r)) = -\left(\frac{\tau n}{n_1} L(\theta; x, y) + \lambda n \left(\frac{1-r}{n_u} - \frac{r}{n_l}\right) H(\theta; x)\right).$$

The proper scoring rule of the fully supervised problem is

$$S(p_\theta, (x, y)) = -L(\theta; x, y).$$

Let $p$ be the true distribution of the data $(x, y, r)$. Under MCAR, $r$ is independent of $x$ and $y$, then $p(x, y, r) = p(r)p(x, y)$.

\[
\begin{align*}
S'(p_\theta, p) &= \int p(x, y, r)S'(p_\theta, (x, y, r)) \, dx \, dy \, dr \\
&= \int p(x, y)p(r)S'(p_\theta, (x, y, r)) \, dx \, dy \, dr \\
&= -\int p(x, y)p(r)\frac{\tau n}{n_1} L(\theta; x, y) + \lambda n \left(\frac{1-r}{n_u} - \frac{r}{n_l}\right) H(\theta; x) \, dx \, dy \, dr \\
&= -\int_{x,y} p(x, y) \left( \int_{r} p(r) \frac{\tau n}{n_1} \, dr \right) L(\theta; x, y) \, dx \, dy \\
&\quad \quad - \lambda n \int_{x,y} p(x, y) \left( \int_{r} p(r) \left(\frac{1-r}{n_u} - \frac{r}{n_l}\right) \, dr \right) H(\theta; x) \, dx \, dy \\
&= -\int_{x,y} p(x, y) L(\theta; x, y) \, dx \, dy \\
&= S(p_\theta, p)
\end{align*}
\]

Therefore, if $S(p_\theta, (x, y)) = -L(\theta; x, y)$ is a proper scoring rule, then $S'(p_\theta, (x, y, r)) = -(\frac{\tau n}{n_1} L(\theta; x, y) + \lambda n \left(\frac{1-r}{n_u} - \frac{r}{n_l}\right) H(\theta; x))$ is also a proper scoring rule.

$\square$
H PROOF OF THEOREM 3.4

Assumption 3.2 the minimum $\theta^*$ of $\mathcal{R}$ is well-separated.

$$\inf_{\theta: d(\theta^*, \theta) \geq \epsilon} \mathcal{R}(\theta) > \mathcal{R}(\theta^*) \quad (28)$$

Assumption 3.3 uniform weak law of large numbers holds for a function $L$ if:

$$\sup_{\theta \in \Theta} \left| \frac{1}{n} \sum_{i=1}^{n} L(\theta, x_i, y_i) - \mathbb{E}[L(\theta, x, y)] \right| \xrightarrow{p} 0 \quad (29)$$

**Theorem 3.4** Under assumption A and assumption B for both $L$ and $H$, $\hat{\theta} = \arg \min \mathcal{R}_{DSSL}$ is asymptotically consistent with respect to $n$.

This result is a direct application of Theorem 5.7 from [van der Vaart (1998)](https://example.com) Chapter 5 that states that under assumption A and B for $L$, $\hat{\theta} = \arg \min \mathcal{R}$ is asymptotically consistent with respect to $n$. Assumption A remains unchanged as we have M-estimators of the same $\mathcal{R}$. We now aim to prove that under assumption B for both $L$ and $H$, we have the assumption B on $\hat{\theta} \rightarrow \frac{c_n}{n} L(\theta; x, y) + \lambda(1 - \frac{c_n}{n}) H(\theta; x)$.

**Lemma H.1.** If the uniform law of large number holds for both $L$ and $H$, then it holds for $\theta \rightarrow \frac{c_n}{n} L(\theta; x, y) + \lambda(1 - \frac{c_n}{n}) H(\theta; x)$.

**Proof.** Suppose assumption B for $L$, then the same result holds if we replace $n$ with $n_L$ as $n$ and $n_L$ are coupled by the law of $r$. Indeed, when $n$ grows to infinity, $n_L$ too and inversely. Therefore,

$$\sup_{\theta \in \Theta} \left| \frac{1}{n} \sum_{i=1}^{n} L(\theta; x_i, y_i) - \mathbb{E}[L(\theta; x, y)] \right| \xrightarrow{p} 0$$

Now, suppose we have assumption B for $H$, then we can make the same remark than for $L$. Now, we have to show that:

$$\sup_{\theta \in \Theta} \left| \frac{1}{n} \sum_{i=1}^{n} \frac{r_n}{n_l} L(\theta; x, y) + \lambda n \left( \frac{1 - r}{n_u} - \frac{r}{n_l} \right) H(\theta; x) - \mathbb{E}[L(\theta; x, y)] \right| \xrightarrow{p} 0$$

We first split the absolute value and the sup operator as

$$\sup_{\theta \in \Theta} \left| \frac{1}{n} \sum_{i=1}^{n} \frac{r_n}{n_l} L(\theta; x, y) + \lambda n \left( \frac{1 - r}{n_u} - \frac{r}{n_l} \right) H(\theta; x) - \mathbb{E}[L(\theta; x, y)] \right| \leq \sup_{\theta \in \Theta} \left| \frac{1}{n_l} \sum_{i=1}^{n} \frac{r_n}{n_l} L(\theta; x, y) - \mathbb{E}[L(\theta; x, y)] \right| + \frac{1}{n} \sum_{i=1}^{n} \lambda n \left( \frac{1 - r}{n_u} - \frac{r}{n_l} \right) H(\theta; x)$$

$$\leq \sup_{\theta \in \Theta} \left| \frac{1}{n_l} \sum_{i=1}^{n} L(\theta; x, y) - \mathbb{E}[L(\theta; x, y)] \right| \xrightarrow{n \to 0} + \sup_{\theta \in \Theta} \left| \frac{1}{n_l} \sum_{i=1}^{n_l} \lambda n \left( \frac{1 - r}{n_u} - \frac{r}{n_l} \right) H(\theta; x) \right|.$$

So we now have to prove that the second term is also converging to 0 in probability. Again by splitting the absolute value and the sup, we have

$$\sup_{\theta \in \Theta} \left| n \sum_{i=1}^{n} \lambda n \left( \frac{1 - r}{n_u} - \frac{r}{n_l} \right) H(\theta; x) \right| = \sup_{\theta \in \Theta} \left| \lambda \frac{n}{n_l} \sum_{i=1}^{n} \left( \frac{1 - r}{n_u} - \frac{r}{n_l} \right) H(\theta; x) \right| = \lambda \frac{n}{n_l} \sum_{i=1}^{n} \left( \frac{1 - r}{n_u} - \frac{r}{n_l} \right) H(\theta; x)$$
Then we have that,

\[ \sup_{\theta \in \Theta} \left\{ \frac{\lambda}{n_u} \sum_{i=1}^{n} (1 - r) H(\theta; x) - \frac{\lambda}{n_l} \sum_{i=1}^{n} r H(\theta; x) \right\} \]

\[ \leq \sup_{\theta \in \Theta} \left[ \frac{\lambda}{n_u} \sum_{i=n-l+1}^{n} \frac{n_i+n_u}{n} H(\theta; x) - \mathbb{E}[H(\theta; x, y)] \right] + \sup_{\theta \in \Theta} \left[ \frac{n_i+n_l}{n} \mathbb{E}[H(\theta; x, y)] \right]. \]

Thus,

\[ \sup_{\theta \in \Theta} \left| \frac{1}{n} \sum_{i=1}^{n} r \frac{n}{n_l} L(\theta; x, y) + \lambda n \left( \frac{1-r}{n_u} - \frac{r}{n_l} \right) H(\theta; x) - \mathbb{E}[L(\theta; x, y)] \right| \xrightarrow{p} 0 \]

And we now just have to apply the results of van der Vaart (1998, Theorem 5.7) to have the asymptotic consistent of \( \hat{\theta} = \arg\min_{\theta} R_{DeSSL} \).

**Remark H.2.** A sufficient condition on the function \( H \) to verify assumption B, the uniform weak law of large numbers, is to be bounded (Newey & McFadden 1994, Lemma 2.4). For instance, the entropy \( H = - \sum_y p_\theta(y|x) \log(p_\theta(y|x)) \) is bounded and therefore, the entropy minimisation is asymptotically consistent.

\[ \square \]
I ASYMPTOTIC NORMALITY OF DESSL

In the following, we study a modified version of the objective to simplify the proof. Let us consider the following DeSSL objective \( L'(\theta; x, y, r) = \frac{r}{n} L(\theta; x, y) + \lambda \left( \frac{1 - r}{1 - \pi} - \frac{r}{\pi} \right) H(\theta; x) \) which has the same properties than the original one (unbiasedness, variance reduction property, consistency and benefit from generalisation error bounds). The idea is to replace \( n_i \) with \( \pi n \) to simplify the expression. The value \( n_i \) converges to \( \pi n \) then the following Theorem should hold with the true DeSSL objective.

We define the cross-covariance matrice between random vectors \( \nabla L(\theta; x, y) \) and \( \nabla H(\theta; x) \) as
\[
K_\theta(i, j) = \text{Cov}(\nabla L(\theta; x, y)_i, \nabla H(\theta; x)_j).
\]

\textbf{Theorem I.1.} Suppose \( L \) and \( H \) are smooth functions in \( \mathcal{C}^2(\Theta, \mathbb{R}) \). Assume \( R(\theta) \) admit a second-order Taylor expansion at \( \theta^* \) with a non-singular second order derivative \( V_{\theta^*} \). Under the MCAR assumption, we have that \( \hat{\theta}_{DeSSL} \) is asymptotically normal with covariance:

\[
\Sigma_{DeSSL} = \frac{1}{n} V_{\theta^*}^{-1} \mathbb{E} \left[ \nabla L(\theta^*; x, y) \nabla L(\theta^*; x, y)^T \right] V_{\theta^*}^{-1}
\]

\[
= \frac{\lambda^2}{\pi(1 - \pi)} V_{\theta^*}^{-1} \mathbb{E} \left[ \nabla H(\theta^*; x, y) \nabla H(\theta^*; x, y)^T \right] V_{\theta^*}^{-1}
\]

\[
- \frac{\lambda}{\pi} V_{\theta^*}^{-1} K_{\theta^*} V_{\theta^*}^{-1}.
\]

As a consequence, we can minimise the trace of the covariance. Indeed, \( \text{Tr}(\Sigma_{DeSSL}) \) reaches its minimum at
\[
\lambda_{opt} = (1 - \pi) \frac{\text{Tr}(V_{\theta^*}^{-1} K_{\theta^*} V_{\theta^*}^{-1})}{\text{Tr}(V_{\theta^*}^{-1} \mathbb{E} [\nabla H(\theta^*; x) \nabla H(\theta^*; x)^T] V_{\theta^*}^{-1})},
\]

and at \( \lambda_{opt} \):
\[
\text{Tr}(\Sigma_{DeSSL}) - \text{Tr}(\Sigma_{CC}) = - \frac{1 - \pi}{\pi} \frac{\text{Tr}(V_{\theta^*}^{-1} K_{\theta^*} V_{\theta^*}^{-1})^2}{\text{Tr}(V_{\theta^*}^{-1} \mathbb{E} [\nabla H(\theta^*; x) \nabla H(\theta^*; x)^T] V_{\theta^*}^{-1})} \leq 0.
\]

The complete case is the special case of DeSSL with \( \lambda = 0 \). Then, the Theorem holds for the complete case.

\textbf{Proof.} We define \( L'(\theta; x, y, r) = \frac{r}{n} L(\theta; x, y) \) \( + \lambda \left( \frac{1 - r}{1 - \pi} - \frac{r}{\pi} \right) H(\theta; x) \) The assumptions of the theorem are sufficient assumptions to apply Theorem 5.23 of Van der Vaart 1998 to the couple \((\hat{\theta}_{DeSSL}, L')\). Hence, we obtain the following representation for representation \( \hat{\theta}_{DeSSL} \):

\[
\sqrt{n}(\hat{\theta}_{DeSSL} - \theta^*) = \frac{1}{\sqrt{n}} V_{\theta^*}^{-1} \sum_{i=1}^{n} \frac{r_i}{\pi} \nabla L(\theta^*; x_i, y_i) + \lambda \left( \frac{1 - r_i}{1 - \pi} - \frac{r_i}{\pi} \right) \nabla H(\theta^*; x_i) + o_p(1).
\]

\[
\sqrt{n}(\hat{\theta}_{DeSSL} - \theta^*) \xrightarrow{\mathcal{D}} N(0, \Sigma_{DeSSL}).
\]

The asymptotic normality follows with variance:

\[
\Sigma_{DeSSL} = V_{\theta^*}^{-1} \mathbb{E} \left[ \nabla L'(\theta^*; x, y) \nabla L'(\theta^*; x, y)^T \right] V_{\theta^*}^{-1}.
\]

Using the MCAR assumption, we simplify the expression of \( \Sigma_{DeSSL} \):
\[
\Sigma_{\text{DeSSL}} = V_{\theta^*}^{-1} \mathbb{E} \left[ \nabla L(\theta^*; x, y) \nabla L(\theta^*; x, y)^T \right] V_{\theta^*}^{-1} \\
= \frac{1}{\pi^2} V_{\theta^*}^{-1} \mathbb{E} \left[ r \nabla L(\theta^*; x, y) \nabla L(\theta^*; x, y)^T \right] V_{\theta^*}^{-1} \\
+ \frac{\lambda^2}{\pi(1 - \pi)} V_{\theta^*}^{-1} \mathbb{E} \left[ \nabla H(\theta^*; x, y) \nabla H(\theta^*; x, y)^T \right] V_{\theta^*}^{-1} \\
- \frac{\lambda}{\pi} V_{\theta^*}^{-1} K_{\theta^*} V_{\theta^*}^{-1}.
\]

We remark that the complete case is the particular case of DeSSL with \( \lambda = 0 \). Then,
\[
\Sigma_{\text{DeSSL}} = \Sigma_{\text{CC}} + \frac{\lambda^2}{\pi(1 - \pi)} V_{\theta^*}^{-1} \mathbb{E} \left[ \nabla H(\theta^*; x, y) \nabla H(\theta^*; x, y)^T \right] V_{\theta^*}^{-1} \\
- \frac{\lambda}{\pi} V_{\theta^*}^{-1} K_{\theta^*} V_{\theta^*}^{-1}.
\]

The asymptotic relative efficiency of consequence, the asymptotic relative efficiency \( \hat{\theta}_{\text{DeSSL}} \) compared to \( \hat{\theta}_{\text{CC}} \) is defined as the quotient \( \frac{\text{Tr}(\Sigma_{\text{DeSSL}})}{\text{Tr}(\Sigma_{\text{CC}})} \). This quotient can be minimised with respect to \( \lambda \):
\[
\lambda_{\text{opt}} = (1 - \pi) \frac{\text{Tr}(V_{\theta^*}^{-1} K_{\theta^*} V_{\theta^*}^{-1})}{\text{Tr}(V_{\theta^*}^{-1} \mathbb{E} \left[ \nabla H(\theta^*; x) \nabla H(\theta^*; x)^T \right] V_{\theta^*}^{-1})}, \tag{33}
\]
and at \( \lambda_{\text{opt}} \):
\[
\frac{\text{Tr}(\Sigma_{\text{DeSSL}})}{\text{Tr}(\Sigma_{\text{CC}})} = 1 - \frac{1 - \pi}{\pi} \frac{\text{Tr}(V_{\theta^*}^{-1} K_{\theta^*} V_{\theta^*}^{-1})^2}{\text{Tr}(V_{\theta^*}^{-1} \mathbb{E} \left[ \nabla H(\theta^*; x) \nabla H(\theta^*; x)^T \right] V_{\theta^*}^{-1}) \text{Tr}(\Sigma_{\text{CC}})} \leq 1. \tag{34}
\]

**Remark 1.2.** **On the sign of** \( \lambda \). It is easy to show that a sufficient condition to have \( \lambda_{\text{opt}} > 0 \) is to have \( K_{\theta^*} \) positive semi-definite. Indeed, using that \( V_{\theta^*} \) is positive definite and Proposition 6.1 of Serre (2010), we show that \( \text{Tr}(V_{\theta^*}^{-1} K_{\theta^*} V_{\theta^*}^{-1}) > 0 \) and then \( \lambda_{\text{opt}} > 0 \).

**Remark 1.3.** **Why minimising the trace of** \( \Sigma_{\text{DeSSL}} \)? Minimising the trace of \( \Sigma_{\text{DeSSL}} \) leads to an estimator with a smaller asymptotic MSE, see Chen et al. (2020).

**Remark 1.4.** **Fully supervised setting.** We also remark that our theorem matches the theorem for the supervised setting. Indeed, observing all the labelled corresponds to the case \( \pi = 1 \) and we obtain:
\[
\Sigma_{\text{DeSSL}} = \Sigma_{\text{CC}} = \Sigma_{\text{Fully supervised}}.
\]
J Proof of Theorem 3.6

Our proof will be based on the following result from Shalev-Shwartz & Ben-David (2014, Theorem 26.5).

Theorem J.1. Let $\mathcal{H}$ be a set of parameters, $z \sim \mathcal{D}$ a random variable living in a space $Z$, $c > 0$, and $\ell : \mathcal{H} \times Z \rightarrow [-c, c]$. We denote $L_\mathcal{D}(h) = \mathbb{E}_z[\ell(h, z)]$, and $L_\mathcal{S}(h) = \frac{1}{m} \sum_{i=1}^m \ell(h, z_i)$, (35)

where $z_1, \ldots, z_m$ are i.i.d. samples from $\mathcal{D}$. For any $\delta > 0$, with probability at least $1 - \delta$, we have

$L_\mathcal{D}(h) \leq L_\mathcal{S}(h) + 2E(\varepsilon_i) \leq \frac{1}{m} \sum_{i=1}^m \varepsilon_i \ell(h, z_i) + 4c \sqrt{\frac{2 \log(4/\delta)}{m}},$ (36)

where $\varepsilon_1, \ldots, \varepsilon_m$ are i.i.d. Rademacher variables independent from $z_1, \ldots, z_m$.

We can now restate and prove our generalisation bound.

Theorem 3.6. We assume that both $L$ and $H$ are bounded and that the labels are MCAR. Then, there exists a constant $\kappa > 0$, that depends on $\lambda, L, H$, and the ratio of observed labels, such that, with probability at least $1 - \delta$, for all $\theta \in \Theta$,

$R(\theta) \leq \hat{R}_{DeSSL}(\theta) + 2R_n + \kappa \sqrt{\frac{\log(4/\delta)}{n}},$ (37)

where $R_n$ is the Rademacher complexity

$R_n = \mathbb{E}_{(\varepsilon_i)_{i \leq n}} \left[ \sup_{\theta \in \Theta} \left( \frac{1}{n \pi} \sum_{i \in \mathcal{L}} \varepsilon_i L(\theta; x_i, y_i) - \frac{\lambda}{n \pi} \sum_{i \in \mathcal{L}} \varepsilon_i H(\theta; x_i) + \frac{\lambda}{n(1 - \pi)} \sum_{i \in \mathcal{L}} \varepsilon_i H(\theta; x_i) \right) \right],$ (38)

with $\varepsilon_1, \ldots, \varepsilon_m$ i.i.d. Rademacher variables independent from the data.

Proof. We use Theorem J.1 with $z = (x, y, r), \mathcal{H} = \Theta, m = n$, and

$\ell(h, z) = \frac{1}{\pi} L(\theta; x_i, y_i) + \lambda \left( \frac{1 - r_i}{1 - \pi} - \frac{r_i}{\pi} \right) H(\theta; x_i).$ (39)

The unbiasedness of our estimate under the MCAR assumption, proven in Appendix D, ensures that the condition of Equation 35 is satisfied with $L_\mathcal{D}(h) = R(\theta)$ and $L_\mathcal{S}(h) = \hat{R}_{DeSSL}(\theta)$. Now, since $L$ and $H$ are bounded, there exists $M > 0$ such that $|L| < M$ and $|H| < M$. We can then bound $\ell$:

$|\ell(h, z)| \leq \frac{M}{\pi} + \lambda \max \left\{ \frac{1}{1 - \pi}, \frac{1}{\pi} \right\} M = c.$ (40)

Now that we have chosen a $c$ that bounds $\ell$, we can use Theorem J.1 and finally get Equation 37 with $\kappa = 4c \sqrt{2}$. □
K DeSSL with $H$ applied on all available data

For consistency-based SSL methods it is common to use all the available data for the consistency term:

$$\hat{R}_{SSL}(\theta) = \frac{1}{n_l} \sum_{i \in \mathcal{L}} L(\theta; x_i, y_i) + \frac{\lambda}{n} \sum_{i=1}^{n} H(\theta; x_i).$$  \hspace{1cm} (41)

With the same idea, we debias the risk estimate with the labelled data:

$$\hat{R}_{DeSSL}(\theta) = \frac{1}{n_l} \sum_{i \in \mathcal{L}} L(\theta; x_i, y_i) + \frac{\lambda}{n} \sum_{i=1}^{n} H(\theta; x_i) - \frac{\lambda}{n_l} \sum_{i \in \mathcal{L}} H(\theta; x_i).$$ \hspace{1cm} (42)

Under MCAR, this risk estimate is unbiased and the main theorem of the article hold with minor modifications. In Theorem 3.1, $\lambda_{opt}$ is slightly different and the expression of the variance at $\lambda_{opt}$ remains the same. The scoring rule in Theorem G.1 is different but the theorem remains the same. Both Theorem 3.4, 3.5 and 3.6 remain the same with very similar proofs.

**Theorem K.1.** The function $\lambda \mapsto \mathbb{V}(\hat{R}_{DeSSL}(\theta))$ reaches its minimum for:

$$\lambda_{opt} = \frac{\text{Cov}(L(\theta; x, y), H(\theta; x))}{\mathbb{V}(H(\theta; x))} \hspace{1cm} (43)$$

and

$$\mathbb{V}(\hat{R}_{DeSSL}(\theta))|_{\lambda_{opt}} = (1 - \frac{n_u}{n} \rho_{L,H}) \mathbb{V}(\hat{R}_{CC}(\theta)) \leq \mathbb{V}(\hat{R}_{CC}(\theta)) \hspace{1cm} (44)$$

where $\rho_{L,H} = \text{Corr}(L(\theta; x, y), H(\theta; x))$.

When $H$ is applied on all labelled and unlabelled data, the scoring rule used in the learning process is then $S'(p_\theta, (x, y, r)) = -\left(\frac{n_u}{n_l} L(\theta; x, y) + \lambda(1 - \frac{n_u}{n_l}) H(\theta; x)\right)$ and we have $S'$ is a proper scoring rule.
L. MNIST AND MedMNIST

L.1 MNIST

![Graphs showing test accuracy, mean test cross-entropy, and mean test ECE for different values of \( \lambda \).](image)

Figure 9: The influence of \( \lambda \) on Pseudo-label and DePseudo-label for a Lenet trained on MNIST with \( n_l = 1000 \): (Left) Test accuracy; (Middle) Mean test cross-entropy; (Right) Mean test ECE, with 95% CI.

L.2 MNIST LABEL NOISE

![Graphs showing mean test accuracy, mean test cross-entropy, and test ECE for different values of \( \lambda \).](image)

Figure 10: The influence of \( \lambda \) on Pseudo-label and DePseudo-label for a Lenet trained on MNIST with label noise with \( n_l = 1000 \): (Left) Mean test accuracy; (Middle) Mean test cross-entropy; (Right) Test ECE, with 95% CI.

L.3 MedMNIST

We trained a 5-layer CNN with a fixed \( \lambda = 1 \) and \( n_l \) at 10% of the training data. We report in Table 2 the mean accuracy and cross-entropy on 5 different splits of the labelled and unlabelled data and the number of labelled data used. We report the AUC in Appendix L. DePseudoLabel competes with PseudoLabel in terms of accuracy and even success when PseudoLabel’s accuracy is less than the complete case. Moreover, DePseudoLabel is always better in terms of cross-entropy, so calibration, whereas PseudoLabel is always worse than the complete case.
Published as a conference paper at ICLR 2023

Table 2: Test accuracy and cross-entropy of Complete Case (CC), PseudoLabel (PL) and DePseudoLabel (DePL) on five datasets of MedMNIST.

<table>
<thead>
<tr>
<th>DATASET</th>
<th>n_i</th>
<th>CC CROSS-ENTROPY</th>
<th>CC ACCURACY</th>
<th>PL CROSS-ENTROPY</th>
<th>PL ACCURACY</th>
<th>DePL CROSS-ENTROPY</th>
<th>DePL ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>DERMA</td>
<td>1000</td>
<td>1.95 ± 0.09</td>
<td>68.99 ± 1.20</td>
<td>2.51 ± 0.20</td>
<td>68.88 ± 1.03</td>
<td>1.88 ± 0.12</td>
<td>69.30 ± 0.85</td>
</tr>
<tr>
<td>PNEUMONIA</td>
<td>585</td>
<td>1.47 ± 0.04</td>
<td>83.94 ± 2.40</td>
<td>2.04 ± 0.04</td>
<td>85.83 ± 2.13</td>
<td>1.40 ± 0.06</td>
<td>84.36 ± 3.79</td>
</tr>
<tr>
<td>RETINA</td>
<td>160</td>
<td>1.68 ± 0.03</td>
<td>48.30 ± 3.06</td>
<td>1.80 ± 0.18</td>
<td>47.75 ± 2.50</td>
<td>1.67 ± 0.06</td>
<td>49.40 ± 2.62</td>
</tr>
<tr>
<td>BREAST</td>
<td>78</td>
<td>0.80 ± 0.04</td>
<td>76.15 ± 0.75</td>
<td>1.00 ± 0.26</td>
<td>74.74 ± 1.04</td>
<td>0.70 ± 0.03</td>
<td>76.67 ± 1.32</td>
</tr>
<tr>
<td>BLOOD</td>
<td>1700</td>
<td>6.11 ± 0.17</td>
<td>84.13 ± 0.83</td>
<td>6.61 ± 0.22</td>
<td>84.09 ± 1.17</td>
<td>6.53 ± 0.30</td>
<td>83.68 ± 0.59</td>
</tr>
</tbody>
</table>

Table 3: Test AUC of Complete Case, PseudoLabel and DePseudoLabel on five datasets of MedMNIST.

<table>
<thead>
<tr>
<th>DATASET</th>
<th>COMPLETE CASE</th>
<th>PSEUDOLABEL</th>
<th>DEPSEUDOLABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DERMA</td>
<td>84.26 ± 0.50</td>
<td>82.64 ± 1.19</td>
<td>83.82 ± 0.95</td>
</tr>
<tr>
<td>PNEUMONIA</td>
<td>94.28 ± 0.46</td>
<td>94.34 ± 0.91</td>
<td>94.15 ± 0.33</td>
</tr>
<tr>
<td>RETINA</td>
<td>70.70 ± 0.74</td>
<td>70.12 ± 1.01</td>
<td>69.97 ± 1.44</td>
</tr>
<tr>
<td>BREAST</td>
<td>74.67 ± 3.68</td>
<td>74.86 ± 3.18</td>
<td>75.33 ± 3.05</td>
</tr>
<tr>
<td>BLOOD</td>
<td>97.83 ± 0.23</td>
<td>97.83 ± 0.23</td>
<td>97.72 ± 0.15</td>
</tr>
</tbody>
</table>
M  PSEUDOLABEL AND DEPSEUDOLABEL ON CIFAR: P-VALUES

M.1  CIFAR-10

Figure 11: The influence of $\lambda$ on Pseudo-label and DePseudo-label on CIFAR-10 with nl= 4000: (Left) Mean test accuracy; (Middle) Mean test cross-entropy; (Right) Test ECE, with 95% CI.

Figure 12: p-values of a paired student test between PseudoLabel and DePseudoLabel (Right) DePseudoLabel is better than PseudoLabel; (Left) DePseudoLabel is worst than PseudoLabel.

M.2  COMPUTATION OF $\lambda_{opt}$ ON THE TEST SET.

As explained in the main text, the estimation of $\text{Cov}(L(\theta; x, y), H(\theta; x))$ with few labels led to extremely unstable unsatisfactory results. However, we test the formula on CIFAR-10 and different methods to provide intuition on the order of $\lambda_{opt}$ and the range of the variance reduction regime (between 0 and $2\lambda_{opt}$). To do so, we estimate $\lambda_{opt}$ on the test set for CIFAR-10 by training a CNN13 using only 4,000 labelled data on 200 epochs. The value of $\lambda_{opt}$ is 1.67, 31.16 and 0.66 for entropy minimisation, pseudo label and Fixmatch. Therefore, the reduced variance regime covers the intuitive choices of $\lambda$ in the SSL literature. Unfortunately, computing $\lambda_{opt}$ on the test set is not applicable in practice.
M.3 CIFAR-100

![Graphs showing the influence of λ on Pseudo-label and DePseudo-label on CIFAR-100 with nl= 4000: (Left) Mean test accuracy; (Middle) Mean test cross-entropy; (Right) Test ECE, with 95% CI.]

Figure 13: The influence of λ on Pseudo-label and DePseudo-label on CIFAR-100 with nl= 4000: (Left) Mean test accuracy; (Middle) Mean test cross-entropy; (Right) Test ECE, with 95% CI.
Fixmatch (Sohn et al., 2020)

We performed a paired Student t-test to ensure that our performances are significantly better on CIFAR-10 on both accuracy, cross-entropy and Brier score. The results show that DeFixmatch is significantly better with a p-value of \(6.5 \times 10^{-5}\) in accuracy, \(3.3 \times 10^{-5}\) in cross-entropy and \(7.6 \times 10^{-5}\) in Brier score.

N.1 Per Class Accuracy

In recent work, Zhu et al. (2022) exposed the disparate effect of SSL on different classes. Indeed, classes with a high complete case accuracy benefit more from SSL than classes with a low baseline accuracy. They introduced a metric called the benefit ratio (BR) that quantifies the impact of SSL on a class \(C\):

\[
BR(C) = \frac{\text{acc}_{SSL}(C) - \text{acc}_{CC}(C)}{\text{acc}_S(C) - \text{acc}_{CC}(C)},
\]

where \(\text{acc}_{SSL}(C)\), \(\text{acc}_{CC}(C)\) and \(\text{acc}_S(C)\) are respectively the accuracy of the class with an SSL trained model, a complete-case model and a fully supervised model (a model that has access to all labels). Inspired by this work, we report the per-class accuracy and the benefit ratio in Table N.1. We see that the “poor” classes such as bird, cat and dog tend to benefit from DeFixmatch much more than from Fixmatch. We compute \(\text{acc}_S(C)\) using a pre-trained model with the same architecture.

Zhu et al. (2022) also promote the idea that a fair SSL algorithm should benefit different sub-classes equally, then having \(BR(C) = BR(C')\) for all \(C, C'\). While perfect equality seems unachievable in practice, we propose to look at the standard deviation of the \(BR\) through the different classes. While the standard deviation of Fixmatch is 0.12, the one of DeFixmatch is 0.06. Therefore, DeFixmatch improves the sub-populations’ accuracies more equally.

Table 4: Mean accuracy per class and mean benefit ratio (BR) on 5 folds for Fixmatch, DeFixmatch and the Complete Case. Bold: “poor” complete case accuracy classes.

<table>
<thead>
<tr>
<th>Class</th>
<th>Complete Case</th>
<th>Fixmatch</th>
<th>BR</th>
<th>DeFixmatch</th>
<th>BR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIRPLANE</td>
<td>86.94</td>
<td>95.94</td>
<td>0.88</td>
<td>96.62</td>
<td>0.94</td>
</tr>
<tr>
<td>AUTOMOBILE</td>
<td>95.26</td>
<td>97.54</td>
<td>0.68</td>
<td>98.22</td>
<td>0.89</td>
</tr>
<tr>
<td>BIRD</td>
<td>80.46</td>
<td>90.80</td>
<td>0.68</td>
<td>92.64</td>
<td>0.80</td>
</tr>
<tr>
<td>CAT</td>
<td>70.08</td>
<td>82.50</td>
<td>0.56</td>
<td>87.16</td>
<td>0.78</td>
</tr>
<tr>
<td>DEER</td>
<td>88.88</td>
<td>95.86</td>
<td>0.78</td>
<td>97.26</td>
<td>0.94</td>
</tr>
<tr>
<td>DOG</td>
<td>79.66</td>
<td>87.16</td>
<td>0.53</td>
<td>90.98</td>
<td>0.81</td>
</tr>
<tr>
<td>FROG</td>
<td>93.12</td>
<td>97.84</td>
<td>0.80</td>
<td>98.26</td>
<td>0.94</td>
</tr>
<tr>
<td>HORSE</td>
<td>90.96</td>
<td>96.94</td>
<td>0.83</td>
<td>97.64</td>
<td>0.92</td>
</tr>
<tr>
<td>SHIP</td>
<td>94.12</td>
<td>97.26</td>
<td>0.67</td>
<td>98.06</td>
<td>0.84</td>
</tr>
<tr>
<td>TRUCK</td>
<td>93.18</td>
<td>96.82</td>
<td>0.84</td>
<td>97.20</td>
<td>0.93</td>
</tr>
</tbody>
</table>

N.2 STL10

DeFixmatch also performs better than Fixmatch on STL-10 using 4200 labelled data for training and 800 for validation.

N.3 Fixmatch Details

As first detailed in Appendix B, Fixmatch is a pseudo-label-based method with data augmentation. Indeed, Fixmatch uses weak augmentations of \(x\) (flip-and-shift) for the pseudo-labels selection and then minimises the likelihood with the prediction of the model on a strongly augmented version of \(x\).
Published as a conference paper at ICLR 2023

Table 5: Test accuracy, worst class accuracy and cross-entropy of Complete Case, Fixmatch and DeFixmatch on 5 folds of CIFAR-10, one fold of CIFAR-100 and one fold of STL10.

<table>
<thead>
<tr>
<th></th>
<th>CIFAR-10 ($n_l = 4000$)</th>
<th>CIFAR-100 ($n_l = 10000$)</th>
<th>STL10 ($n_l = 5000$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Complete Case</td>
<td>Fixmatch</td>
<td>DeFixmatch</td>
</tr>
<tr>
<td></td>
<td>Complete Case</td>
<td>Fixmatch</td>
<td>DeFixmatch</td>
</tr>
<tr>
<td></td>
<td>Complete Case</td>
<td>Fixmatch</td>
<td>DeFixmatch</td>
</tr>
<tr>
<td>Accuracy</td>
<td>87.27 ± 0.25</td>
<td>93.87 ± 0.13</td>
<td>95.44 ± 0.10</td>
</tr>
<tr>
<td>Worst class accuracy</td>
<td>70.08 ± 0.93</td>
<td>82.25 ± 2.27</td>
<td>87.16 ± 0.46</td>
</tr>
<tr>
<td>Cross entropy</td>
<td>0.60 ± 0.01</td>
<td>0.27 ± 0.01</td>
<td>0.20 ± 0.01</td>
</tr>
<tr>
<td>Brier score</td>
<td>0.214 ± 0.005</td>
<td>0.101 ± 0.003</td>
<td>0.076 ± 0.001</td>
</tr>
</tbody>
</table>

Weak augmentations are also used for the supervised part of the loss. In this context,

\[
L(\theta; x, y) = \mathbb{E}_{x_1 \sim \text{weak}(x)} \left[ - \log(p_\theta(y|x_1)) \right]
\]

and

\[
H(\theta; x) = \mathbb{E}_{x_1 \sim \text{weak}(x)} \left[ \mathbb{I}[\max_y p_\theta(y|x_1) > \tau] \mathbb{E}_{x_2 \sim \text{strong}(x)} \left[ - \log(p_\theta(\arg\max_y p_\theta(y|x_1)|x_2)) \right] \right]
\]

where \(x_1\) is a weak augmentation of \(x\) and \(x_2\) is a strong augmentation. We tried to debias an implementation of Fixmatch\(^1\) however training was very unstable and led to model that were much worst than the complete case. We believed that this behaviour is because the supervised part of the loss does not include strong augmentation. Indeed, our theoretical results encourage to have a strong correlation between \(L\) and \(H\), therefore including strong augmentations in the supervised term. Moreover, a solid baseline for CIFAR-10 using only labelled data integrated strong augmentations (Cubuk et al., 2020). We modify the implementation, see Code\(^2\). Therefore, the supervised loss term can be written as:

\[
L(\theta; x, y) = \frac{1}{2} \left( \mathbb{E}_{x_1 \sim \text{weak}(x)} \left[ - \log(p_\theta(y|x_1)) \right] + \mathbb{E}_{x_2 \sim \text{strong}(x)} \left[ - \log(p_\theta(y|x_2)) \right] \right), \quad (46)
\]

where \(x_1\) is a weak augmentation of \(x\) and \(x_2\) is a strong augmentation. This modification encourages us to choose \(\lambda = \frac{1}{2}\) as the original Fixmatch implementation used \(\lambda = 1\). We also remark that this modification degrades the performance of Fixmatch (less than 2%) reported in the work of Sohn et al. (2020). However, including strong augmentations in the supervised part greatly improves the performance of the Complete Case.

\(^1\)https://github.com/LeeDoYup/FixMatch-pytorch
\(^2\)https://github.com/HugoSchmutz/DeFixmatch.git
In this section, we present the results on CIFAR and SVHN by debiasing the implementation of \cite{Oliver2018} of II-Model, Mean-Teacher and VAT. We mimic the experiments of \cite{Oliver2018} figure-4 with the same configuration and the exact same hyperparameters (Oliver et al., 2018 Appendix B and C). We perform an early stopping independently on both cross-entropy and accuracy. As reported below, we reach almost the same results as the biased methods.

### 0.1 CIFAR-10

![Figure 14](https://github.com/brain-research/realistic-ssl-evaluation)

**Figure 14:** Test accuracy for each SSL approaches on CIFAR-10 with various amounts of labelled data $n_l$. (Left) II-model and DeII-model. (Right) VAT+EntMin and VAT+DeEntMin. (Bottom) Mean-teacher and DeMean-teacher. Shadows represent 95% CI.

![Figure 15](https://github.com/brain-research/realistic-ssl-evaluation)

**Figure 15:** Test cross-entropy for each SSL approaches on CIFAR-10 with various amounts of labelled data $n_l$. (Left) II-model and DeII-model. (Right) VAT+EntMin and VAT+DeEntMin. (Bottom) Mean-teacher and DeMean-teacher. Shadows represent 95% CI.
### O.2 SVHN

#### Figure 16: Test accuracy for each SSL approaches on CIFAR-10 with various amounts of labelled data $n_l$. (Left) II-model and DeII-model. (Right) VAT+EntMin and VAT+DeEntMin. (Bottom) Mean-teacher and DeMean-teacher. Shadows represent 95% CI.

<table>
<thead>
<tr>
<th>$n_l$</th>
<th>Complete Case</th>
<th>Pi-Model</th>
<th>DePi-Model</th>
<th>VAT + EntMin</th>
<th>VAT + DeEntMin</th>
<th>Mean-Teacher</th>
<th>DeMean-Teacher</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>0.70</td>
<td>0.65</td>
<td>0.70</td>
<td>0.70</td>
<td>0.65</td>
<td>0.70</td>
<td>0.65</td>
</tr>
<tr>
<td>500</td>
<td>0.75</td>
<td>0.65</td>
<td>0.75</td>
<td>0.75</td>
<td>0.65</td>
<td>0.75</td>
<td>0.65</td>
</tr>
<tr>
<td>1000</td>
<td>0.80</td>
<td>0.65</td>
<td>0.80</td>
<td>0.80</td>
<td>0.65</td>
<td>0.80</td>
<td>0.65</td>
</tr>
<tr>
<td>2000</td>
<td>0.85</td>
<td>0.65</td>
<td>0.85</td>
<td>0.85</td>
<td>0.65</td>
<td>0.85</td>
<td>0.65</td>
</tr>
<tr>
<td>4000</td>
<td>0.90</td>
<td>0.65</td>
<td>0.90</td>
<td>0.90</td>
<td>0.65</td>
<td>0.90</td>
<td>0.65</td>
</tr>
<tr>
<td>8000</td>
<td>0.95</td>
<td>0.65</td>
<td>0.95</td>
<td>0.95</td>
<td>0.65</td>
<td>0.95</td>
<td>0.65</td>
</tr>
</tbody>
</table>

#### Figure 17: Test cross-entropy for each SSL approaches on CIFAR-10 with various amounts of labelled data $n_l$. (Left) II-model and DeII-model. (Right) VAT+EntMin and VAT+DeEntMin. (Bottom) Mean-teacher and DeMean-teacher. Shadows represent 95% CI.

<table>
<thead>
<tr>
<th>$n_l$</th>
<th>Complete Case</th>
<th>Pi-Model</th>
<th>DePi-Model</th>
<th>VAT + EntMin</th>
<th>VAT + DeEntMin</th>
<th>Mean-Teacher</th>
<th>DeMean-Teacher</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>2.00</td>
<td>1.98</td>
<td>2.00</td>
<td>2.00</td>
<td>1.98</td>
<td>2.00</td>
<td>1.98</td>
</tr>
<tr>
<td>500</td>
<td>2.05</td>
<td>1.98</td>
<td>2.05</td>
<td>2.05</td>
<td>1.98</td>
<td>2.05</td>
<td>1.98</td>
</tr>
<tr>
<td>1000</td>
<td>2.10</td>
<td>1.98</td>
<td>2.10</td>
<td>2.10</td>
<td>1.98</td>
<td>2.10</td>
<td>1.98</td>
</tr>
<tr>
<td>2000</td>
<td>2.15</td>
<td>1.98</td>
<td>2.15</td>
<td>2.15</td>
<td>1.98</td>
<td>2.15</td>
<td>1.98</td>
</tr>
<tr>
<td>4000</td>
<td>2.20</td>
<td>1.98</td>
<td>2.20</td>
<td>2.20</td>
<td>1.98</td>
<td>2.20</td>
<td>1.98</td>
</tr>
<tr>
<td>8000</td>
<td>2.25</td>
<td>1.98</td>
<td>2.25</td>
<td>2.25</td>
<td>1.98</td>
<td>2.25</td>
<td>1.98</td>
</tr>
</tbody>
</table>
P COMPUTATION DETAILS

P.1 COMPUTATION RESOURCES

Deep Learning experiments of this work required approximately 11,250 hours of GPU computation. In particular, Fixmatch was trained using 4 GPUs. Here are the details:

- MNIST: 300 hours
- medMNIST: 3 hours
- CIFAR-10: 525 hours
- CIFAR-100: 1500 hours
- Fixmatch: 3000 hours
- Realistic SSL evaluation on both CIFAR and SVHN: 5880 hours

P.2 COMPUTATION LIBRARIES AND TOOLS

- Python [Van Rossum & Drake Jr. (1995)]
- PyTorch [Paszke et al. (2019)]
- TensorFlow [Abadi et al. (2015)]
- Scikit-learn [Pedregosa et al. (2011)]
- Seaborn [Waskom et al. (2017)]
- Python imaging library [Lundh et al. (2012)]
- Numpy [Harris et al. (2020)]
- Pandas [McKinney et al. (2010)]
- RandAugment [Cubuk et al. (2020)]
- Fixmatch-Pytorch [4]
- Realistic-SSL-evaluation [Oliver et al. (2018)]

[4] https://github.com/LeeDoYup/FixMatch-pytorch