A Notation

For two vectors $p, q \in \mathbf{R}^d$ we denote by $p \cdot q = \sum_{i=1}^{d} p_i q_i$ their inner product. We use $p * q$ to denote their element-wise product, i.e., $(p * q)_i = p_i q_i$. We use the notation $\max_i p$ to denote the i-th largest element of the vector p. We use $\mathrm{margin}(p)$ to denote the difference between the top-2 elements of p, i.e., $\mathrm{margin}(p) = \max_1 p - \max_2 p$. Moreover, we use $\mathrm{margin}_k(p)$ to denote the top-k margin, i.e., $\mathrm{margin}_k(p) = \sum_{i=1}^{k} \max_i p - \max_{k+1} p$. Given a function $f(w) : \mathbf{R}^d \mapsto \mathbf{R}$ we denote by $\partial_w f(w)$ the gradient of f with respect to the parameter w.

B Detailed Description of SLaM

B.1 Estimating the Teacher's Accuracy Parameters: $\alpha(x), k(x)$

Estimating the Teacher's Accuracy $\alpha(x)$ via Isotonic Regression We now turn our attention to the problem of estimating $\alpha(x)$ for each x of dataset B, i.e., the dataset labeled by the teacher model. In [27] the authors empirically observed that $\alpha(x)$ correlates with metrics of teacher's confidence such as the "margin", i.e., the difference between the probabilities assigned in the top-1 class and the second largest class according to the teacher's soft label y_s. In particular, the larger the margin is the more likely is that the corresponding teacher label is correct. We exploit (and enforce) this monotonicity by employing isotonic regression on a small validation dataset to learn the mapping from the teacher's margin at an example x to the corresponding teacher's accuracy $\alpha(x)$.

To perform this regression task we use a small validation dataset V with correct labels that the teacher has not seen during training. For every example $x \in V$ we compute the corresponding soft-teacher label $y_s(x)$ and compute its margin $\mathrm{margin}(x) = \max_1(y_s(x)) - \max_2(y_s(x))$. For every $x \in V$ we also compute the hard-prediction of the teacher and compare it with the ground-truth, i.e., for every x the covariate and responce pair is $(\mathrm{margin}(x), 1 - \mathrm{err}(g(x), y(x)))$. We then use isotonic regression to fit a piecewise constant, increasing function to the data. Sorting the regression data $\{(\mathrm{margin}(x), 1 - \mathrm{err}(g(x), y(x)))x \in V\}$ by increasing margin to obtain a list $(c^{(1)}, \ldots, r^{(1)}), \ldots, (c^{(m)}, r^{(m)})$, isotonic regression solves the following task

$$\begin{aligned} \min_{\hat{r}^{(1)}, \ldots, \hat{r}^{(m)}} & \sum_{i=1}^{m} (r^{(i)} - \hat{r}^{(i)})^2 \\ \text{subject to} \quad & \mathrm{lb} \le \hat{r}^{(i)} \le \hat{r}^{(i+1)} \le 1, \end{aligned}$$

where the parameter lb is a lower bound on the values $\hat{r}^{(i)}$ and is a hyper-parameter that we tune. On the other hand, the upper bound for the values can be set to 1 since we know that the true value $\alpha(x)$ is at most 1 for every x (since it corresponds to the probability that the teacher-label is correct). After we compute the values $\hat{r}^{(1)}, \ldots, \hat{r}^{(m)}$ for any given $c \in [0, 1]$ the output of the regressor is the value of $\hat{r}^{(i)}$ corresponding to the smallest $c^{(i)}$ that is larger-than or equal to c. This is going to be our estimate for $\alpha(x)$. We remark that finding the values $r^{(i)}$ can be done efficiently in $O(n)$ time after sorting the data (which has a runtime of $O(n \log n)$) so the whole isotonic regression task can be done very efficiently.

Estimating $k(x)$. We now describe our process for estimating the values of $\alpha(x)$ and $k(x)$ for every example of dataset B. Similarly to the binary classification setting, we estimate the accuracy probability $\alpha(x)$ using isotonic regression on a small validation dataset. The value of $k(x)$ can be set to be equal to a fixed value of k for all data, so that the top-k accuracy of the teacher on the validation data is reasonable (say above 60%). For example, in our ImageNet experiments, we used $k = 5$. We also provide a data-dependent method to find different values $k(x)$ for every example x. To do this we adapt the method for estimating the top-1 accuracy $\alpha(x)$ of the teacher from the validation dataset. For every value of $k = 2, \ldots, L - 1$ we compute the top-k margin of the teacher's predictions on the validation data which is equal to the sum of the top-k probabilities of the teacher soft-label minus the probability assigned to the $k + 1$-th class, i.e.,

$$\mathrm{margin}_k(y_s(x)) = \Big(\sum_{i=1}^{k} \max_i y_s(x) \Big) - \max_{k+1} y_s(x) \,.$$

Using the top-k margin as the covariate and the top-k accuracy as the response we solve the corresponding regression task using isotonic regression to obtain the value $\alpha_k(x)$ representing the probability that the true label belongs in the top-k predictions of the teacher soft-label. For some threshold, say 90%, for every x we set $k(x)$ to be the smallest value of k so that $\alpha_k(x) \geq 90\%$. We empirically observed that using larger thresholds for the top-k accuracy (e.g., 90% or 95%), is better. We remark that while using the top-k margin as the covariate in the regression task is reasonable, our method can be used with other "uncertainty metrics" of the teacher's soft-labels, e.g., the entropy of the distribution of $y_s(x)$ after grouping together the top-k elements. The higher this entropy metric is the more likely that the top-k accuracy probability $\alpha(x)_k$ of the teacher is low.

B.2 SLaM for Distillation with Unlabeled Examples: Pseudocode

In this section we present pseudo-code describing the distillation with unlabeled examples setting and the SLaM method, Algorithm 1.

Remark B.1. We remark that in our experiments, we observed that not normalizing the mixing operation with $k(x) - 1$ resulted in better results overall. Therefore, the mixing operation used in our experimental evaluation of SLaM is $\mathrm{mix}(f(x;w); \alpha(x), k(x)) = \alpha(x) f(x;w) + (1 - \alpha(x))(1 - f(x;w)) * \mathrm{top}(y_s(x); k(x))$. For more details we refer the reader to the code provided in the supplementary material.

Algorithm 1 Student Label Mixing (SLaM) Distillation

```
Input: Labeled Dataset A, Labeled Validation dataset V, Unlabeled Dataset U
Output: A trained Student model $f(x; w)$

Train Teacher model on Labeled Dataset A
Pre-train Student model on Labeled Dataset A

# Label examples of Dataset U using the Teacher
$B \leftarrow \emptyset$
for each $x \in U$ do
    Add $(x, y_s(x))$ to $B$    # For hard-distillation use $y(x)$
end for

# Learn Teacher Accuracy Statistics $\alpha(x), k(x)$ Algorithm 2
$\hat{\alpha}(x), \hat{k}(x) \leftarrow \mathrm{LearnAccuracyStatistics}(y(\cdot), V, B)$
Train student $f(x; w)$ using the SLaM loss:
    $\sum_{(x,y) \in A \cup V} \ell(y, f(x;w)) + \sum_{(x,y) \in B} \ell(y, \mathrm{mix}(f(x;w); \hat{a}(x), \hat{k}(x)))$
```

C SLaM Consistency

In the following proposition we show that any minimizer of the SLaM loss over the noisy teacher-data must agree with the ground-truth for all x (that have positive density). To keep the presentation simple and avoid measurability issues (e.g., considering measure zero sets under X) in the following we will assume that the example distribution X is supported on a finite set. We remark that one can easily adapt the proof to hold for any distribution X (but the result will hold after excluding measure-zero sets under X).

Proposition C.1 (SLaM Consistency)**.** *Let D be the distribution of the teacher-labeled examples of dataset B, i.e., we first draw $x \sim X$ and then label it using the noisy teacher of Definition 3.2. Moreover, assume that there exists some parameter $w^* \in \mathcal{W}$ such that the ground-truth $g(x) = f(x; w^*)$. Denote by $\mathcal{L}^{\mathrm{SLaM}}(w) = \mathbf{E}_{(x,y)\sim D}[\ell(y, \mathrm{mix}(f(x;w); \alpha(x), k(x))]$. the SLaM objective. The following hold true.*

1. *w^* minimizes the SLaM objective.*

Algorithm 2 Estimating Teacher's Accuracy Statistics $\alpha(x), k(x)$

```
Input: (Noisy) Teacher Model $y_s(x)$, Labeled Validation dataset V,
    Isotonic-Regression lower-bound $\mathrm{lb} \in [0,1]$, and top-k accuracy threshold $t \in [0,1]$.
Output: Estimates $\hat{\alpha}(x), \hat{k}(x)$ of the actual $\alpha(x), k(x)$.

Create Soft-labels for the Validation dataset using the teacher model $\{y_s(x) : x \in V\}$.
for $j = 1$ to $L - 1$ do
    # Map $y_s(x)$ to top-j margin and accuracy pairs on the Validation V
    $C \leftarrow \left\{ \left( \sum_{r=1}^{j} \max_{r} y_s(x) - \max_{j+1} y_s(x),\ 1 - \mathrm{err}(y_s(x), z) \right) : (x,z) \in V \right\}$ .
    Set $\hat{\alpha}_j(x)$ to be the output of Isotonic-Regression with lower-bound $\mathrm{lb}$ on the (covariate, responce) pairs in $C$.    # See Appendix B.1
end for
$\hat{a}(x) \leftarrow \hat{a}_1(x)$
$\hat{a}_L(x) \leftarrow 1$    # The top-L accuracy is always (trivially) equal to 1
Given example $x$ for some threshold $t$ set $\hat{k}(x)$ to be the smallest integer $r \in \{1, \ldots, L\}$ so that $a_r(x) \geq t$.
```

2. *Assuming further that for all x it holds that $\alpha(x)k(x) \neq 1$, we have that* any *minimizer w of the SLaM objective satisfies: $f(x;w) = g(x)$ for all x.*

Proof. Fix any example $x \in X$. By Definition 3.2 we have that the corresponding teacher label y is correct with probability $\alpha(x)$ and a uniformly random incorrect label out of the top-k labels according to the teacher soft-label $y_s(x)$. Recall for an L-dimension score vector p, by $\mathrm{top}(p;k) \in \{0,1\}^L$ we denote the vector that has 1 on the positions of the top-k elements of p, e.g., $\mathrm{top}((1,2,3,4,5);2) = (0,0,0,1,1)$. Conditional on x, the corresponding expected noisy teacher label is

$$\begin{aligned}\mathbf{E}[y \mid x] &= \mathbf{P}[y = g(x) \mid x] g(x) + \mathbf{P}[y \neq g(x)] \, \mathbf{E}[y \mid x, y \neq g(x)] \\ &= \alpha(x) g(x) + (1 - \alpha(x)) \, \mathbf{E}[y \mid y \neq g(x), x] \,.\end{aligned}$$

We know that the expected teacher label conditional on it being wrong $\mathbf{E}[y \mid y \neq g(x), x]$ is a uniformly random incorrect label from the top-k labels of the corresponding teacher soft-label $y_s(x)$. Assume first that $k = L$, since the ground-truth is represented by a one-hot vector, the distribution of uniformly random incorrect labels conditional on x can be written as $(1 - g(x))/(L-1)$. For example, if the ground-truth label is $g(x) = (1,0,0,0,0)$ then a uniformly random incorrect label has probability distribution $(0, 1/4, 1/4, 1/4, 1/4)$. Assume now that $k(x) = 3$ and $\mathrm{top}(y_s(x);3) = (1,1,1,0,0)$. Then the distribution of the (incorrect) teacher label becomes $(0, 1/2, 1/2, 0, 0)$. Using $*$ to denote element-wise multiplication of two vectors, we have

$$\mathbf{E}[y \mid x, y \neq g(x)] = \frac{1 - g(x)}{k(x) - 1} * \mathrm{top}(y_s(x); k(x))$$

Therefore, we obtain

$$\mathbf{E}[y \mid x] = \alpha(x) g(x) + (1 - \alpha(x)) \frac{1 - g(x)}{k(x) - 1} * \mathrm{top}(y_s(x); k(x)) = \mathrm{mix}(g(x); \alpha(x), k(x)) \,.$$

Therefore, by using the fact that Cross-Entropy is linear in its first argument, we obtain that the expected SLaM loss on some example x is

$$\begin{aligned}\mathbf{E}[\mathrm{ce}(y, \mathrm{mix}(f(x;w); \alpha(x), k(x))) \mid x] &= \mathrm{ce}(\mathbf{E}[y \mid x], \mathrm{mix}(f(x;w); \alpha(x), k(x))) \\ &= \mathrm{ce}(\mathrm{mix}(g(x;w); \alpha(x), k(x)), \mathrm{mix}(f(x;w); \alpha(x), k(x))) \,.\end{aligned}$$

We first have to show that there exist some parameter $w \in \mathcal{W}$ that matches the (expected) observed labels $\mathbf{E}[y \mid x]$. Observe first that by using the realizability assumption, i.e.,that there exists w^* so that $f(x;w^*) = g(x)$ we obtain that, for every x, it holds $\mathrm{mix}(g(x); \alpha(x), k(x)) = \mathrm{mix}(f(x;w^*); \alpha(x), k(x))$. In fact, by Gibb's inequality (convexity of Cross-Entropy) we have that w^* is a (global) minimizer of the SLaM objective.

We next show that *any (global) minimizer* of the SLaM objective must agree with the ground-truth for every x. Since we have shown that w^* is able to match the (expected) labels $\mathbf{E}[y \mid x]$ any other minimizer w must also satisfy $\mathrm{mix}(g(x); \alpha(x), k(x)) = \mathrm{mix}(f(x; w); \alpha(x), k(x)))$. Assume without loss of generality that $g_0 = 1$, i.e., the ground-truth label is 0. We observe that by using that $\mathrm{mix}(g(x; w); \alpha(x), k(x)) = \alpha(x)g(x) + (1 - \alpha(x))\frac{1-g(x)}{k(x)-1} * \mathrm{top}(y_s(x); k(x))$ and the fact that the ground-truth belongs in the top-$k(x)$ of the teacher's predictions conditional that the teacher's top-1 prediction is incorrect (thus $\mathrm{top}(y_s(x))_0 = 1$), we obtain that

$$\alpha(x)g_0(x)+(1-\alpha(x))(1-g_0(x))/(1-k(x)) = \alpha(x)f(x;w)_0+(1-\alpha(x))(1-f(x;w)_0)/(k(x)-1)\,.$$

Using the fact that $g_0 = 1$ we can simplify the above expression to

$$(1 - f(x; w)_0)\left(\alpha(x) - \frac{1 - \alpha(x)}{k(x) - 1}\right) = 0\,.$$

Using the assumption that $a(x)k(x) \neq 1$ we obtain that the term $\left(\alpha(x) - \frac{1-\alpha(x)}{k(x)-1}\right)$ is not vanishing and therefore it must hold that $f(x; w)_0 = 1 = g_0$, i.e., the student model must be equal to the ground-truth.

$\square$

D Extended Experimental Evaluation

We implemented all algorithms in Python and used the TensorFlow deep learning library [1]. We ran our experiments on 64 Cloud TPU v4s each with two cores.

D.1 Implementation Details: Vision Datasets

Here we present the implementation details for the vision datasets we considered.

Remark D.1. We note that in all our experiments, "VID" corresponds to the implementation of the loss described in equation (2), (4) and (6) of [2] (which requires appropriately modifying the student model so that we have access to its embedding layer).

Experiments on CIFAR-{10/100} and CelebA For the experiments on CIFAR-10/100 and CelebA we use the Adam optimizer with initial learning rate $\mathrm{lr} = 0.001$. We then proceed according to the following learning rate schedule (see, e.g., [25]):

$$\mathrm{lr} \leftarrow \begin{cases} \mathrm{lr} \cdot 0.5 \cdot 10^{-3}, & \text{if \#epochs} > 180 \\ \mathrm{lr} \cdot 10^{-3}, & \text{if \#epochs} > 160 \\ \mathrm{lr} \cdot 10^{-2}, & \text{if \#epochs} > 120 \\ \mathrm{lr} \cdot 10^{-1}, & \text{if \#epochs} > 80 \end{cases}$$

Finally, we use data-augmentation. In particular, we use random horizontal flipping and random width and height translations with width and height factor, respectively, equal to 0.1.

The hyperparameters of each method are optimized as follows. For SLaM we always use 0.5 as the lower bound for isotonic regression (i.e., the parameter lb in Algorithm 2). As CelebA is a binary classification benchmark $k(x)$ is naturally set to 2 for all examples. For CIFAR-10/10 we used the data-dependent method for estimating $k(x)$ (see Algorithm 2) with threshold parameter $t = 0.9$. For weighted distillation we do a grid search over updating the weights every $\{1, 25, 50, 100, 200\}$ epochs and we report the best average accuracy achieved. Finally, for VID we search over $\{0.001, 0.1, 0.2, 0.5, 0.8, 1.0, 2.0, 10.0, 50.0, 100.0\}$ for the coefficient of the VID-related term of the loss function, and for the PolyLoss we optimize its hyperparameter over $\{-1.0, -0.8, -0.6, -0.4, -0.2, 0.5, 1.0, 2.0, 50.0, 100.0\}$.

Experiments on ImageNet For the ImageNet experiments we use SGD with momentum 0.9 as the optimizer. For data-augmentation we use random horizontal flipping and random cropping. Finally, the learning rate schedule is as follows. For the first 5 epochs the learning rate lr is increased from

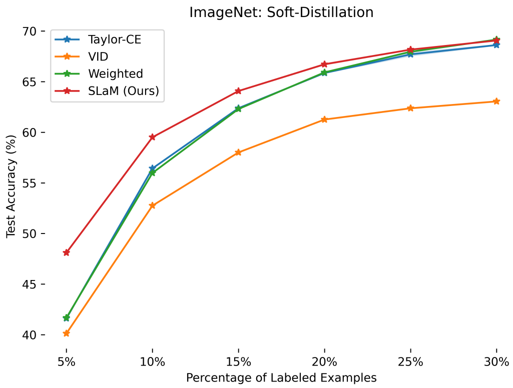

Figure 3: Comparison of distillation methods on ImageNet. On the horizontal axis we plot the size of Dataset A as a percentage of the whole training dataset. On the vertical axis we plot the accuracy of the trained student-model on the test dataset.

Table 5: Experiments on CIFAR-10 (**hard**-distillation). See Section 4.2 for details.

Labeled Examples	5000	7500	10000	12500	15000	17500
Teacher	61.30	68.98	72.42	73.92	76.63	78.63
Vanilla	62.26 ± 0.45	69.07 ± 0.11	72.09 ± 0.11	73.43 ± 0.16	75.93 ± 0.25	77.43 ± 0.15
Taylor-CE [20]	63.14 ± 0.07	69.98 ± 0.11	72.72 ± 0.36	73.77 ± 0.28	76.26 ± 0.29	77.88 ± 0.20
UPS [48]	64.27 ± 0.08	70.93 ± 0.26	73.78 ± 0.16	74.66 ± 0.29	77.38 ± 0.37	78.95 ± 0.08
VID [3]	61.95 ± 0.22	66.91 ± 0.21	69.59 ± 0.24	72.16 ± 0.47	74.83 ± 0.11	75.55 ± 0.21
Weighted [27]	63.22 ± 0.45	71.04 ± 0.26	72.84 ± 0.12	74.20 ± 0.16	76.56 ± 0.24	78.23 ± 0.15
SLaM (Ours)	$\mathbf{66.40 \pm 0.31}$	$\mathbf{72.44 \pm 0.17}$	$\mathbf{74.77 \pm 0.13}$	$\mathbf{75.64 \pm 0.19}$	$\mathbf{77.99 \pm 0.36}$	$\mathbf{79.26 \pm 0.26}$

0.0 to 0.1 linearly. After that, the learning rate changes as follows:

$$\text{lr} = \begin{cases} 0.01, & \text{if } \#\text{epochs} > 30 \\ 0.001, & \text{if } \#\text{epochs} > 60 \\ 0.0001, & \text{if } \#\text{epochs} > 80\,. \end{cases}$$

The hyperparameters of each method are optimized as follows. For SLaM we do a hyperparameter search over $\{0.55, 0.60, 0.65, 0.70\}$ for the lower bound for isotonic regression, and we keep the best performing value for each potential size of dataset A. We used the fixed value 5 for $k(x)$, as the top-5 accuracy of the teacher model was satisfactory (much higher than its top-1 accuracy) on the validation dataset. For Taylor-CE we did a hyper-parameter search for the Taylor series truncation values in $\{1, 2, 3, 4, 5, 6, 10, 20, 50, 80, 100\}$. For weighted distillation we compute the weights in a one-shot fashion using the pre-trained student (as in the ImageNet experiments in [27]). For VID we search over $\{0.1, 0.3, 0.5\}$ for the coefficient of the VID-related term of the loss function, and for the PolyLoss we optimize its hyperparameter over $\{1.0, 2.0, 50.0, 100.0\}$.

D.2 Hard-Distillation

Here we present results on hard-distillation. The hyper-parameters of all methods are chosen the same way as in our soft-distillation experiments, see Appendix D.1. Tables 5, 6 and 7 contain our results on CIFAR-10, CIFAR-100 and CelebA, respectively. We observe that in almost all cases, SLaM consistently outperforms the other baselines. Moreover, for CIFAR-10 and CIFAR-100 hard-distillation performs worse than soft-distillation (as it is typical the case) but in CelebA hard-distillation seems to be performing on par with (sometimes even outperforming) soft-distillation. A plausible explanation for the latter outcome is that in our CelebA experiments the teacher and student have different architectures (MobileNet and ResNet, respectively) so that soft-labels from the teacher are not so informative for the student. (This is also a binary classification task where the information passed from the teacher to the student through its soft-labels is limited.)

Table 6: Experiments on CIFAR-100 (**hard**-distillation). See Section 4.2 for details.

Labeled Examples	5000	7500	10000	12500	15000	17500
Teacher	35.97	44.65	49.62	55.68	59.19	62.05
Vanilla	36.36 ± 0.04	44.15 ± 0.10	50.22 ± 0.07	55.55 ± 0.24	58.85 ± 0.1	61.43 ± 0.19
Taylor-CE [20]	39.12 ± 0.14	46.87 ± 0.10	52.64 ± 0.22	57.19 ± 0.28	59.95 ± 0.11	62.36 ± 0.21
UPS [48]	39.49 ± 0.13	48.36 ± 0.44	53.95 ± 0.10	57.95 ± 0.10	60.59 ± 0.29	62.09 ± 0.28
VID [3]	37.19 ± 0.09	44.67 ± 0.16	50.63 ± 0.35	54.78 ± 0.07	59.27 ± 0.14	62.01 ± 0.05
Weighted [27]	38.04 ± 0.29	46.45 ± 0.22	52.33 ± 0.18	57.43 ± 0.13	60.81 ± 0.09	63.02 ± 0.06
SLaM (Ours)	$\mathbf{42.01 \pm 0.29}$	$\mathbf{49.08 \pm 0.14}$	$\mathbf{54.49 \pm 0.17}$	$\mathbf{58.53 \pm 0.04}$	$\mathbf{61.12 \pm 0.15}$	$\mathbf{63.21 \pm 0.18}$

Table 7: Experiments on CelebA (**hard**-distillation). See Section 4.2 for details.

Labeled Examples	2%	3%	4%	5%	6%	7%
Teacher	86.19	88.25	88.95	91.31	92.09	92.62
Vanilla	89.73 ± 0.08	91.61 ± 0.09	92.05 ± 0.11	93.41 ± 0.13	94.02 ± 0.15	94.05 ± 0.04
Taylor-CE [20]	$\mathbf{90.62 \pm 0.05}$	92.19 ± 0.02	92.66 ± 0.11	93.60 ± 0.14	94.00 ± 0.04	94.38 ± 0.10
UPS [48]	89.35 ± 0.04	91.30 ± 0.04	91.95 ± 0.12	93.18 ± 0.07	93.71 ± 0.04	94.18 ± 0.03
VID [3]	89.92 ± 0.21	91.60 ± 0.11	92.20 ± 0.12	93.51 ± 0.15	94.08 ± 0.15	94.27 ± 0.10
Weighted [27]	90.06 ± 0.06	91.97 ± 0.13	92.45 ± 0.10	93.60 ± 0.07	93.94 ± 0.12	94.25 ± 0.16
SLaM (Ours)	90.43 ± 0.05	$\mathbf{92.25 \pm 0.11}$	$\mathbf{92.71 \pm 0.08}$	$\mathbf{93.96 \pm 0.17}$	$\mathbf{94.39 \pm 0.21}$	$\mathbf{94.52 \pm 0.12}$

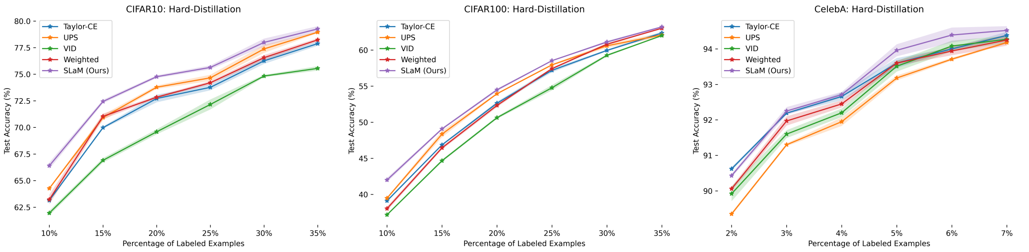

Figure 4: Comparison of distillation methods on CIFAR-10,100 and CelebA. On the horizontal axis we plot the size of Dataset A as a percentage of the whole training dataset. On the vertical axis we plot the accuracy of the trained student-model on the test dataset.

D.3 Large Movies Reviews Dataset Results

Here we present the results and the implementation details regarding the experiments on the Large Movies Reviews dataset. Recall that we use an ALBERT-large model as a teacher, and an ALBERT-base model as a student. We also use $2\%, 4\%, 8\%, 40\%$ percent (or 500, 1000, 2000, 10000 examples) from the training dataset and split the remaining data in a validation dataset of 500 examples and an unlabeled dataset U. We compare the methods on the soft-distillation. For each trial we train the student model for 40 epochs and keep the best test accuracy over all epochs. We perform 3 trials and report the average of each method and the variance of the achieved accuracies over the trials. The results of our experiments can be found in Table 8. We remark that we did not implement the UPS method for this dataset as the data-augmentation method for estimating the teacher's accuracy could not be readily used for this NLP dataset. Moreover, using dropout and Monte Carlo estimation for the uncertainty was also not compatible with the Albert model used in this experiment.

Since we are dealing with ALBERT-models (which are already pre-trained), we do not pre-train the student model on dataset A except in the case of "weighted-distillation" [27], where we pre-train the student model on dataset A just for 1 epoch. The teacher model is trained using the Adam optimizer for 20 epochs with initial learning rate 10^{-6}. The student model is trained also using the Adam optimizer but for 40 epochs and with learning rate 10^{-7}.

The hyperparameters of each method are optimized as follows. For SLaM we do a hyperparameter search over $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ for the lower bound for isotonic regression, and we keep the best performing value for each potential size of dataset A. As this is a binary classification benchmark we naturally set $k(x) = 2$ for all examples. For weighted distillation we do a grid search over updating the weights every $\{1, 10, 20, 40\}$ epochs and, similarly, we report the best average accuracy achieved. Finally, for VID (recall also Remark D.1) we search over $\{0.1, 0.5, 1.0, 2.0\}$ for the coefficient of

Table 8: Experiments on the Large Movies Reviews Dataset (**soft**-distillation). See Section D.3 for details.

Labeled Examples	2%	4%	8%	40%
Teacher	77.52	84.04	85.44	88.3
Vanilla	80.93 ± 0.10	85.12 ± 0.29	85.99 ± 0.08	87.50 ± 0.6
Taylor-CE [20]	79.5 ± 0.38	85.14 ± 0.13	85.98 ± 0.14	87.57 ± 0.3
VID [3]	81.76 ± 0.32	85.33 ± 0.35	86.17 ± 0.06	87.71 ± 0.01
Weighted [27]	$81.1 \pm +0.1$	85.2 ± 0.05	86.13 ± 0.17	$\mathbf{87.8 \pm 0.25}$
SLaM (Ours)	$\mathbf{81.88 \pm 0.23}$	$\mathbf{85.5 \pm 0.09}$	$\mathbf{86.23 \pm 0.13}$	87.73 ± 0.38

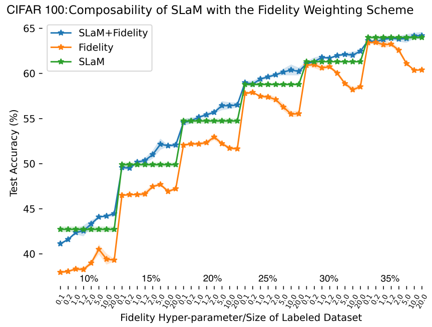

Figure 5: Composability the fidelity-based weighting scheme of [17]. The x-axis shows the different values of the fidelity hyper-parameter β and the size of dataset A. From left to right we increase the size of dataset A from 10% to 35% and for each size we try different values of β. We observe that SLaM on its own (shown in green) is usually much better than the fidelity weighting scheme (shown in orange). Moreover, using SLaM on top of the fidelity weighting scheme (shown in blue) consistently improves its performance.

the VID-related term of the loss function, and for the PolyLoss we opitmize its hyperparameter over $\{-1.0, -0.8, -0.6, -0.4, -0.2, 0.5, 1.0, 2.0\}$.

D.4 Combining with Teacher-Uncertainty-Based Reweighting Techniques

As we discussed in Section 2, our method can in principle be combined with teacher-uncertainty filtering and weighting schemes as these can be seen as preprocessing steps. To demonstrate this, we combine our method with the so-called fidelity-based weighting scheme of [17]. The fidelity weighting scheme reweights examples using some uncertainty measure for teacher's labels, e.g., by performing random data-augmentations and estimating the variance of the resulting teacher labels or using dropout and Monte Carlo estimation. More precisely, for every example x in the teacher-labeled dataset B, the fidelity-weighting scheme assigns the weight $w^{\mathrm{Fid}}(x) = \exp(-\beta \text{ uncertainty}^{\text{teacher}}(x))$ for some hyper-parameter $\beta > 0$. In our experiments we performed 10 random data augmentations (random crop and resize), estimated the coordinate-wise variance of the resulting teacher soft-labels, and finally computed the average of the variances of the k-classes, as proposed in [17]. We normalized the above uncertainty of each example by the total uncertainty of the teacher over the whole dataset B. The weights of examples in dataset A are set to 1 and the reweighted objective is optimized over the combination of the datasets A, B.

$$\mathcal{L}^{\mathrm{fid}}(w) = \frac{1}{|A \cup B|} \left(\sum_{(x,y)\in A} \ell(y, f(x; w)) + \sum_{(x,y)\in B} w^{\mathrm{Fid}}(x)\, \ell(y, f(x; w)) \right). \tag{3}$$

To demonstrate the composability of our method with such uncertainty-based weighting schemes, we use CIFAR100 and the percentage of the labeled dataset A (as a fraction of the whole training set) is $10\%, 15\%, 20\%, 25\%, 30\%, 35\%$, similar to the setting of Section 4.2. The teacher is a ResNet110 and the student is a ResNet56. We first train the student using only the fidelity weighting scheme, i.e., optimize the loss function of Equation (4) using different values for the hyperparameter $\beta \in \{0.1, 0.2, 1.0, 1.2, 2.0, 5.0, 10.0, 20.0\}$, i.e., ranging from mildly reweighting the examples of

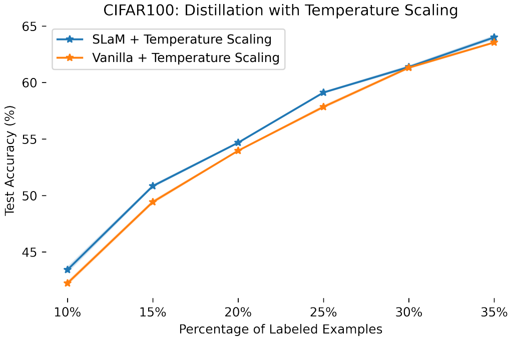

Figure 6: CIFAR100: Temperature Ablation. On the x-axis we have the size of the labeled dataset (as a percentage of the whole training dataset) that the teacher model uses for training.

dataset B to more agressively "removing" examples where the teacher's entropy is large. For the same values of β we then train the student using the reweighted SLaM objective:

$$\mathcal{L}^{\text{Fid+SLaM}}(w) = \frac{1}{|A \cup B|}\left(\sum_{(x,y)\in A} \ell(y, f(x;w)) + \sum_{(x,y)\in B} w^{\text{fid}}(x)\, \ell(y, \text{mix}(f(x;w); \alpha(x), k(x))) \right). \tag{4}$$

For the combined SLaM + Fidelity method we did not perform hyper-parameter search and used the same parameters for the isotonic regression as we did in the "standard" SLaM experiment in CIFAR100 of Appendix D.1. We present our comprehensive results for all sizes of dataset A and values of the hyper-parameter β in Figure 5. Our results show that, regardless of the value of the hyperparameter β and the size of the labeled dataset A, using SLaM together with the fidelity weighting scheme provides consistent improvements. Moreover, in Figure 5, we observe that by using SLaM the achieved accuracy depends less on the hyper-parameter β: since SLaM takes into account the fact that some of the teacher's predictions are incorrect, it is not crucial to down-weight them or filter them out.

D.5 Using Distillation Temperature

In this section we show that our approach can be effectively combined with temperature-scaling [26]. Choosing the right distillation temperature often provides significant improvements. In our setting, the teacher provides much more confident predictions (e.g., soft-labels with high-margin) on dataset A (where the teacher was trained) compared to the teacher soft-labels of dataset B where the teacher is, on average, less confident. Given this observation, it is reasonable to use different distillation temperatures for dataset A and dataset B. We try different temperatures for dataset A and dataset B and perform vanilla distillation with temperature and also consider applying the temperature scaling before applying SLaM. For each size of dataset A we try pairs of temperatures $t_A, t_B \in \{0.01, 0.1, 0.5, 0.8, 1., 2., 5., 10., 100.\}$ and report the best accuracy achieved by vanilla distillation and the best achieved by first applying temperature scaling and then SLaM. In Figure 6 we observe that SLaM with temperature scaling consistently improves over vanilla distillation with temperature.

D.6 Using SLaM with other loss functions beyond cross-entropy

In this section, we demonstrate that our method can be successfully applied when the student loss function comes from the families of losses introduced in [20] and [35]. We perform experiments on CIFAR-100 and ImageNet following the setting of Section 4.2. In particular, we compare vanilla distillation with unlabeled examples using the Taylor-CE loss of [20] and the PolyLoss of [35], with combining SLaM with these losses. For the Taylor-CE loss we set the "degree" hyperparameter to be 2 (as suggested in [20]) and we set the hyperparameter of the PolyLoss to be 2.0 (as suggested in [35]). The corresponding results can be found in Figure 7.

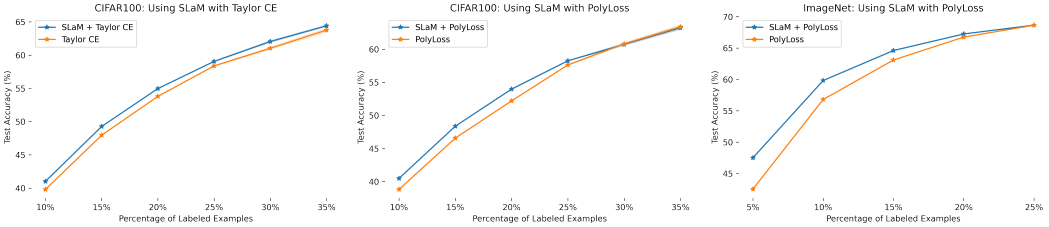

Figure 7: Using SLaM with PolyLoss [35] and Taylor CE [20]. On the x-axis we have the size of the labeled dataset (as a percentage of the whole training dataset) that the teacher model uses for training. See Appendix D.6 for more details.

E Distilling Linear Models and Learning Noisy Halfspaces

In this section we state and prove our convergence result for the SLaM method when applied to linear models. Our assumption is that the ground-truth $g(x)$ corresponds to a halfspace, i.e., $g(x) = (\mathbf{1}\{w^* \cdot x > 0\}, \mathbf{1}\{w^* \cdot x \leq 0\})$ for some unknown weight vector w^*. We show that using SLaM with a linear model as the student will recover the ground truth classifier. We make the standard assumption that the ground-truth halfspace has γ-margin, i.e., that $\|w^*\|_2 = 1$ and that it holds $|w^* \cdot x| \geq \gamma$ for all examples x. For a fixed example x, the observed noisy teacher-label y satisfies Definition 3.2, i.e., $y = g(x)$ w.p. $\alpha(x)$ and $y = 1 - g(x)$ w.p. $1 - \alpha(x)$ (since $k = 2$ for binary classification). Our approach consists of using the standard cross-entropy loss $\mathrm{ce}(p, q)$ and training a student-model consisting of a linear layer plus a soft-max activation, i.e.,

$$f(x; w) = (f_0(x; w), f_1(x; w)) = \left(\frac{1}{1 + e^{-w \cdot x}}, \frac{e^{-w \cdot x}}{1 + e^{-w \cdot x}} \right) .$$

Recall, that for binary classification, we define the mixing operation as

$$\mathrm{mix}(f(x; w); \alpha(x)) = \alpha(x) f(x; w) + (1 - \alpha(x))(1 - f(x; w)) .$$

Algorithm 3 SLaM for Linear Models

```
Initialiaze weight vector of student $w^{(0)} \leftarrow 0$
for $t = 1, \ldots, T$ do
    Draw example $x^{(t)} \sim X$.
    Label $x^{(t)}$ with (noisy) teacher to obtain $y^{(t)}$
    Compute the gradient of the SLaM loss at $(x^{(t)}, y^{(t)})$:
        $g^{(t)} \leftarrow \partial_w \mathrm{ce}(y^{(t)}, \mathrm{mix}(f(x^{(t)}; w^{(t-1)}), \alpha(x^{(t)})) \mid_{w=w^{(t-1)}}$
    Compute step size: $\lambda^{(t)} \leftarrow 1/r(f(x^{(t)}; w^{(t-1)}), \alpha(x^{(t)}))$ (see Lemma E.3 for the definition of $r(\cdot, \cdot)$).
    Update the student model: $w^{(t)} \leftarrow w^{(t-1)} - \lambda^{(t)} \, g^{(t)}$
end for
```

Theorem E.1 (Student Label Mixing Convergence)**.** *Let X be a distribution on $\mathbf{R}^d$ and $g(x)$ be the ground-truth halfspace with normal vector $w^* \in \mathbf{R}^d$. Let D be the distribution over (noisy) teacher-labeled examples (x, y) whose x-marginal is X. We denote by $\alpha(x)$ the probability that the teacher label $y \in [0, 1]^2$ is correct, i.e., $\alpha(x) = \mathbf{P}_{(x,y)\sim D}[\mathrm{argmax}(y) = g(x) \mid x]$. Assume that there exist $\beta, \gamma > 0$ such that for all examples x in the support of X it holds that $|w^* \cdot x| \geq \gamma$ and $|1/2 - \alpha(x)| \leq \beta$. Let $\epsilon > 0$. After $T = O(1/(\beta^2 \gamma^2 \epsilon^2))$ iterations of SLaM (Algorithm 3), with probability at least 99%, there exists an iteration $t \leq T$ where $\mathbf{P}_{x\sim X}[\mathrm{err}(f(x; w^{(t)}), g(x))] \leq \epsilon$.*

Remark E.2 (High-Probability Result). We remark that even though our learner succeeds with constant probability (at least $\%99$) we can amplify its success probability to $1 - \delta$ by standard amplification techniques (i.e., by repeating the algorithm $O(\log(1/\delta))$ times and keeping the best result). To achieve success probability $1 - \delta$ the total sample complexity is $O(\log(1/\delta)/(\epsilon^2 \gamma^2 \beta^2))$.

Proof. We first provide simplified expressions for the gradient of the SLaM objective and the update vectors $\lambda^{(t)} g^{(t)}$ used in Algorithm 3. In what follows we remark that for any binary classification model $f(x;w) = (f_0(x;w), f_1(x;w))$ we have the following identities: (i) $(\mathrm{mix}(f(x;w);\alpha(x)))_0 = \mathrm{mix}(f_0(x;w);\alpha(x))$, where to simplify notation we overload the mixing operation to also act on the scalar $f_0(x;w)$, i.e., $\mathrm{mix}(f_0(x;w);\alpha(x)) = \alpha(x) f_0(x;w) + (1-\alpha(x))(1-f_0(x;w))$; and (ii) $f_1(x;w) = 1 - f_0(x;w)$.

Lemma E.3 (SLaM Gradient)**.** *The gradient of the SLaM objective is equal to*

$$\partial_w \mathrm{ce}(y, \mathrm{mix}(f(x;w);\alpha(x)) = r(f_0(x;w);\alpha(x))\ \mathrm{sgn}(2\alpha(x)-1)\ ((\mathrm{mix}(f_0(x;w);\alpha(x)) - y_0)x,$$

where

$$r(f(x;w);\alpha(x)) = \frac{f_0(x;w)(1-f_0(x;w))}{\mathrm{mix}(f_0(x;w);\alpha(x))(1-\mathrm{mix}(f_0(x;w),\alpha(x)))}\ |2\alpha(x)-1|$$

Let $L(x;w) = \mathbf{E}_{(x,y)\sim D}[\mathrm{ce}(y, \mathrm{mix}(f(x;w),\alpha(x)) \mid x]$ be the expected student label mixing loss conditional on some example $x \in \mathbf{R}^d$. It holds $\partial_w L(x;w) = r(f(x;w),\alpha(x))\ |2\alpha(x)-1|\ (f_0(x;w) - g_0(x))\ x$.

Proof. We first show the formula

$$\partial_w \mathrm{ce}(y, \mathrm{mix}(f(x;w),\alpha(x)) = r(f_0(x;w),\alpha(x))\ \mathrm{sgn}(2\alpha(x)-1)\ ((\mathrm{mix}(f_0(x;w),\alpha(x)) - y_0)x\,. \tag{5}$$

Using the chain rule, we obtain

$$\begin{aligned}\partial_w \mathrm{ce}(y,&\mathrm{mix}(f(x;w);\alpha(x)) = \\ &- \frac{y_0}{\mathrm{mix}(f_0(x;w),\alpha(x))}\partial_w(\mathrm{mix}(f_0(x;w);\alpha(x)) \\ &- \frac{y_1}{\mathrm{mix}(f_1(x;w),\alpha(x))}\partial_w(\mathrm{mix}(f_1(x;w);\alpha(x))\,.\end{aligned}$$

Now we observe that that for binary classification, it holds that $y_1 = 1 - y_0$, $\mathrm{mix}(f_1(x;w);\alpha(x)) = 1 - \mathrm{mix}(f_0(x;w);\alpha(x))$, and therefore, also $\partial_w \mathrm{mix}(f(x;w);\alpha(x))_1) = -\partial_w \mathrm{mix}(f(x;w);\alpha(x))_0)$ to obtain the simplified expression:

$$\begin{aligned}\partial_w \mathrm{ce}(y,&\mathrm{mix}(f(x;w);\alpha(x)) = \\ &- \frac{y_0}{\mathrm{mix}(f_0(x;w),\alpha(x))}\partial_w(\mathrm{mix}(f_0(x;w);\alpha(x)) \\ &+ \frac{1-y_0}{1-\mathrm{mix}(f_0(x;w),\alpha(x))}\partial_w(\mathrm{mix}(f_0(x;w);\alpha(x))\,.\end{aligned}$$

Further simplifying the above expression, we obtain:

$$\begin{aligned}&\partial_w \mathrm{ce}(y, \mathrm{mix}(f(x;w);\alpha(x)) = \\ &= \frac{\mathrm{mix}(f_0(x;w),\alpha(x)) - y_0}{\mathrm{mix}(f_0(x;w),\alpha(x))\ (1-\mathrm{mix}(f_0(x;w),\alpha(x)))}\partial_w(\mathrm{mix}(f_0(x;w);\alpha(x))\,.\end{aligned}$$

Using again the chain rule we obtain that

$$\partial_w(\mathrm{mix}(f_0(x;w);\alpha(x)) = \alpha(x)\partial_w(f_0(x;w)) + (1-\alpha(x))\partial_w(1-f_0(x;w)) = (2\alpha(x)-1)\ \partial_w f_0(x;w)\,.$$

Using the fact that the derivative of the sigmoid function $r(t) = 1/(1+e^{-t})$, is $r'(t) = e^{-t}/(1-e^{-t})^2 = r(t)(1-r(t))$, and the chain rule, we obtain that $\partial_w f_0(x;w) = f_0(x;w)(1-f_0(x;w))x$. Putting everything together we obtain the claimed formula for $\partial_w \mathrm{ce}(y, \mathrm{mix}(f(x;w);\alpha(x)))$.

To obtain the gradient formula for the expected loss conditional on some fixed example x, we can use the fact that $\partial_w \mathbf{E}[\mathrm{ce}(y, \mathrm{mix}(f(x;w);\alpha(x))) \mid x] = \mathbf{E}[\partial_w \mathrm{ce}(y, \mathrm{mix}(f(x;w);\alpha(x))) \mid x]$. Now using the formula of Equation (5) and the fact that $\mathbf{E}[y_0 \mid x] = \mathrm{mix}(g_0(x);\alpha(x))$ by the definition of our noise model, we obtain that

$$\begin{aligned}\partial_w L(x;w) &= r(f_0(x;w);\alpha(x))\mathrm{sgn}(2\alpha(x)-1)(\mathrm{mix}(f_0(x;w);\alpha(x)) - \mathrm{mix}(g_0(x);\alpha(x))) \\ &= r(f_0(x;w);\alpha(x))(2\alpha(x)-1)(f_0(x;w) - g_0(x))\end{aligned}$$

□

We first show the following claim proving that after roughly $T = 1/(\beta^2\gamma^2\epsilon^2)$ gradient iterations the student parameter vector $w^{(t)}$ will have good correlation with the ground-truth vector w^*.

Claim 1. Fix any T larger than a sufficiently large constant multiple of $\log(1/\delta)/(\epsilon^2\gamma^2\beta^2)$, and assume that for all $t \le T$ it holds that $\mathbf{P}_{x\sim X}[\mathrm{err}(f(x; w^{(t)}), g(x))] > \epsilon$. Then, we have $w^{(T)} \cdot w^* = \Omega(\beta\gamma\epsilon)\, T$, with probability at least $1 - \delta$.

Proof. Denote by $u^{(t)} = -\lambda^{(t)} g^{(t)}$ the update vector used in Algorithm 3. We observe that the weight vector at round T is equal to $w^{(T)} = \sum_{t=1}^{T} u^{(t)}$. In what follows we denote by $\mathcal{F}^{(t)}$ the filtration corresponding to the randomness of the updates of Algorithm 3. We define the martingale $q^{(T)} = \sum_{t=1}^{T} (\mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] - u^{(t)})$ with $q^{(0)} = 0$. We first show that under the assumption that $\mathbf{P}_{x\sim X}[\mathrm{argmax}(f(x; w^{(t)})) \ne g(x)] > \epsilon$, for all $t \le T$, it holds that $\sum_{t=1}^{T} \mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] \cdot w^* \ge (\epsilon\gamma\beta/2)\, T$. Using the SLaM gradient expression of Lemma E.3 and the definition of the step size $\lambda^{(t)}$ we obtain that $\mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] = \mathbf{E}_{x\sim X}[|2\alpha(x) - 1|\, (g_0(x) - f_0(x; w^{(t-1)}))\, x]$. Take any step t. We have that

$$\begin{aligned} \mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] \cdot w^* &= \mathbf{E}_{x\sim X}[|2\alpha(x) - 1|\, (g_0(x) - f_0(x; w^{(t-1)}))\, (x \cdot w^*)] \\ &= \mathbf{E}_{x\sim X}[|2\alpha(x) - 1|\, |g_0(x) - f_0(x; w^{(t-1)})|\, |x \cdot w^*|]\,, \end{aligned}$$

where we used the fact that $(g_0(x) - f_0(x; w^{(t-1)}))\, \mathrm{sgn}(x \cdot w^*) = |g_0(x) - f_0(x; w^{(t-1)})|$. Now, using the γ-margin assumption of the distribution D and the fact that $|2\alpha(x) - 1| \ge \beta$ we obtain

$$\begin{aligned} \mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] \cdot w^* &\ge \beta\gamma\ \mathbf{E}_{x\sim X}[|g_0(x) - f_0(x; w^{(t-1)})|] \\ &\ge \beta\gamma\ \mathbf{E}_{x\sim X}[|g_0(x) - f_0(x; w^{(t-1)})|\, \mathrm{err}(g(x), f(x; w^{(t-1)}))] \\ &\ge (\beta\gamma/2)\ \mathbf{P}_{x\sim X}[\mathrm{err}(g(x), f(x; w^{(t-1)}))] \ge \beta\gamma\epsilon/2\,, \end{aligned}$$

where for the penultimate inequality we used the fact that when $g(x)$ and $f(x; w^{(t-1)})$ disagree it holds that $|g_0(x) - f_0(x; w^{(t-1)})| \ge 1/2$. Take, for example, the case where $g_0(x) = 1$. Then $f_0(x; w^{(t-1)})$ must be smaller than $1/2$ otherwise the prediction of the model $\mathrm{argmax}\, f(x; w^{(t-1)})$ would also be 0 (and would agree with the prediction of $g(x)$). Finally, for the last inequality we used the fact that, by our assumption, it holds that $\mathbf{P}_{x\sim X}[\mathrm{err}(g(x), f(x; w^{(t-1)}))] \ge \epsilon$. Therefore, we conclude that $\sum_{t=1}^{T} \mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] \cdot w^* \ge (\epsilon\gamma\beta/2)\, T$. Next, we shall show that $w^{(T)}$ also achieves good correlation with the optimal direction w^* with high probability. We will use the fact that $q^{(t)}$ is a martingale and the Azuma-Hoeffding inequality to show that $w^{(T)} \cdot w^*$ will not be very far from its expectation.

Lemma E.4 (Azuma-Hoeffding)**.** *Let $\xi^{(t)}$ be a martingale with bounded increments, i.e., $|\xi^{(t)} - \xi^{(t-1)}| \le M$. It holds that $\mathbf{P}[\xi^{(T)} \ge \xi^{(0)} + \lambda] \le e^{-\lambda^2/(2M^2T)}$.*

Recall that from Lemma E.3 we have that $\mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] = \mathbf{E}_{x\sim X}[|2\alpha(x) - 1|\, (g_0(x) - f_0(x; w^{(t-1)}))\, x]$ and

$$u^{(t)} = \mathrm{sgn}(2\alpha(x^{(t)}) - 1)\, (y_0^{(t)} - \mathrm{mix}(f_0(x^{(t)}; w^{(t-1)}), \alpha(x^{(t)}))\, x^{(t)}\,.$$

Observe that since $\|x\|_2 \le 1$ for all x it holds that $\|u^{(t)}\|_2 \le 1$. Therefore, the difference $\|\,\mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] - u^{(t)}\| \le 2$ with probability 1. Since $\|w^*\|_2 = 1$, using Cauchy-Schwarz, we also obtain that $|\,\mathbf{E}[u^{(t)} \cdot w^* \mid \mathcal{F}^{(t-1)}] - u^{(t)} \cdot w^*| \le 2$.

Using Lemma E.4, and the fact that $q^{(0)} = 0$ we obtain that $\mathbf{P}[q^{(t)} \cdot w^* \ge (\beta\gamma\epsilon/4)\, T] \le e^{-\beta^2\gamma^2\epsilon^2 T/128}$. Therefore we conclude that for any T larger than $128 \log(1/\delta)/(\beta^2\gamma^2\epsilon^2)$, with probability at least $1 - \delta$, it holds that $q^{(T)} \cdot w^* \ge (\beta\gamma\epsilon/4) T$ or equivalently $w^{(T)} \cdot w^* \ge (\beta\gamma\epsilon/4)\, T$, where we used our previously obtained bound for the expected updates $\sum_{t=1}^{T} \mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] \cdot w^* \ge (\beta\gamma\epsilon/2)\, T$.

$\square$

Claim 2. Fix any $T \ge 1$. Then, we have $\|w^{(T)}\|_2 = O(\sqrt{T})$, with probability at least 99%.

Proof. We have that $\|w^{(T)}\|_2^2 = \|w^{(T-1)}\|_2^2 + 2u^{(T)} \cdot w^{(T-1)} + \|u^{(T)}\|_2^2$. Unrolling the iteration, we obtain that

$$\|w^{(T)}\|_2^2 = 2\sum_{t=1}^{T} u^{(t)} \cdot w^{(t-1)} + \sum_{t=1}^{T} \|u^{(t)}\|_2^2 \le 2\sum_{t=1}^{T} u^{(t)} \cdot w^{(t-1)} + T\,, \tag{6}$$

where we used the fact that, since $\|x^{(t)}\|_2 \le 1$, it holds that $\|u^{(t)}\|_2 \le 1$ (see the proof of Claim 1). We first show that $\sum_{t=1}^{T} \mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] \cdot w^{(t-1)} = O(T)$. We have

$$\begin{aligned}\mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] \cdot w^{(t-1)} &= \mathbf{E}_{x\sim X}[|2\alpha(x) - 1|\,(g_0(x) - f_0(x; w^{(t-1)}))\,(x \cdot w^{(t-1)})] \\ &\le \mathbf{E}_{x\sim X}[(g_0(x) - f_0(x; w^{(t-1)}))\,(x \cdot w^{(t-1)})]\,.\end{aligned}$$

We will show that for x it holds that

$$g_0(x) - f(x; w^{(t-1)})(x \cdot w^{(t-1)}) \le \frac{1}{e}\,.$$

Fix some x and let $s = w^{(t-1)} \cdot x$. Assume first that $g_0(x) = 1$. Then, we have

$$g_0(x) - f(x; w^{(t-1)})(x \cdot w^{(t-1)}) = \left(1 - \frac{1}{1+e^{-s}}\right) s = s\,\frac{e^{-s}}{1+e^{-s}} \le \frac{1}{e}\,,$$

where we used the fact that $s\,\frac{e^{-s}}{1+e^{-s}} \le 0$ for $s \le 0$ and $s\,\frac{e^{-s}}{1+e^{-s}} \le se^{-s} \le 1/e$ for $s \ge 0$ (using the elementary inequality $ze^{-z} \le 1/e$ for all $z \in \mathbf{R}$). When $g_0(x) = 0$ we similarly have that

$$g_0(x) - f(x; w^{(t-1)})(x \cdot w^{(t-1)}) = -\frac{s}{1+e^{-s}} \le \frac{1}{e}\,,$$

where we used the fact that when $s \ge 0$ it holds that $-\frac{s}{1+e^{-s}} \le 0$ and when $s \le 0$, $-\frac{s}{1+e^{-s}} \le -s/e^{-s} = -se^{s}$. For the final inequality, we used again the inequality $ze^{-z} \le 1/e$ for all $z \in \mathbf{R}$ (where we replaced z with $-s$).

Therefore, we obtain that $\mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] \cdot w^{(t-1)} \le 1/e$ and $\sum_{t=1}^{T} \mathbf{E}[u^{(t)} \mid \mathcal{F}^{(t-1)}] \cdot w^{(t-1)} \le T/e$. Using the decomposition of Equation (6), linearity of expectation, and the tower rule for conditional expectations, we conclude that $\mathbf{E}[\|w^{(T)}\|_2^2] \le (2/e + 1)T$. Using Markov's inequality we obtain that with probability at least 99% it holds that $\|w^{(T)}\|_2^2 = O(T)$ or equivalently $\|w^{(T)}\|_2 = O(\sqrt{T})$.

□

We can now finish the proof of Theorem 5.1. Assume, in order to reach a contradiction, that for all $t \le T$ it holds that $\mathbf{P}_{x\sim X}[\mathrm{err}(f(x; w^{(t)}), g(x))] > \epsilon$. Now picking T to be larger than a sufficiently large constant multiple of $1/(\epsilon^2\gamma^2\beta^2)$ and using Claim 1 and Claim 2 we obtain that, with probability at least 99%, it holds that $w^{(T)} \cdot w^*/\|w^{(T)}\|_2 \ge \Omega(\beta\gamma\epsilon\sqrt{T})$, which can be made to be larger than 1 by our choice of T. However, this is a contradiction as by Cauchy-Schwarz we have $w^{(T)} \cdot w^*/\|w^{(T)}\|_2 \le \|w^*\|_2 \le 1$. Therefore, with probability at least 99%, it must be that for some $t \le T$ it holds that $\mathbf{P}_{x\sim X}[\mathrm{err}(f(x; w^{(t)}), g(x))] \le \epsilon$.

□

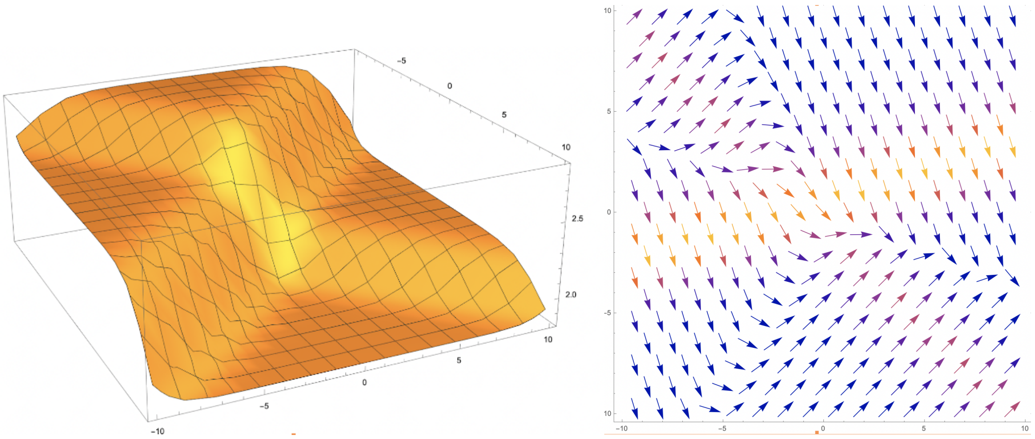

Figure 8: The landscape and gradient field of the population student label mixing loss for a simple 2 dimensional feature problem with a ground truth corresponding to a halfspace. We observe that the landscape is non-convex; however we can see that the corresponding gradient field "points towards the optimal direction" and therefore gradient descent converges to the global minimizer. A potential issue is the fact that the landscape contains regions where the gradients may almost vanish and this could lead to the gradient iteration of the student getting trapped there. To handle this issue, in Algorithm 3 we multiply the gradient of SLaM with an appropriate step-size.