SPARSIFYING LABEL SMOOTHING TO BETTER CALIBRATE CLASSIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Label smoothing is widely used to reduce neural network overconfidence. Existing variants, whether uniform or adaptive, differ in how much smoothing mass they assign and how they weight it, but all spread that mass densely across the label space, so semantically plausible alternatives and unrelated labels alike receive probability mass. We argue that effective label smoothing depends not only on the smoothing strength but also on the support over which smoothing mass is assigned: smoothing should be sparse and similarity-aware. We instantiate this principle with Sparse Similarity-Aware Label Smoothing (SSA-LS), a simple and practical method that uses latent-space class neighborhoods to redistribute probability mass only toward plausible alternatives. Across multiple datasets and architectures, SSA-LS consistently improves confidence calibration while preserving, and sometimes improving, accuracy. Beyond SSA-LS, similarity-aware sparsity acts as a general wrapper, improving the calibration of existing adaptive label smoothing methods without hurting accuracy. We further show that strong representations do not by themselves guarantee well-calibrated predictions, and through sparsity ablations and neighborhood corruption studies, that the gains arise specifically from structured, similarity-aware sparsity, establishing it as a critical ingredient for label smoothing.

1 INTRODUCTION

Misclassifying a truck as a cat is clearly wrong. So is calling it a car. But are these mistakes equally severe? This distinction matters not only for accuracy, but also for confidence: a model that assigns uncertainty to plausible alternatives is qualitatively different from one that spreads or concentrates probability mass without regard to label-space structure. Modern deep neural networks achieve high top-1 accuracy, but their predicted probabilities are often poorly aligned with empirical correctness (Guo et al., 2017). This miscalibration gap provides unreliable confidence estimates, and can contaminate downstream decision-making, which is an especially significant problem for risk-aware systems that depend on well-calibrated confidence scores. This has motivated extensive work on calibration metrics and procedures, such as reliability diagrams and expected calibration error (ECE), as well as post-hoc methods such as temperature scaling (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005; Naeini et al., 2015; Guo et al., 2017).

Beyond post-hoc correction, a range of training-time objectives target calibration directly, including focal loss (Mukhoti et al., 2020) and explicit calibration penalties (Kumar et al., 2018; Hebbalaguppe et al., 2022). Perhaps the most widely adopted training-time remedy is *label smoothing*, which replaces one-hot targets with a convex combination of the hard label and a prior distribution over classes, typically uniform (Szegedy et al., 2016). By discouraging sharply peaked predictive distributions, label smoothing acts as an effective regularizer to mitigate overconfident predictions (Müller et al., 2019; Pereyra et al., 2017). More broadly, training strategies that mix or soften labels—such as mixup—attribute part of their calibration benefit to the implicit label smoothing effect (Thulasidasan et al., 2019).

However, uniform label smoothing treats every incorrect class alike: in our opening example, it assigns the same probability to *car* as to *cat*. Semantic structure in the label space has been used to make mistakes less severe (Bertinetto et al., 2020), and two lines of work have refined label smoothing itself. *Adaptive* schemes replace the uniform prior with class- or instance-dependent distributions

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

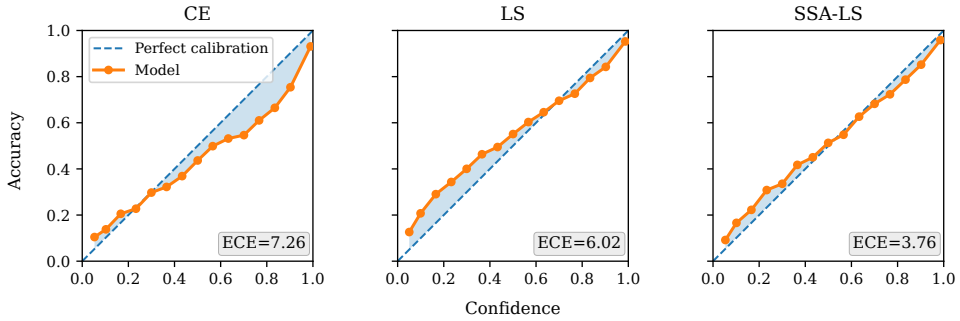


Figure 1: Reliability diagrams (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005) comparing using Cross Entropy (CE), Uniform Label Smoothing (LS) (Szegedy et al., 2016), and SSA-LS (ours) targets in an example setting (ResNet-18 on TinyImageNet). Dashed diagonal denotes perfect calibration; orange curve shows empirical accuracy within confidence bins. SSA-LS moves the reliability curve closer to the diagonal and substantially reduces ECE compared with CE and LS.

that are, for instance, learned or accumulated from the model's own predictions (Chhabra et al., 2024; Zhang et al., 2021; Maher & Kull, 2021; Liu & JaJa, 2021). *Margin-based* methods instead reinterpret label smoothing as a constraint on logit distances and relax it into a margin penalty (Liu et al., 2022), with class-adaptive (Liu et al., 2023) and conditional (Park et al., 2023) extensions. Both lines change how much smoothing is applied or how it is weighted, yet both still act on every non-target class. We ask a complementary question: must smoothing assign non-zero probability to every incorrect class at all? This motivates an appealing principle: label smoothing should smooth *toward plausible alternatives only*.

In this work, we identify **similarity-aware sparsity** as a design principle for label smoothing. Our central claim is that calibration depends not only on how much smoothing mass is assigned, but also on *where* it is assigned: the support of the smoothing distribution is a first-class design choice, and it should be restricted to semantically plausible alternatives. We study this principle through **Sparse Similarity-Aware Label Smoothing** (SSA-LS), its simplest instantiation: a drop-in target construction rule that assigns smoothing mass only to neighboring classes in a latent space. Our main contributions are as follows:

- **Neighborhood size primarily controls calibration.** Varying only the size of the neighborhood that receives smoothing mass changes calibration substantially while leaving accuracy largely unchanged, making the support a distinct design choice that governs calibration.
- **Sparsity must be structured, not merely sparse.** Random or corrupted neighborhoods degrade calibration, whereas neighborhoods derived from different latent spaces, including pretrained CLIP and DINO representations and an in-domain β-VAE, yield similar gains. At the same time, strong representations alone do not guarantee calibrated predictions; calibration improves when smoothing explicitly exploits their semantic structure.
- **The principle generalizes beyond SSA-LS.** Applied as a wrapper to existing adaptive methods such as LS++ (Chhabra et al., 2024) and OLS (Zhang et al., 2021), similarity-aware sparsity improves their calibration without degrading accuracy, and combined with margin-based logit regularization, it consistently improves and stabilizes their calibration.
- **Even its simplest instantiation is highly effective.** SSA-LS requires only a one-time precomputation per dataset and incurs no training-time overhead, yet achieves the best calibration among label smoothing methods across architectures and datasets.

2 Preliminaries

Let $\mathcal{X}$ and $\mathcal{Y} = \{1, 2, \ldots, C\}$ denote the input and output spaces respectively, where C is the number of classes present. We train neural network $h_\theta : \mathcal{X} \to \mathbb{R}^C$ that outputs logits $h_\theta(\boldsymbol{x})$ for input $\boldsymbol{x} \in \mathcal{X}$ to predict the true class label $y \in \mathcal{Y}$. Here, the model then predicts class $\hat{y} = \arg\max_c p_\theta(c|\boldsymbol{x})$ with associated confidence $\hat{p} = \max_c p_\theta(c|\boldsymbol{x})$ where $p_\theta(c|\boldsymbol{x}) = \mathrm{softmax}_c(h_\theta(\boldsymbol{x}))$. A classifier is

perfectly calibrated if

$$\mathbb{P}(\hat{y} = y|\hat{p} = p) = p, \quad \forall p \in [0, 1] \tag{1}$$

Essentially, it means that among all predictions made with confidence p, the model is correct $100p\%$ times.

Label Smoothing. Label smoothing replaces the one-hot target distribution with a softened target. Let the smoothing strength ϵ denote the probability mass redistributed among incorrect classes, which may depend on the sample $\boldsymbol{x}$, true label y, and timestep t for adaptive or online label smoothing methods. In the general case, we have target distribution

$$q(c \mid \boldsymbol{x}, y, t) = (1 - \epsilon(\boldsymbol{x}, y, t))\mathbb{1}(c = y) + \epsilon(\boldsymbol{x}, y, t) \cdot s(c \mid \boldsymbol{x}, y, t) \tag{2}$$

where $s(c \mid \boldsymbol{x}, y, t)$ is a valid probability distribution such that $s(y|\boldsymbol{x}, y, t) = 0$.

Uniform label smoothing (LS) (Szegedy et al., 2016) is the standard and simple method where a fixed ϵ is redistributed uniformly over all incorrect classes. Specifically, we have

$$q_\epsilon^{\text{LS}}(c \mid y) = \begin{cases} 1 - \epsilon, & \text{if } c = y, \\ \dfrac{\epsilon}{C - 1}, & \text{otherwise.} \end{cases} \tag{3}$$

Metrics. Expected calibration error, a useful scalar summary statistic of calibration (Guo et al., 2017), is the expectation of difference between the model's confidence and accuracy

$$\mathop{\mathbb{E}}_{\hat{p}}\left[|\mathbb{P}\left(\hat{y} = y|\hat{p} = p\right) - p|\right] \tag{4}$$

To approximately compute ECE, we partition predictions into M equal intervals (Naeini et al., 2015; Guo et al., 2017). Let interval $I_m = \left(\frac{m-1}{M}, \frac{m}{M}\right]$ and B_m be the corresponding set of indices of samples based on prediction confidences. Then, we define

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{y}_i = y_i) \tag{5}$$

and

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \tag{6}$$

Finally, we may approximate ECE with the interval bins as the weighted average

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)| \tag{7}$$

where n is the total number of samples and, in practice, M is set to 15. Observe that for a perfectly calibrated model we have $\text{acc}(B_m) = \text{conf}(B_m)$ for all m i.e. $\text{ECE} = 0$. This is the primary metric we employ to compare performance of label smoothing methods ahead.

We also consider classification accuracy, negative log likelihood (NLL), and maximum calibration error (Naeini et al., 2015) defined as

$$\text{MCE} = \max_m |\text{acc}(B_m) - \text{conf}(B_m)| \tag{8}$$

as additional metrics. ECE and MCE are reported as percentages (%), as is classification accuracy. For visual aid, we rely on reliability diagrams (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005).

3 Similarity-aware sparsity for label smoothing

Overview. We propose a general recipe to incorporate structured sparsity, based on a learned latent space, into training target distributions. A fixed latent encoder is used to extract features for all train samples, from which class-wise centroids and inter-class distances are computed. For each class, these distances are used to identify a fixed number of plausible misclassifications, and the smoothing mass is redistributed among them in a similarity-aware manner. This procedure yields a sparse, similarity-aware, and smooth target distribution for training. We further introduce SSA-LS, a simple instantiation of this framework that enables controlled analysis of similarity-aware sparsity.

Construction. Let f denote the chosen encoder, and $\boldsymbol{z}_i \in \mathbb{R}^d$ such that $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$ denote the latent representation of sample $\boldsymbol{x}_i$ where d is the corresponding latent space's dimensionality. Let $D_c = \{\boldsymbol{x}_i : y_i = c\}$. Then, for class D_c, we have the latent class centroid

$$\boldsymbol{\mu}_c = \frac{1}{|D_c|} \sum_{\boldsymbol{x}_i \in D_c} f(\boldsymbol{x}_i) \tag{9}$$

The class-wise distances are computed based on these centroids, where the metric $d^{(e)}$ is chosen to match the inductive bias of the representation space. In practice, we use cosine distance for CLIP (Radford et al., 2021) and DINO (Oquab et al., 2023), and euclidean distance for β-VAE (Higgins et al., 2017; Kingma & Welling, 2014). The framework itself is agnostic to the choice of latent representation.

Now, we introduce the sparsity factor k. Let $\mathcal{N}_k(c)$ denote the set containing the k closest neighbors of class c, excluding itself, under the chosen distance metric $d^{(e)}$. Let $q(c \mid \boldsymbol{x}, y, t)$ denote the target distribution produced by a base label smoothing method as defined in Equation 2. We then construct a sparse similarity-aware variant $q^{\text{SSA}}(c \mid \boldsymbol{x}, y, t)$, by redistributing the non-target probability mass among classes in $\mathcal{N}_k(y)$:

$$q^{\text{SSA}}(c \mid \boldsymbol{x}, y, t) = (1 - \epsilon(\boldsymbol{x}, y, t))\mathbb{1}\{c = y\} + \epsilon(\boldsymbol{x}, y, t) \cdot s^{\text{SSA}}(c \mid \boldsymbol{x}, y, t) \tag{10}$$

where the sparse similarity-aware redistribution is

$$s^{\text{SSA}}(c \mid \boldsymbol{x}, y, t) = \frac{s(c \mid \boldsymbol{x}, y, t) \cdot \mathbb{1}\{c \in \mathcal{N}_k(y)\}}{\sum_{j \in \mathcal{N}_k(y)} s(j \mid \boldsymbol{x}, y, t)} \tag{11}$$

To study the effect of similarity-aware sparsity, we introduce SSA-LS, a simple label smoothing method based on the precomputed class-wise distances. We define

$$q_\epsilon^{\text{SSA-LS}}(c \mid y) = \begin{cases} 1 - \epsilon, & \text{if } c = y, \\ \epsilon \cdot \dfrac{\exp\left(-d^{(e)}(\boldsymbol{\mu}_c, \boldsymbol{\mu}_y)/T\right)}{\sum\limits_{j \in \mathcal{N}_k(y)} \exp\left(-d^{(e)}(\boldsymbol{\mu}_j, \boldsymbol{\mu}_y)/T\right)}, & \text{if } c \in \mathcal{N}_k(y), \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

where temperature parameter $T > 0$ is set to $T = 1.2$ by default; we analyze its sensitivity in Appendix I. Since our goal is to study similarity-aware sparsity as a principle rather than to engineer the strongest possible method, we deliberately keep SSA-LS minimal: neighborhoods are fixed per class, computed once before training, and independent of the model. With no learned, adaptive, or model-dependent components, any change in calibration can be attributed to the similarity-aware sparsity of the smoothing support itself.

4 Experiments & Analysis

We evaluate on CIFAR-100 (Krizhevsky & Hinton, 2009), TinyImageNet (Le & Yang, 2015), and ImageNet-1k (Deng et al., 2009). For CIFAR-100, we train ResNet-18, ResNet-34 (He et al., 2016), DenseNet-121 (Huang et al., 2017), and WideResNet-28-10 (Zagoruyko & Komodakis, 2016); for TinyImageNet, we train ResNet-18 and DenseNet-121. For ImageNet-1k, we train lightweight 2-layer MLP probes defined as

$$\text{Linear}(d \to 1024) \to \text{ReLU} \to \text{Dropout}(p = 0.2) \to \text{Linear}(1024 \to C) \tag{13}$$

where d is the latent feature size and C is the number of classes, on frozen features using the CLIP ViT-B/32 (Radford et al., 2021) image encoder.

The latent spaces used to construct our similarity-aware targets are CLIP ViT-B/32 (OpenAI pretrained) (Radford et al., 2021), DINOv2 ViT-B/14 (Oquab et al., 2023), and β-VAE (Kingma & Welling, 2014; Higgins et al., 2017). CLIP and DINOv2 are used as fixed pretrained encoders, while the classification models and β-VAE are trained from scratch. We use CLIP as the default and compare these latent spaces in section 4.3.

Hyperparameters are selected on a held-out validation split, and all models are trained under the same optimization protocol within each dataset–architecture setting. All results report mean ($\pm$ standard deviation) over 5 seeds. Full experimental setup and training details are presented in Appendix A.

Baselines. We compare SSA-LS against training-time methods that modify the loss or targets to improve calibration, spanning target-construction methods, which reshape the soft-label distribution (LS (Szegedy et al., 2016), OLS (Zhang et al., 2021), ILS (Maher & Kull, 2021), LS++ (Chhabra et al., 2024)), and logit-regularization methods, which penalize the logit margin directly (MbLS (Liu et al., 2022), CALS (Liu et al., 2023), ACLS (Park et al., 2023)). We additionally compare against standard cross-entropy (CE).

For a fair comparison, all methods parameterized by smoothing strength use a fixed ϵ, tuned once on LS per setting and shared across methods; we ablate ϵ in Appendix H. The margin-based methods (MbLS, CALS, ACLS), whose primary hyperparameter is the margin size m, receive a validation-searched m; SSA-LS uses a sparsity factor k selected per dataset-architecture. All remaining hyperparameters follow each method's official defaults; complete details in Appendix A.

4.1 SSA-LS CONSISTENTLY WINS CALIBRATION

We first demonstrate that our sparse similarity-aware redistribution results in the best-calibrated models compared to all evaluated baselines in varying settings.

Table 1 reports results on CIFAR-100 for ResNet-18 and DenseNet-121, a residual and a densely-connected architecture. Even though the architectures differ significantly, SSA-LS outperforms all evaluated baselines and achieves the lowest ECE in both settings. We observe similar results for ResNet-34 and WideResNet-28-10 on CIFAR-100 (Appendix B), where SSA-LS again results in lowest ECE. The consistency of these gains is important: calibration is known to depend on model-design choices such as depth and width (Guo et al., 2017; Minderer et al., 2021), yet SSA-LS improves calibration across every evaluated backbone.

Table 1: Calibration and accuracy comparison across various settings. SSA-LS consistently achieves the lowest ECE, while MCE and NLL are also improved in most cases, and maintains accuracy. Full metrics (including MCE, NLL and $\pm$std for each) and additional architectures in Appendix B.

	CIFAR-100 ResNet-18		CIFAR-100 DenseNet-121		TinyImageNet ResNet-18		ImageNet-1k MLP	
Method	ECE ↓	Acc ↑	ECE ↓	Acc ↑	ECE ↓	Acc ↑	ECE ↓	Acc ↑
CE	4.49	78.45	10.26	67.99	7.26	66.22	6.76	74.26
LS	5.84	79.13	5.89	67.64	6.02	65.91	2.25	74.73
OLS	3.43	79.32	17.04	66.93	6.70	66.64	13.54	74.97
ILS	4.31	78.64	9.64	67.81	7.13	66.21	6.80	74.24
LS++	6.38	79.32	9.27	67.57	7.71	66.2	1.46	74.52
MbLS	4.71	79.35	11.99	67.75	7.03	66.25	4.37	74.72
CALS	5.44	78.63	13.79	66.82	6.04	66.27	19.66	74.53
ACLS	3.86	79.43	12.40	67.98	5.87	65.97	2.91	74.66
SSA-LS	**2.34**	78.62	**4.58**	67.68	**3.76**	65.71	**1.14**	74.34

Importantly, SSA-LS does not buy calibration by sacrificing predictive performance. Regularization methods often trade accuracy for calibration (Zhang et al., 2022), however, SSA-LS maintains accuracy comparable to the strongest baselines and often improves it. Thus, the gains are not explained by stronger regularization, but by a better use of the same smoothing budget: concentrating probability mass on semantically plausible alternatives rather than spreading it across all non-target classes. It also reduces MCE and NLL in most settings (see Appendix B).

We observe the same qualitative trend beyond CIFAR-100. On TinyImageNet and ImageNet-1k, SSA-LS again improves calibration while maintaining competitive accuracy. Together, these results show that sparse similarity-aware redistribution yields consistently better-calibrated models across label-space scales and architecture families.

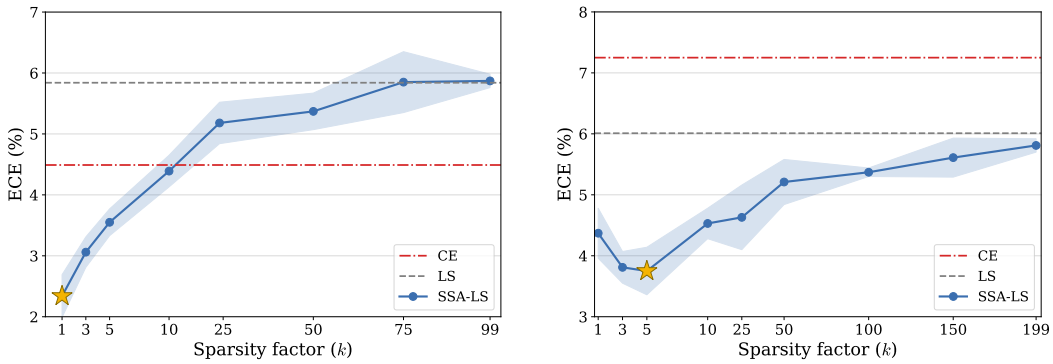
Our ImageNet-1k frozen-feature experiment from Table 1 further shows that strong pretrained representations alone do not guarantee good calibration, and that representation-aware target construction remains essential even in this regime. We further discuss this in Appendix G.

4.2 Effects of sparsity on calibration and accuracy

Since our sparse label smoothing protocol redistributes probability mass among similar classes, the sparsity factor k—the number of classes receiving this mass—is a central design choice. We show that sparsity is not a secondary hyperparameter, but a defining mechanism governing the effectiveness of label smoothing. In particular, we hypothesize that restricting redistribution to a small, semantically meaningful neighborhood is essential for achieving strong calibration and show strong evidence towards it.

Redistributing smoothing mass across many classes weakens the locality assumption underlying similarity-aware smoothing methods (Zhang et al., 2021; Maher & Kull, 2021; Liu & JaJa, 2021; Chhabra et al., 2024). In large-scale settings, this effect is amplified: if mass is spread more broadly, the target distribution approaches uniform label smoothing, since variations in allocation become negligible relative to the dominant true class mass $1 - \epsilon$. As a result, the benefits of incorporating semantic similarity are diminished. Within SSA-LS, we observe this behavior concretely: increasing k consistently degrades calibration, with large-k regimes effectively collapsing to uniform smoothing.

Conversely, overly restrictive sparsity underutilizes available neighborhood information and increases sensitivity to noise in the latent structure. These observations highlight a fundamental trade-off: effective smoothing must preserve semantic locality while leveraging sufficient neighborhood information. We therefore hypothesize that *moderate sparsity*—dependent on dataset scale and underlying class structure—yields optimal calibration by preserving tight, semantically meaningful neighborhoods while maximizing the exploitable information within them. We validate this hypothesis through controlled ablations over k (Figure 2), using ResNet-18 as a controlled setting. Varying the neighborhood size k changes the support of the smoothing distribution, allowing us to quantify how sparsity affects calibration while monitoring its effect on classification accuracy.



(a) CIFAR-100 (b) TinyImageNet

Figure 2: Impact of sparsity on calibration (ECE ↓) across datasets. Sparsity directly controls calibration: SSA-LS achieves optimal performance at small to moderate k, while larger k degrades performance. Star denotes configuration with lowest ECE; error bands indicate ± 1 std across seeds.

For CIFAR-100, we observe that the ECE increases monotonically with k, with the lowest ECE achieved at $k = 1$. In contrast, TinyImageNet exhibits a non-monotonic relationship, where ECE decreases to a minimum at moderate sparsity ($k = 3$–5) before increasing again as sparsity is further relaxed.

These results highlight the dataset-dependent nature of optimal sparsity. For CIFAR-100, which contains only 100 classes, redistributing the smoothing mass to strictly the closest neighbor appears sufficient to capture all meaningful semantic relationships, while additional neighbors introduce noise and dilute locality. On the other hand, TinyImageNet benefits from slightly larger neighborhoods ($k = 3$–5). Notably, ImageNet-1k, despite its substantially larger label space, exhibits the same optimal range as TinyImageNet rather than requiring further growth in k (Appendix C), suggesting that the relevant notion of scale is not the raw number of classes but the density of semantically similar classes within the label space.

This reinforces that sparsity should be treated as a primary design choice, governed by dataset structure, rather than a secondary hyperparameter. While the exact optimal k exhibits architecture dependence, the overall trend remains consistent: calibration improves at small-to-moderate sparsity and degrades as k increases.

The accuracy–calibration trade-off (Zhang et al., 2022) remains a potential concern here. However, we observe that classification accuracy remains largely stable across sparsity levels. Pooled accuracy over all k values remains stable at $78.99 \pm 0.30\%$ and $65.65 \pm 0.27\%$ for CIFAR-100 and TinyImageNet, respectively. Mean top-1 accuracy varies narrowly from 78.62% to 79.19% for CIFAR-100 and 65.38% to 65.84% for TinyImageNet. That is, although minor variations are present, they are small in magnitude and do not exhibit a consistent trade-off with calibration (see Appendix C for complete results). Sparsity thus primarily affects calibration rather than predictive performance, making k a practical lever for reducing ECE with minimal impact on accuracy.

4.3 The structure of sparsity

Similarity-aware vs. naive sparsity. The question arises: what if we discard the learned latent space and impose random sparsity? Here, we investigate two claims: (i) sparsity must be similarity-aware, and (ii) SSA-LS is robust to imperfect representations.

We conduct controlled corruption experiments on the similarity structure used to construct SSA-LS targets. We artificially corrupt the target construction: for each class, we replace a fraction p of its top-k neighbors with randomly selected classes (not in the original top-k), progressively degrading the semantic quality of the neighborhood while keeping sparsity fixed. Setting $k = 5$, we train using these corrupted targets (Table 2).

Table 2: Effect of neighborhood corruption on calibration (ResNet-18, CIFAR-100). Corruption replaces a fraction of top-k neighbors with random classes. Random sparsity performs substantially worse than similarity-aware sparsity, and SSA-LS degrades towards CE under extreme corruption.

Method	Corruption	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
CE	–	4.49 ± 0.16	78.45 ± 0.23	0.87 ± 0.01	14.58 ± 5.41
SSA-LS	0.0	$\mathbf{3.49 \pm 0.27}$	78.97 ± 0.21	$\mathbf{0.83 \pm 0.01}$	15.95 ± 2.72
SSA-LS	0.2	3.79 ± 0.40	78.95 ± 0.36	0.85 ± 0.01	$\mathbf{14.42 \pm 1.58}$
SSA-LS	0.4	3.85 ± 0.31	78.98 ± 0.21	0.87 ± 0.01	17.14 ± 2.04
SSA-LS	0.6	3.98 ± 0.30	78.76 ± 0.19	0.91 ± 0.00	17.00 ± 2.12
SSA-LS	0.8	4.25 ± 0.49	78.87 ± 0.32	0.94 ± 0.01	18.68 ± 1.31
SSA-LS	1.0	4.29 ± 0.16	78.72 ± 0.20	0.97 ± 0.01	18.49 ± 1.72

SSA-LS exhibits graceful degradation: as the corruption level increases ($p = 0.0 \rightarrow 1.0$), calibration performance (ECE, MCE, NLL) degrade steadily rather than collapsing, while accuracy remains largely stable. Crucially, SSA-LS outperforms CE across a wide range of corruption levels and only approaches CE under extreme corruption, where semantic structure is entirely destroyed. This supports our claim that similarity-aware sparsity—rather than sparsity alone—is what improves calibration and that SSA-LS is robust to poor representations.

Robustness to latent space choice. Our formulation defines semantic neighborhoods using distances between class centroids in a latent space. This raises a natural question: are the gains tied to a particular representation, or does SSA-LS only require a reasonably meaningful notion of class similarity? To test whether the gains are tied to a particular representation, we construct neighborhoods (k=5) from three substantially different latent spaces: pretrained CLIP and DINO representations, and a β-VAE trained directly on the dataset. All three reduce ECE relative to CE by similar margins while maintaining accuracy, on both ResNet-18 (4.49 to 3.55–3.69) and WideResNet-28-10 (4.64 to 3.15–3.73) on CIFAR-100 (see Appendix E for full results). CLIP performs best overall, but all three representations yield consistent gains: SSA-LS does not depend on a privileged encoder, only on a meaningful class-neighborhood structure.

Table 3: Effect of similarity-aware sparsification on existing label smoothing methods. We construct sparse variants (SSA) of LS++ (Chhabra et al., 2024), OLS (Zhang et al., 2021), and ILS (Maher & Kull, 2021) that preserve each method's target-generation mechanism but restrict the smoothing mass to the k nearest neighbors (Equation 10; k=5). Sparsification improves the calibration of LS++ and OLS on both architectures without reducing accuracy, and also improves the accuracy of OLS.

(a) ResNet-34

Method	ECE ↓	Accuracy ↑
LS++	5.76 ± 0.23	79.52 ± 0.19
SSA-LS++	5.09 ± 0.27	79.67 ± 0.50
OLS	9.78 ± 0.23	79.01 ± 0.16
SSA-OLS	7.93 ± 0.98	79.54 ± 0.28
ILS	6.52 ± 0.57	79.16 ± 0.48
SSA-ILS	6.65 ± 0.21	79.12 ± 0.23

(b) DenseNet-121

Method	ECE ↓	Accuracy ↑
LS++	9.27 ± 0.20	67.57 ± 0.24
SSA-LS++	8.65 ± 0.15	67.64 ± 0.43
OLS	17.04 ± 0.35	66.93 ± 0.11
SSA-OLS	12.39 ± 0.44	67.60 ± 0.41
ILS	9.64 ± 0.39	67.81 ± 0.33
SSA-ILS	9.61 ± 0.44	68.08 ± 0.41

4.4 Sparsifying existing label smoothing methods

We directly evaluate similarity-aware sparsity as a first-class principle in label smoothing and assess whether it can act as a general wrapper for existing label smoothing methods. We construct sparse variants of LS++ (Chhabra et al., 2024), OLS (Zhang et al., 2021), and ILS (Maher & Kull, 2021) by preserving each method's original target-generation mechanism and applying SSA only to the final target distribution: the target-class probability is unchanged, while the non-target mass is redistributed over the top-k semantic neighbors as in Equation 10.

As shown in Table 3, sparsification consistently improves the calibration of LS++ (ECE $5.76 \rightarrow 5.09$ on ResNet-34 and $9.27 \rightarrow 8.65$ on DenseNet-121) and OLS ($9.78 \rightarrow 7.93$ and $17.04 \rightarrow 12.39$) without reducing accuracy, and additionally improves the accuracy of OLS by 0.5–0.7% on both architectures. ILS, in contrast, is essentially unaffected (ECE 6.52 vs. 6.65 on ResNet-34 and 9.64 vs. 9.61 on DenseNet-121, with comparable accuracy).

To understand why, we measure the targets SSA-ILS constructs during training (ResNet-34, ϵ=0.02, k=5). Sparsification does reshape ILS's non-target distribution: only $\sim 60\%$ of the teacher's non-target mass lies within $\mathcal{N}_k(y)$, and the sparse and dense distributions differ by a total variation (TV) distance of 0.34–0.43. However, because ILS scales the smoothing strength by the teacher's confidence, its per-instance strength averages only $\epsilon_i \approx 0.0012$ at a nominal ϵ=0.02, so its targets are $\sim 99.9\%$ one-hot and change by less than 0.001 in TV. ILS is thus limited not by support density but by how little mass it redistributes, consistent with its unchanged performance across ϵ (Appendix H).

Taken together, these results establish similarity-aware sparsity as a general principle rather than a property of SSA-LS alone. The three methods generate their smoothing distributions in fundamentally different ways (learned, accumulated from the model's own predictions, or derived from a teacher), yet sparsification did not hurt any of them and improved calibration for those that redistribute substantial mass. This reflects a dimension of label smoothing that existing methods leave unconstrained: they determine how much mass to redistribute and how to weight it, while similarity-aware sparsity determines where it may go. The same holds beyond label smoothing: combining SSA-LS with margin-based logit regularization or post-hoc temperature scaling (Guo et al., 2017) likewise improves calibration over these methods alone (Appendix D).

5 Related Work

Calibration. The calibration of modern neural networks has been extensively studied (Niculescu-Mizil & Caruana, 2005; Guo et al., 2017; Minderer et al., 2021). Guo et al. (2017) show that contemporary deep models are often systematically overconfident and demonstrate that temperature scaling, a simple post-hoc rescaling of logits, can substantially improve calibration. Post-hoc methods leave training unchanged and can be applied on top of any training-time remedy, including ours.

Training-time calibration. Several training objectives target calibration directly, including focal loss (Mukhoti et al., 2020), entropy-based confidence penalties (Pereyra et al., 2017), and differentiable calibration objectives such as MMCE (Kumar et al., 2018) and MDCA (Hebbalaguppe et al., 2022); data augmentation such as mixup (Thulasidasan et al., 2019) also improves calibration. Most closely related to label smoothing, margin-based methods reinterpret it as a constraint on logit distances, relaxed into a margin penalty by MbLS (Liu et al., 2022) and made class-adaptive and conditional by CALS (Liu et al., 2023) and ACLS (Park et al., 2023), respectively.

Label smoothing and its variants. Label smoothing (Szegedy et al., 2016) mixes one-hot targets with a uniform distribution and is an effective calibration regularizer (Müller et al., 2019). Adaptive variants derive the smoothing distribution from training signals: the model's accumulated predictions (Zhang et al., 2021), a teacher (Maher & Kull, 2021), or class-wise targets learned jointly with the network (Chhabra et al., 2024). Others exploit label-space structure explicitly, deriving soft targets from latent spaces and word embeddings (Liu & JaJa, 2021) or from a class hierarchy (Bertinetto et al., 2020).

Despite these advances, existing approaches primarily focus on how smoothing mass should be distributed, not on *where* it should be distributed. To the best of our knowledge, prior work does not explicitly investigate sparsity as a design principle in label smoothing. Our core contribution is this principle rather than a particular method: SSA-LS is its simplest instantiation, and because similarity-aware sparsity constrains only the support of the smoothing distribution, it is orthogonal to how existing methods weight smoothing mass and can be applied on top of them (subsection 4.4, Appendix D).

6 Limitations

Choice of neighborhood size. The optimal sparsity factor k varies across architectures and datasets and therefore requires a small validation search. The search is inexpensive, as effective values consistently fall in a narrow range (k roughly in $[1, C/5]$), and four candidate values suffice in our experiments. Selected values for a variety of architectures and datasets are reported in subsection A.1.

A deliberately minimal instantiation. SSA-LS is intentionally simple, using fixed class-wise neighborhoods so that gains can be attributed to similarity-aware sparsity itself. This leaves substantial room for more sophisticated methods built on the same principle, for example selecting the neighborhood size adaptively, jointly with the smoothing strength, or per instance.

7 Conclusion

We presented similarity-aware sparsity as a design principle for label smoothing: calibration is governed not only by how much smoothing mass is assigned, but also by the support on which it is placed. SSA-LS, the simplest instantiation of this principle, consistently improves calibration across architectures and datasets while preserving accuracy. Through sparsity ablations, corrupted and random neighborhoods, and latent-space comparisons, we show that these gains arise from structured semantic support rather than sparsity alone. The principle also extends beyond SSA-LS: applied as a wrapper, it improves existing adaptive label smoothing methods, and it remains compatible with margin-based logit regularization and post-hoc temperature scaling. We further show that strong representations do not by themselves guarantee calibrated predictions: representations can reveal class structure, but the training objective must still exploit it. Overall, similarity-aware sparsity reframes label smoothing as a question of *where* uncertainty should be assigned, not merely how much: effective label smoothing should not simply soften labels, but place the softened mass sparsely on the right alternatives.

Since SSA-LS was deliberately kept minimal to isolate the effect of the principle, a natural next step is to design more expressive methods built on it. A theoretical characterization of similarity-aware sparsity, including how the optimal support size depends on the smoothing strength and the label-space structure, is another promising direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments, implement methods, and lightly interpret results. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, support qualitative and thematic data analysis, or clean and reformat dataset. Generating synthetic data sets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and assistance with translation are not applicable to this work. Additionally, we used generative AI tools to create or edit code, summarize or analyze existing literature, discover research topics or identify gaps, and edit the research paper to improve readability.

We have reviewed all AI-assisted work. The authors took feedback on the research methodology to ensure fair comparisons with baselines and reviewed it to ensure the suggestions reflect standard experimental protocol. The authors used generative AI to implement the method, and verified and tested the code for correctness. AI was used to implement plotting code for the figures, and the numbers in the figures were crosschecked and confirmed by the authors. AI was used to lightly polish the prose, and it was ensured the core material of the text remained unchanged. Finally, AI was used after a manual literature search to identify any gaps in the list of related works. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work is compliant with the ICLR Code of Ethics. Improved confidence calibration can have positive societal impact by making model predictions more reliable and interpretable, especially in settings where confidence estimates inform downstream decisions. However, better calibration on standard benchmarks does not by itself guarantee safe or fair deployment. Calibration may degrade under distribution shift, class imbalance, or subgroup-specific variation, and calibrated confidence should not be treated as a substitute for broader safety, robustness, fairness, or human-oversight mechanisms. Our work is evaluated on standard image classification benchmarks and is not intended for direct deployment in high-stakes settings without further validation.

REPRODUCIBILITY STATEMENT

We describe the proposed principle and method in a self-contained manner in section 3. The full experimental setup, including training hyperparameters, data processing, the validation search for methods' hyperparameters, and the selected values for each architecture and dataset, is provided in Appendix A. All experiments use publicly available image classification benchmarks, standard architectures, and open-source latent models, and all results are reported as means and standard deviations over five random seeds. To facilitate reproduction, we release the code for our full pipeline, including neighborhood precomputation and training, in the supplementary material.

REFERENCES

Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A. Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Sachin Chhabra, Hemanth Venkateswara, and Baoxin Li. Label smoothing++: Enhanced label regularization for training neural networks. In *British Machine Vision Conference (BMVC)*, 2024.

Siran Dai, Qianqian Xu, Peisong Wen, Yang Liu, and Qingming Huang. Exploring structural degradation in dense representations for self-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 16715–16764, 2025.

Morris H. DeGroot and Stephen E. Fienberg. The comparison and evaluation of forecasters. In *Proceedings of the 1982 I.O.S. Annual Conference on Practical Bayesian Statistics*, pp. 12–22, 1983.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, pp. 2–3, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16081–16090, June 2022.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks . In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

Weiran Huang, Mingyang Yi, Xuyang Zhao, and Zihao Jiang. Towards the generalization of contrastive self-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations*, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2805–2814, 2018.

Y. Le and X. Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. The devil is in the margin: Margin-based label smoothing for network calibration. In *CVPR*, 2022.

Bingyuan Liu, Jérôme Rony, Adrian Galdran, Jose Dolz, and Ismail Ben Ayed. Class adaptive network calibration. In *CVPR*, 2023.

Chihuang Liu and Joseph JaJa. Class-similarity based label smoothing for confidence calibration. In *Artificial Neural Networks and Machine Learning – ICANN 2021*, pp. 190–201, 2021.

Mohamed Maher and Meelis Kull. Instance-based label smoothing for better calibrated classification networks. *20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 746–753, 2021.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15682–15694, 2021.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, pp. 15288–15299, 2020.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2901, 2015.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22th International Conference on Machine Learning (ICML 2005)*, pp. 625–632, 2005.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

Hyekang Park, Jongyoun Noh, Youngmin Oh, Donghyeon Baek, and Bumsub Ham. ACLS: Adaptive and conditional label smoothing for network calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *International Conference on Learning Representations (ICLR), Workshop Track Proceedings*, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, {Alexander C.} Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, December 2015.

Florian Seligmann, Philipp Becker, Michael Volpp, and Gerhard Neumann. Beyond deep ensembles: A large-scale evaluation of bayesian deep learning under distribution shift. In *Advances in Neural Information Processing Systems*, pp. 29372–29405, 2023.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. *On mixup training: improved calibration and predictive uncertainty for deep neural networks*. 2019.

Jarne Verhaeghe, Thomas De Corte, Christopher M. Sauer, Tom Hendriks, Olivier W.M. Thijssens, Femke Ongenae, Paul Elbers, Jan De Waele, and Sofie Van Hoecke. Generalizable calibrated machine learning models for real-time atrial fibrillation risk prediction in icu patients. *International Journal of Medical Informatics*, 175:105086, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng. Delving deep into label smoothing. *IEEE Transactions on Image Processing*, pp. 5984–5996, 2021.

Hanlin Zhang, Xuechen Li, Prithviraj Sen, Salim Roukos, and Tatsunori Hashimoto. A closer look at the calibration of differentially private learners. In *NeurIPS 2022 Workshop on Algorithmic Fairness through the Lens of Causality and Privacy*, 2022.

A Experimental Setup

Datasets. We evaluate on three standard image classification benchmarks: CIFAR-100 (Krizhevsky & Hinton, 2009), TinyImageNet (Le & Yang, 2015), and ImageNet-1k (Deng et al., 2009). These cover varying dataset scales—the number of classes is 100 in CIFAR-100, 200 in TinyImageNet, and 1000 in ImageNet-1k. For data augmentation, we apply random cropping and horizontal flipping (Krizhevsky et al., 2012) to the training sets. Apart from dataset normalization, no other preprocessing is performed.

Hyperparameter selection protocol. To select a particular hyperparameter, we hold out 10% of train data for a validation split so no hyperparameter decision uses the test set. We train a single seed for each value in a small grid and pick the value with the lowest validation ECE. Then, we discard the validation models and train fresh models over 5 seeds on the full training set and evaluate on test data, which constitute the headline results.

Three hyperparameters are tuned, each over a comparable four-point grid:

- Smoothing factor/strength (ϵ). This is the nominal smoothing budget and controls the amount of probability mass redistributed across non-target class; it applies to LS, ILS, LS++, SSA-LS. To keep the budget fixed and directly comparably across methods, rather than tuning it separately for each, we select a single value from $\epsilon \in [0.01, 0.02, 0.05, 0.1]$ and apply it throughout. We tuned on plain LS, the original and most direct baseline parameterized by ϵ, and subsequently fixed $\epsilon = 0.02$ for CIFAR-100 and TinyImagenet, and $\epsilon = 0.05$ for ImageNet-1k. For a fair comparison, we additionally perform a complete ablation on ϵ in Appendix H, which confirms that our conclusions are not an artifact of this particular value.
- Sparsity factor (k). Controls the number of non-target classes that receive non-zero probability mass and is relevant primarily for the SSA-LS models. For each dataset-architecture pair, we search for k in a small grid depending on dataset scale ([1, 3, 5, 10] for CIFAR100, [1, 5, 10, 15] for TinyImageNet, [10, 25, 100, 200] for ImageNet); the selected values are reported below in subsection A.1. For our experiment involving the sparsification of other methods (subsection 4.4), where k is not the object of study, we fix $k = 5$ a priori instead of tuning.
- Margin (m). Applies to the logit-margin methods MbLS, CALS, and ACLS. (Liu et al., 2022; Park et al., 2023; Liu et al., 2023). We perform a validation search over m for each of these methods individually. Following these methods' original protocols, which fix a single margin per dataset or task type rather than per architecture, we search $m \in [6, 8, 10, 12]$ once and reuse the selected value across architectures. For datasets spanning multiple architectures we run this search on ResNet-18; for ImageNet-1k, which uses a single architecture, the search is performed directly on that architecture and no cross-architecture transfer is involved. Consequently, in order of MbLS, CALS, ACLS, we fix margins $\{10, 8, 8\}$ for CIFAR-100, $\{10, 12, 12\}$ for TinyImageNet, and $\{12, 12, 12\}$ for ImageNet-1k.

The remaining SSA-LS hyperparameter, the softmax temperature T, is fixed a priori at $T = 1.2$ for all experiments rather than tuned; the accompanying sensitivity analysis (Appendix I) shows performance remains stable across a wide range of T, so it is not a sensitive knob. For all other methods we use the hyperparameters recommended by their original authors.

Baselines. We compare against training-time methods that modify the loss or targets to improve calibration, which are the most direct baselines for SSA-LS. These broadly fall into two families: target-construction methods, which reshape the soft-label distribution, and logit-regularization methods, which penalize the logit margin directly. We adapt the official released code for the evaluated baselines. Explicitly, the baselines are as follows:

1. Cross Entropy (CE) (standard one-hot targets).
2. Uniform Label Smoothing (LS) (Szegedy et al., 2016).
3. Online Label Smoothing[1] (OLS) (Zhang et al., 2021).
4. Instance-based Label Smoothing[2] (ILS) (Maher & Kull, 2021).
5. LS++[3] (Chhabra et al., 2024).
6. Margin-based Label Smoothing[4] (MbLS) (Liu et al., 2022).
7. Class Adaptive Label Smoothing[5] (CALS) (Liu et al., 2023).
8. Adaptive and Conditional Label Smoothing[6] (ACLS) (Park et al., 2023).

A.1 Training Details

Classification models. All convolutional networks are trained from scratch for 200 epochs using SGD with Nesterov momentum 0.9 and weight decay 5×10^{-4}. We use cosine learning-rate decay with an initial learning rate of 0.1 and batch size 128. For ResNet-18 and ResNet-34, we replace the ImageNet stem with a 3×3 convolution with stride 1 and remove the initial max-pooling layer.

The MLP probe (Equation 13) for frozen-feature experiments is trained with AdamW using a learning rate of 1×10^{-3}. All other evaluation settings are unchanged and follow the hyperparameters above.

In-domain β-VAE. We train a β-VAE on CIFAR-100 and TinyImageNet as an additional latent space, where β is annealed linearly from 0 to 4 over the first 20% epochs. For the in-domain latent representation, we train a convolutional β-VAE separately on each dataset and use the latent mean μ to compute class centroids. The complete list of hyperparameters used is listed in Table 4.

Table 4: Training details for the in-domain β-VAE representations used to construct class neighborhoods.

Setting	CIFAR-100	TinyImageNet
Input resolution	32×32	64×64
Encoder/decoder depth	3 stages	3 stages
Normalization	GroupNorm	GroupNorm
Latent dimension	128	32
Batch size	256	256
Optimizer	Adam	Adam
Learning rate	2×10^{-4}	2×10^{-4}
Weight decay	2×10^{-5}	1×10^{-5}
Training epochs	300	300

Smoothing methods. For SSA-LS on CIFAR-100, the sparsity factor is set to $k = 10$ for DenseNet-121, $k = 1$ for ResNet-18, $k = 5$ for ResNet-34, and $k = 1$ for WideResNet-28-10. For TinyImageNet, the sparsity factor is set to $k = 15$ for DenseNet-121 and $k = 5$ for ResNet-18. For ImageNet-1k, the sparsity factor is set to $k = 10$. Class neighborhoods are computed once per dataset and reused across seeds and architectures.

B Additional results on CIFAR-100 & TinyImageNet

We provide complete metrics for Table 1 and additional backbones on CIFAR-100 and TinyImageNet (Table 5, Table 6) to support the main findings. These experiments support the main conclusion

[1] https://github.com/ankandrew/online-label-smoothing-pt
[2] https://github.com/mmaher22/Instance-based-smoothing
[3] https://github.com/s-chh/LSPP
[4] https://github.com/by-liu/MbLS
[5] https://github.com/by-liu/CALS
[6] https://github.com/cvlab-yonsei/ACLS

that the calibration benefits of SSA-LS are not specific to a single model or dataset. In all settings, SSA-LS reduces ECE relative to standard label smoothing and other soft-target baselines, while preserving or even improving top-1 accuracy.

C Extended sparsity ablation results

We report the full numerical results corresponding to the sparsity analysis from subsection 4.2 in Table 7. Further, we extend the sparsity factor k ablation to ImageNet-1k (Figure 3, Table 8).

Consistent with our findings on CIFAR-100 and TinyImageNet, accuracy on ImageNet-1k remains stable across sparsity levels, ranging from 73.91% to 74.71% (a spread of 0.80%), while ECE varies substantially with k (Figure 3). This confirms that sparsity primarily governs calibration rather than accuracy, even at ImageNet-1k scale, and that this decoupling is not an artifact of smaller-scale benchmarks.

Notably, the optimal sparsity on ImageNet-1k remains small ($k = 3$–5), matching TinyImageNet rather than growing with the fivefold increase in class count. This is consistent with ImageNet-1k's WordNet-derived taxonomy, which is known to contain large clusters of fine-grained, semantically adjacent classes (Deng et al., 2009; Russakovsky et al., 2015), a property TinyImageNet inherits by construction, as its classes are drawn from the same hierarchy (Le & Yang, 2015). The CIFAR-100 label space, by contrast, is organized into a shallow, uniform hierarchy of 20 superclasses with exactly five leaf classes each (Krizhevsky & Hinton, 2009) yielding smaller neighborhoods that plausibly explain why $k = 1$ already suffices there. Taken together, these results support treating optimal sparsity as being governed by the semantic density of the label space rather than its raw cardinality.

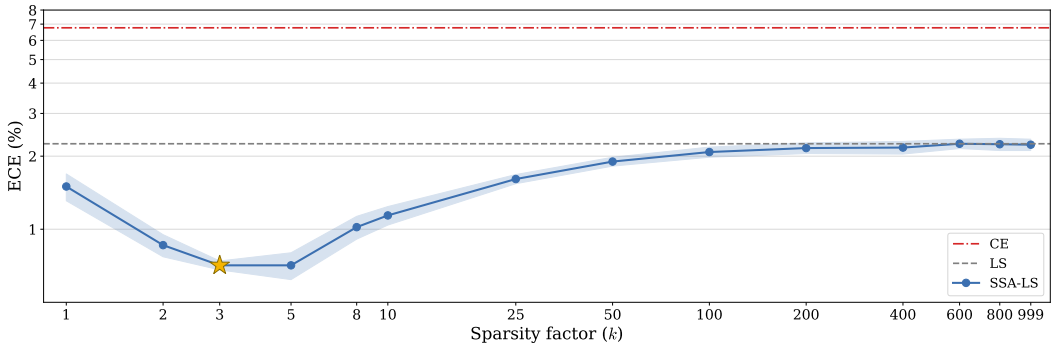


Figure 3: Impact of sparsity on calibration (ECE ↓) on ImageNet-1k. The results further validate the trend, over a much larger dataset scale, observed in subsection 4.2: a tight, semantically meaningful neighborhood is essential for better calibration and larger k degrades performance. Horizontal lines denote CE and LS baselines; star denotes configuration with lowest ECE. Axes are log-scale; error bands indicate ± 1 standard deviation across seeds.

D Complementarity with existing calibration methods

As a target construction method, SSA-LS is naturally combined with methods that operate in a different plane such as post-hoc calibration and calibration-focused augmentation, and even orthogonal training-time methods such as the logit-regularization methods. A natural question is whether the gains from SSA-LS persist under such combinations, or whether they are subsumed by what these methods already correct. Let $\mathcal{M}$ be a calibration method whose training objective contains a cross-entropy term. We say SSA-LS is *compatible* with $\mathcal{M}$ if replacing that term with the SSA-LS objective improves calibration over $\mathcal{M}$ alone, and *complementary* to $\mathcal{M}$ if the combination SSA-LS + $\mathcal{M}$ improves calibration over both SSA-LS and $\mathcal{M}$ alone. We test this against temperature scaling (TS) (Guo et al., 2017), perhaps the most popular post-hoc calibration technique, and against the logit-regularization baselines MbLS, CALS, and ACLS.

Table 5: Calibration and accuracy comparison across various architectures on CIFAR-100.

(a) ResNet-18

Method	ECE ↓	Acc ↑	NLL ↓	MCE ↓
CE	4.49 ± 0.16	78.45 ± 0.23	0.87 ± 0.01	14.58 ± 5.41
LS	5.84 ± 0.31	79.13 ± 0.10	0.93 ± 0.00	20.60 ± 1.56
OLS	3.43 ± 0.17	79.32 ± 0.20	**0.81 ± 0.00**	12.88 ± 4.74
ILS	4.31 ± 0.18	78.64 ± 0.30	0.87 ± 0.01	15.01 ± 2.83
LS++	6.38 ± 0.19	79.32 ± 0.12	0.94 ± 0.00	22.74 ± 0.91
MbLS	4.71 ± 2.97	79.35 ± 0.28	0.89 ± 0.05	14.75 ± 4.07
CALS	5.44 ± 3.72	78.63 ± 0.30	0.98 ± 0.10	16.45 ± 3.13
ACLS	3.86 ± 0.59	**79.43 ± 0.23**	0.87 ± 0.02	13.52 ± 1.84
SSA-LS	**2.34 ± 0.35**	78.62 ± 0.21	0.84 ± 0.01	**12.82 ± 1.55**

(b) DenseNet-121

Method	ECE ↓	Acc ↑	NLL ↓	MCE ↓
CE	10.26 ± 0.33	**67.99 ± 0.41**	1.32 ± 0.01	21.29 ± 0.94
LS	5.89 ± 0.53	67.64 ± 0.16	1.35 ± 0.01	15.66 ± 1.92
OLS	17.04 ± 0.35	66.93 ± 0.11	1.48 ± 0.02	38.48 ± 1.48
ILS	9.64 ± 0.39	67.81 ± 0.33	1.33 ± 0.01	24.84 ± 10.51
LS++	9.27 ± 0.20	67.57 ± 0.24	1.40 ± 0.01	23.10 ± 1.86
MbLS	11.99 ± 0.43	67.75 ± 0.20	1.40 ± 0.02	27.95 ± 2.21
CALS	13.79 ± 0.21	66.82 ± 0.33	1.54 ± 0.02	35.96 ± 1.86
ACLS	12.40 ± 0.59	67.98 ± 0.23	1.41 ± 0.03	29.55 ± 1.58
SSA-LS	**4.58 ± 0.51**	67.68 ± 0.29	**1.28 ± 0.01**	**11.32 ± 1.88**

(c) WideResNet-28-10

Method	ECE ↓	Acc ↑	NLL ↓	MCE ↓
CE	4.64 ± 0.17	81.04 ± 0.18	0.78 ± 0.01	17.46 ± 6.51
LS	3.97 ± 0.48	81.07 ± 0.18	0.82 ± 0.01	15.86 ± 6.68
OLS	8.74 ± 0.19	80.71 ± 0.23	0.84 ± 0.00	39.53 ± 30.51
ILS	4.52 ± 0.34	**81.29 ± 0.23**	0.78 ± 0.01	13.16 ± 2.75
LS++	4.97 ± 0.20	80.73 ± 0.22	0.84 ± 0.00	18.75 ± 2.14
MBLS	7.68 ± 0.21	80.73 ± 0.21	0.86 ± 0.01	24.94 ± 1.17
CALS	4.78 ± 0.32	80.52 ± 0.29	0.92 ± 0.01	21.64 ± 0.98
ACLS	6.90 ± 0.10	80.49 ± 0.26	0.87 ± 0.01	25.50 ± 2.77
SSA-LS	**3.20 ± 0.14**	80.97 ± 0.31	**0.76 ± 0.01**	**12.87 ± 1.40**

(d) ResNet-34

Method	ECE ↓	Acc ↑	NLL ↓	MCE ↓
CE	7.13 ± 0.44	79.30 ± 0.41	0.86 ± 0.01	20.32 ± 2.68
LS	5.26 ± 0.51	79.46 ± 0.49	0.89 ± 0.02	**17.16 ± 1.49**
OLS	9.78 ± 0.23	79.01 ± 0.16	0.89 ± 0.00	43.26 ± 28.52
ILS	6.52 ± 0.57	79.16 ± 0.48	0.86 ± 0.01	19.22 ± 2.56
LS++	5.76 ± 0.23	**79.52 ± 0.19**	0.89 ± 0.00	20.27 ± 4.41
MBLS	9.04 ± 0.14	79.00 ± 0.17	0.93 ± 0.01	27.61 ± 0.96
CALS	6.20 ± 0.30	78.69 ± 0.39	1.00 ± 0.02	25.73 ± 1.97
ACLS	8.55 ± 0.58	79.11 ± 0.27	0.94 ± 0.02	27.47 ± 2.30
SSA-LS	**4.83 ± 0.80**	79.50 ± 0.63	**0.81 ± 0.02**	17.25 ± 2.68

Table 6: Calibration and accuracy comparison for ResNet-18 and DenseNet-121 on TinyImageNet.

(a) ResNet-18

Method	ECE ↓	Acc ↑	NLL ↓	MCE ↓
CE	7.26 ± 0.20	66.22 ± 0.22	1.52 ± 0.01	17.24 ± 1.45
LS	6.02 ± 0.56	65.91 ± 0.62	1.59 ± 0.01	13.07 ± 1.60
OLS	6.70 ± 0.15	**66.64 ± 0.32**	1.52 ± 0.00	16.37 ± 1.05
ILS	7.13 ± 0.09	66.21 ± 0.16	1.51 ± 0.01	18.71 ± 2.17
LS++	7.71 ± 0.29	66.20 ± 0.15	1.61 ± 0.01	18.32 ± 2.10
MBLS	7.03 ± 0.15	66.25 ± 0.40	1.60 ± 0.01	16.63 ± 1.30
CALS	6.04 ± 0.60	66.27 ± 0.23	1.57 ± 0.01	13.53 ± 1.30
ACLS	5.87 ± 0.40	65.97 ± 0.11	1.54 ± 0.01	11.77 ± 0.79
SSA-LS	**3.76 ± 0.40**	65.71 ± 0.14	**1.49 ± 0.01**	**8.85 ± 1.58**

(b) DenseNet-121

Method	ECE ↓	Acc ↑	NLL ↓	MCE ↓
CE	14.49 ± 0.27	59.55 ± 0.16	1.90 ± 0.02	29.79 ± 1.40
LS	11.57 ± 0.19	59.48 ± 0.33	1.84 ± 0.01	15.27 ± 0.58
OLS	12.96 ± 0.50	**60.41 ± 0.38**	1.84 ± 0.02	25.72 ± 2.19
ILS	14.08 ± 0.35	59.99 ± 0.29	1.88 ± 0.00	26.48 ± 1.10
LS++	9.36 ± 0.25	60.08 ± 0.15	1.84 ± 0.01	18.97 ± 1.34
MBLS	7.53 ± 0.40	59.95 ± 0.40	1.81 ± 0.01	15.79 ± 0.95
CALS	11.48 ± 0.24	60.23 ± 0.18	1.84 ± 0.01	22.89 ± 1.13
ACLS	11.39 ± 0.34	60.10 ± 0.25	1.81 ± 0.01	22.52 ± 1.90
SSA-LS	**7.39 ± 0.27**	59.77 ± 0.33	**1.78 ± 0.02**	**14.92 ± 1.62**

D.1 Post-hoc: SSA-LS with temperature scaling

Temperature scaling divides all logits by a single learned scalar, uniformly compressing or expanding the confidence distribution. This correction is effective only when miscalibration is globally consistent —that is, when confidence is systematically shifted in one direction across all levels. To diagnose the *direction* of miscalibration we consider the signed variant of Equation 7, sECE (Seligmann et al., 2023; Verhaeghe et al., 2023), which removes the absolute value:

$$\text{sECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} (\text{acc}(B_m) - \text{conf}(B_m)) \tag{14}$$

A negative sECE indicates overconfidence and a positive value indicates underconfidence; more importantly, its magnitude reflects how much global bias the model exhibits.

Following the temperature scaling protocol (Guo et al., 2017), we hold out 10% of the training data and train two architectures on CIFAR-100 using CE, LS, and SSA-LS targets. After training, the held-out split is used to fit the temperature for TS (Table 9).

On ResNet-18, CE and SSA-LS both have small global bias (sECE: CE $= -1.91$, SSA-LS $= 2.03$), so TS has little to correct in either case: CE's and SSA-LS's ECE is essentially unchanged ($4.90 \rightarrow 4.89$ and $2.84 \rightarrow 2.86$, respectively). LS, however, is strongly underconfident (sECE $= +6.17$), as seen in the reliability diagram (Figure 4), giving TS a large unidirectional bias to remove, and ECE drops from 6.37 to 4.21. However, even after this correction, LS + TS still trails SSA-LS without any post-hoc adjustment.

On ResNet-34, all three methods exhibit substantial overconfidence (sECE: CE $= -7.28$, LS $= -4.60$, SSA-LS $= -4.24$), and TS is correspondingly effective across the board: CE drops from 7.52 to 4.63, LS from 5.41 to 4.81, and SSA-LS from 4.52 to 3.18. Even here, SSA-LS + TS achieves the lowest ECE.

Table 7: Effect of sparsity on calibration and performance. Varying the neighborhood size k shows that calibration is strongly affected by the support of the smoothing distribution, while accuracy remains largely stable. Small semantic neighborhoods yield the best ECE and MCE on both CIFAR-100 and TinyImageNet, indicating that smoothing mass should be concentrated on a limited set of plausible alternatives rather than spread broadly across many classes.

(a) CIFAR-100

Method	Sparsity (k)	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
CE	–	4.49 ± 0.16	78.45 ± 0.23	0.87 ± 0.01	14.58 ± 5.41
LS	–	5.84 ± 0.31	**79.13 ± 0.10**	0.93 ± 0.01	20.60 ± 1.56
SSA-LS	1	**2.34 ± 0.35**	78.62 ± 0.21	0.84 ± 0.01	**12.82 ± 1.55**
SSA-LS	3	3.06 ± 0.25	78.86 ± 0.24	**0.83 ± 0.00**	14.85 ± 1.79
SSA-LS	5	3.55 ± 0.22	79.02 ± 0.23	0.83 ± 0.01	15.15 ± 1.28
SSA-LS	10	4.39 ± 0.26	79.02 ± 0.35	0.85 ± 0.01	16.19 ± 0.83
SSA-LS	25	5.18 ± 0.34	79.16 ± 0.23	0.88 ± 0.01	19.43 ± 2.69
SSA-LS	50	5.37 ± 0.30	79.04 ± 0.22	0.91 ± 0.01	18.48 ± 0.81
SSA-LS	75	5.85 ± 0.50	79.19 ± 0.34	0.92 ± 0.01	20.52 ± 2.68
SSA-LS	99	5.87 ± 0.11	79.00 ± 0.27	0.93 ± 0.01	20.67 ± 2.01

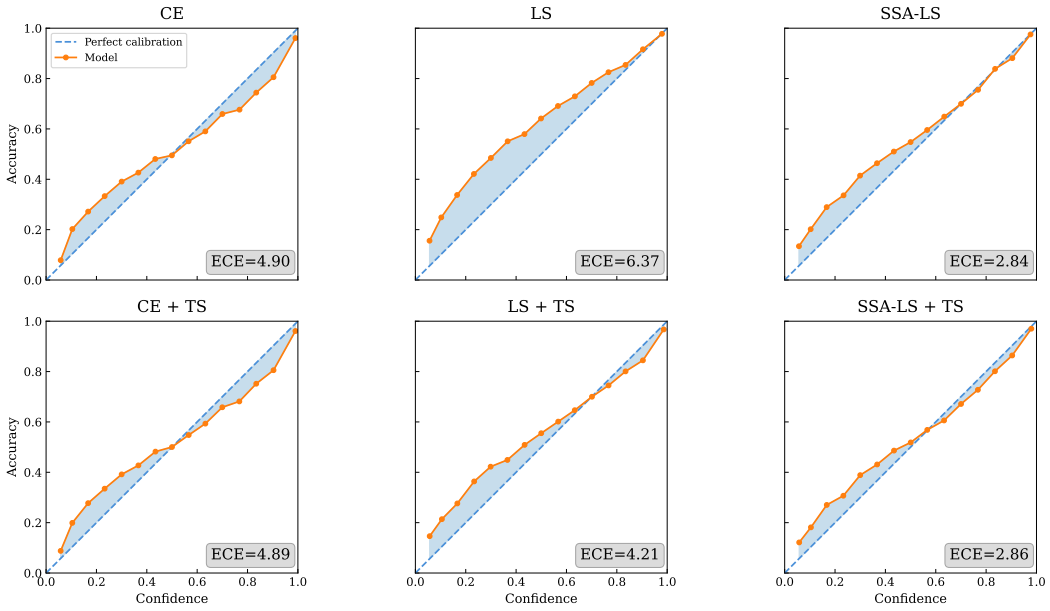
(b) TinyImageNet

Method	Sparsity (k)	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
CE	–	7.25 ± 0.21	**66.22 ± 0.22**	1.52 ± 0.01	17.25 ± 1.39
LS	–	6.01 ± 0.56	65.91 ± 0.62	1.59 ± 0.01	13.12 ± 1.67
SSA-LS	1	4.37 ± 0.41	65.55 ± 0.26	1.50 ± 0.01	10.40 ± 1.04
SSA-LS	3	3.81 ± 0.26	65.38 ± 0.25	1.50 ± 0.01	9.14 ± 0.71
SSA-LS	5	**3.75 ± 0.39**	65.71 ± 0.14	**1.49 ± 0.01**	**8.82 ± 1.65**
SSA-LS	10	4.53 ± 0.25	65.84 ± 0.28	**1.49 ± 0.01**	9.21 ± 0.91
SSA-LS	25	4.63 ± 0.53	65.55 ± 0.19	1.52 ± 0.01	10.29 ± 1.90
SSA-LS	50	5.21 ± 0.37	65.79 ± 0.26	1.53 ± 0.01	11.72 ± 1.94
SSA-LS	100	5.37 ± 0.07	65.66 ± 0.40	1.56 ± 0.01	12.47 ± 0.68
SSA-LS	150	5.61 ± 0.32	65.72 ± 0.28	1.57 ± 0.01	12.34 ± 1.06
SSA-LS	199	5.81 ± 0.11	65.65 ± 0.23	1.59 ± 0.01	12.28 ± 1.52

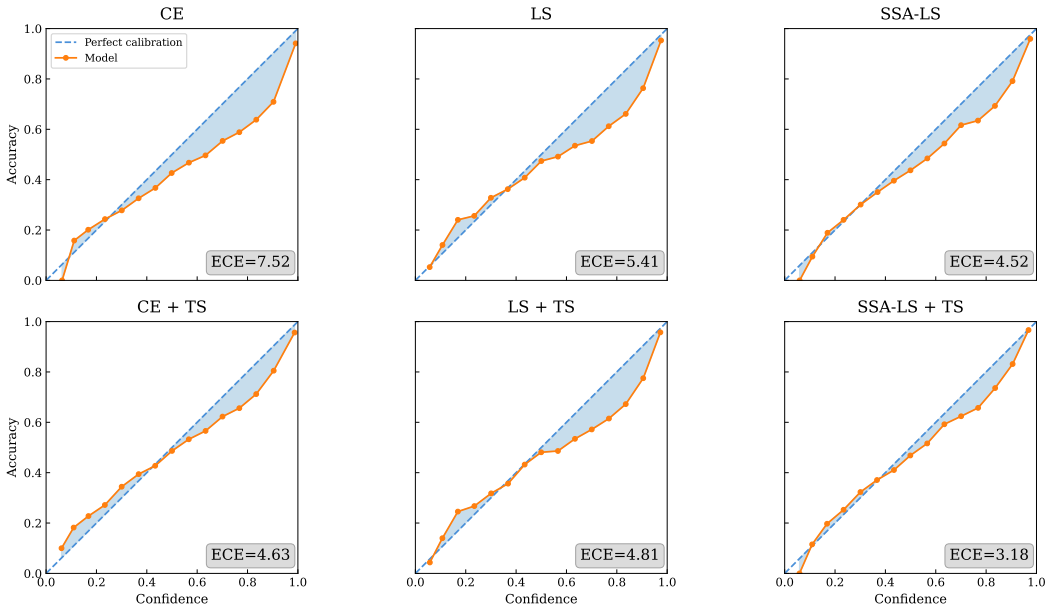
Table 8: Effect of sparsity on calibration and performance ImageNet-1k.

Method	Sparsity (k)	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
CE	–	6.76 ± 0.11	74.26 ± 0.09	1.01 ± 0.00	12.86 ± 0.85
LS	–	2.25 ± 0.08	74.73 ± 0.09	0.98 ± 0.00	5.65 ± 0.43
SSA-LS	1	1.50 ± 0.19	73.91 ± 0.20	1.00 ± 0.00	5.83 ± 0.78
SSA-LS	2	0.86 ± 0.09	74.09 ± 0.07	0.99 ± 0.00	**5.03 ± 0.96**
SSA-LS	3	**0.71 ± 0.03**	74.10 ± 0.08	0.98 ± 0.00	41.24 ± 48.25
SSA-LS	5	**0.71 ± 0.09**	74.24 ± 0.10	0.98 ± 0.00	5.20 ± 1.48
SSA-LS	8	1.02 ± 0.11	74.31 ± 0.08	0.98 ± 0.00	13.53 ± 17.27
SSA-LS	10	1.14 ± 0.10	74.34 ± 0.10	0.98 ± 0.00	6.14 ± 0.24
SSA-LS	25	1.61 ± 0.07	74.41 ± 0.07	0.98 ± 0.00	18.40 ± 23.57
SSA-LS	50	1.90 ± 0.08	74.51 ± 0.09	0.98 ± 0.00	6.49 ± 0.97
SSA-LS	100	2.08 ± 0.10	74.60 ± 0.09	0.98 ± 0.00	6.23 ± 1.26
SSA-LS	200	2.16 ± 0.11	74.61 ± 0.09	**0.97 ± 0.00**	7.61 ± 3.61
SSA-LS	400	2.17 ± 0.13	74.61 ± 0.12	0.98 ± 0.00	12.88 ± 7.00
SSA-LS	600	2.25 ± 0.10	**74.71 ± 0.09**	0.98 ± 0.00	6.28 ± 1.63
SSA-LS	800	2.24 ± 0.13	**74.71 ± 0.11**	0.98 ± 0.00	5.62 ± 0.71
SSA-LS	999	2.23 ± 0.12	74.69 ± 0.13	0.98 ± 0.00	5.82 ± 0.34

The pattern across both settings is consistent. TS is a global scalar correction whose benefit scales with |sECE|: it helps most when miscalibration is large and unidirectional, and has little effect when the global signed bias is already small. Since temperature scaling optimizes held-out likelihood, it generally does not harm calibration, while SSA-LS consistently improves it at training time. The two act on different stages of the pipeline and their gains compose, confirming that SSA-LS is complementary to temperature scaling.



(a) ResNet-18



(b) ResNet-34

Figure 4: Reliability diagrams before (top row) and after (bottom row) temperature scaling on CIFAR-100. Temperature scaling corrects global confidence shifts (e.g. CE on ResNet-34), but provides little additional benefit when there is no systematic confidence shift to correct. (e.g. SSA-LS on ResNet-18). When both are combined, SSA-LS + TS achieves the lowest ECE across both architectures.

Table 9: Complementarity with temperature scaling. SSA-LS consistently improves calibration regardless of architecture, whereas TS benefits vary. When both are effective, the two compose for the lowest ECE.

(a) ResNet-18

Method	ECE ↓	Accuracy ↑
CE	4.90 ± 0.12	77.70 ± 0.25
CE + TS	4.89 ± 0.11	77.70 ± 0.25
LS	6.37 ± 0.55	**78.07 ± 0.36**
LS + TS	4.21 ± 0.29	**78.07 ± 0.36**
SSA-LS	**2.83 ± 0.41**	77.91 ± 0.27
SSA-LS + TS	2.86 ± 0.29	77.91 ± 0.27

(b) ResNet-34

Method	ECE ↓	Accuracy ↑
CE	7.52 ± 0.33	78.19 ± 0.49
CE + TS	4.63 ± 0.70	78.19 ± 0.49
LS	5.41 ± 0.98	78.42 ± 0.48
LS + TS	4.81 ± 0.37	78.42 ± 0.48
SSA-LS	4.52 ± 1.01	**78.65 ± 0.25**
SSA-LS + TS	**3.18 ± 0.37**	**78.65 ± 0.25**

Table 10: Compatibility with margin-based logit-regularization methods. Replacing these methods' cross-entropy term with the SSA-LS objective lowers and stabilizes ECE while maintaining the accuracy boost induced by these methods. However, no combination surpasses SSA-LS alone on calibration.

Method	ECE ↓	Acc ↑	NLL ↓	MCE ↓
CE	4.49 ± 0.16	78.45 ± 0.23	0.87 ± 0.01	14.58 ± 5.41
SSA-LS	**2.34 ± 0.35**	78.62 ± 0.21	**0.84 ± 0.01**	**12.82 ± 1.55**
MbLS	4.71 ± 2.97	79.35 ± 0.28	0.89 ± 0.05	14.75 ± 4.07
MbLS + SSA-LS	3.09 ± 0.19	79.19 ± 0.19	**0.84 ± 0.00**	12.96 ± 1.06
CALS	5.44 ± 3.72	78.63 ± 0.30	0.98 ± 0.10	16.45 ± 3.13
CALS + SSA-LS	4.72 ± 0.29	78.40 ± 0.30	0.94 ± 0.01	13.61 ± 2.07
ACLS	3.86 ± 0.59	**79.43 ± 0.23**	0.87 ± 0.02	13.52 ± 1.84
ACLS + SSA-LS	3.35 ± 0.25	79.23 ± 0.35	0.86 ± 0.00	15.71 ± 2.64

D.2 Training-time: SSA-LS with logit-margin regularization

A complementary line of calibration methods regularizes the logits directly rather than the targets: MbLS (Liu et al., 2022), CALS (Liu et al., 2023), and ACLS (Park et al., 2023) add margin-based penalties on logit distances to a standard cross-entropy objective. Since SSA-LS only changes the targets of that cross-entropy term, the two approaches can be combined directly by replacing each method's one-hot cross-entropy with the SSA-LS objective while keeping its penalty unchanged. We evaluate all combinations with ResNet-18 on CIFAR-100, using each individual method's standalone hyperparameters as reported in Appendix A; the hyperparameters were not re-tuned for the combined objective.

Table 10 shows that SSA-LS is compatible with all three margin-based methods. Replacing their cross-entropy term with the SSA-LS objective lowers ECE in every case (MbLS $4.71 \to 3.09$, CALS $5.44 \to 4.72$, ACLS $3.86 \to 3.35$), mirroring its effect on plain cross-entropy ($4.49 \to 2.34$), while NLL is maintained or improved and top-1 accuracy changes are within one standard deviation. SSA-LS also markedly stabilizes these methods: the across-seed standard deviation of ECE drops by up to an order of magnitude (MbLS $2.97 \to 0.19$, CALS $3.72 \to 0.29$), indicating that similarity-aware targets make margin-based training far less sensitive to initialization. MCE improves for MbLS and CALS, but not for ACLS ($13.52 \to 15.71$).

The combinations are, however, not complementary: none surpasses SSA-LS alone in calibration, whose ECE of 2.34 remains the lowest. Instead, the combinations occupy a different point on the accuracy–calibration trade-off (Zhang et al., 2022): they retain most of the margin methods' accuracy advantage over SSA-LS (e.g., MbLS + SSA-LS reaches 79.19% top-1 versus 78.62% for SSA-LS, with equal NLL) while recovering a substantial part of SSA-LS's calibration gains. Re-tuning the

margin hyperparameters for the combined objective may shift this trade-off further, which we leave to future work.

E Robustness to latent space choice

We provide explicit numerical results for the second half of subsection 4.3. Table 11 shows that SSA-LS is robust to the choice of latent space. Across two architectures, CLIP, DINO, and β-VAE all reduce ECE relative to CE while maintaining comparable accuracy. The measurements are also largely similar across latent spaces: while CLIP performs best overall, DINO and β-VAE remain close.

Table 11: Robustness to latent space choice. Setting $k = 5$, we construct SSA neighborhoods using CLIP, DINO, and β-VAE latent spaces. Across ResNet-18 and WideResNet-28-10 on CIFAR-100, all latent spaces improve ECE over CE while maintaining comparable accuracy, indicating that SSA-LS is robust to the representation used to define class similarity.

(a) ResNet-18

Latent Space	ECE ↓	Accuracy ↑
CE	4.49 ± 0.16	78.45 ± 0.23
CLIP	3.55 ± 0.22	79.02 ± 0.23
DINO	3.69 ± 0.37	78.96 ± 0.31
β-VAE	3.65 ± 0.21	78.87 ± 0.10

(b) WideResNet-28-10

Latent Space	ECE ↓	Accuracy ↑
CE	4.64 ± 0.17	81.04 ± 0.18
CLIP	3.15 ± 0.09	81.08 ± 0.24
DINO	3.55 ± 0.25	81.10 ± 0.10
β-VAE	3.73 ± 0.25	80.96 ± 0.06

These results suggest that SSA-LS does not depend on a single privileged encoder. Pretrained representations such as CLIP and DINO provide strong off-the-shelf neighborhoods, but an in-domain β-VAE also yields consistent calibration gains. Thus, the key ingredient is not the specific latent model, but the presence of a useful class-neighborhood structure for sparsely redistributing smoothing.

F Class- vs. instance-level neighborhoods

For SSA-LS, we deliberately opted for a class-wise scheme to isolate the effect of similarity-aware sparsity under a stable similarity structure, since instance-wise variability would confound this analysis. We now extend our analysis to the instance-level variant, termed SIA-LS. For a sample $\boldsymbol{x}$, we build instance-wise neighborhood $\mathcal{N}_{k,\text{inst}}(\boldsymbol{x})$ which includes the k-nearest classes based on distances $d^{(e)}(f(\boldsymbol{x}), \boldsymbol{\mu}_c)$ where f is the latent encoder and $\boldsymbol{\mu}_c$ is the latent centroid of class c. The redistribution of non-target mass is identical to SSA-LS, except that classes in the instance-wise neighborhood receive non-zero probability mass instead of classes in the instance's class-wise neighborhood.

First, we quantify how much neighborhoods of the two different granularities differ. Let n denote the dataset size. We consider the following two metrics:

$$\text{mean overlap} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\mathcal{N}_{\text{k,inst}}(\boldsymbol{x}_i) \cap \mathcal{N}_{k,\text{class}}(y_i)|}{k} \tag{15}$$

which measures how much an instance's neighborhood typically matches its class prototype, and

$$\text{mean normalized union size} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{C-1}\left|\bigcup_{i:\, y_i=c}\mathcal{N}_{k,\text{inst}}(\boldsymbol{x}_i)\right| \tag{16}$$

which measures what proportion of the label space a class's instances typically span over. Fixing $k = 5$, we observe that instance-level neighborhoods differ substantially from class-level ones: typically less than half of an instance's neighbors intersect with its class's neighborhood (mean overlap is 0.519 for CIFAR-100, 0.444 for TinyImageNet, and 0.352 for ImageNet-1k) and a class's instances collectively typically span a significantly large part of the the label space (mean normalized

union size is 0.639 for CIFAR-100, 0.529 for TinyImageNet and 0.211 for ImageNet-1k). Now, we investigate whether these differing neighborhood constructions lead to a difference in performance. We train ResNet-18 models on CIFAR-100 and TinyImageNet, and MLPs on ImageNet-1k.

Table 12: Effect of similarity granularity on sparse label smoothing. We compare class-wise (SSA-LS) and instance-wise (SIA-LS) variants that differ solely in whether neighborhoods are constructed at the class or instance level. Neither granularity consistently dominates: differences in ECE, accuracy, NLL, and MCE are small and inconsistent, indicating that instance-wise awareness yields no systematic gain over class-wise.

Dataset	Method	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
CIFAR-100	SIA-LS	4.05 ± 0.36	79.31 ± 0.28	0.81 ± 0.01	15.44 ± 1.05
	SSA-LS	3.55 ± 0.22	79.02 ± 0.23	0.83 ± 0.01	15.15 ± 1.28
TinyImageNet	SIA-LS	4.08 ± 0.35	66.23 ± 0.14	1.45 ± 0.01	7.96 ± 1.18
	SSA-LS	3.75 ± 0.39	65.71 ± 0.14	1.49 ± 0.01	8.82 ± 1.65
ImageNet-1k	SIA-LS	0.94 ± 0.09	74.39 ± 0.10	0.96 ± 0.00	5.40 ± 1.06
	SSA-LS	0.71 ± 0.09	74.24 ± 0.10	0.98 ± 0.00	5.20 ± 1.48

As shown in Table 12, SIA-LS and SSA-LS perform comparably across various settings, with no obvious general trend in calibration or accuracy across the datasets, even though the neighborhoods used to construct targets differ significantly. The finer instance-level granularity thus offers no systematic benefit over the class-level variant. This directly contrasts with our corruption experiments in subsection 4.3, where degrading the neighborhood structure produces systematic performance drops. Together these support our central claim: the operative variable is the presence of structured sparse support, not how it is exactly defined.

G Strong representations do not induce calibration

Employing a learned latent space to impose sparsity in target construction implicitly assumes that the encoder captures meaningful class structure. Prior work (Huang et al., 2023; Dai et al., 2025) has shown that even simple nearest-neighbor classifiers trained on well-structured representations achieve strong accuracy, raising a critical concern: do strong representations alone induce good calibration?

To address this, we evaluate in a frozen-feature setting. We extract latent embeddings using the pretrained CLIP ViT-B/32 (Radford et al., 2021) image encoder and train a lightweight 2-layer MLP (Equation 13) directly on these fixed features.

Importantly, the representation remains entirely fixed and only a small classifier is trained on top. This setup removes any confounding effects of representation learning or model capacity, and isolates the impact of target constructions (CE, LS, SSA-LS) on model calibration. We train on CIFAR-100 and ImageNet-1k, datasets of vastly different scale, as reported in Table 13.

We find that strong pretrained representations alone do not yield well-calibrated models. For ImageNet-1k, despite competitive accuracy (74.26%), the CE baseline exhibits poor calibration (ECE 6.76), indicating that high-quality features do not inherently produce reliable confidence estimates. Uniform label smoothing (LS) substantially improves calibration (ECE 2.25), and SSA-LS further significantly reduces ECE to 1.14, highlighting that how the smoothing mass is distributed between classes is critical. Importantly, even in strong representation regimes, incorporating representation-aware priors into target distributions remains essential for achieving well-calibrated predictions. The CIFAR-100 experiment follows identical trends. We further expand our evaluation of the frozen-feature setting below in Appendix H.

H Matched smoothing strength comparisons

In our main experiments, all methods that take a smoothing strength share the same ϵ, which provides a controlled comparison but could favor methods whose optimal ϵ happens to lie near the shared value. To rule this out, we repeat the frozen-feature experiment, an MLP probe trained

Table 13: Calibration and accuracy comparison (mean ± std) on CIFAR-100 and ImageNet-1k using a lightweight MLP trained on frozen features extracted from CLIP encoder. The model trained on Cross Entropy (CE) targets achieves competitive accuracy but poor calibration. A well-structured latent space does not guarantee well-calibrated models.

(a) CIFAR-100

Method	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
CE	9.03 ± 0.20	78.12 ± 0.18	0.86 ± 0.00	19.98 ± 0.74
LS	2.83 ± 0.23	$\mathbf{78.82 \pm 0.13}$	$\mathbf{0.78 \pm 0.00}$	$\mathbf{7.31 \pm 1.49}$
SSA-LS	$\mathbf{2.03 \pm 0.19}$	78.17 ± 0.20	$\mathbf{0.78 \pm 0.00}$	12.52 ± 0.57

(b) ImageNet-1k

Method	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
CE	6.76 ± 0.11	74.26 ± 0.09	1.01 ± 0.00	12.86 ± 0.85
LS	2.25 ± 0.08	$\mathbf{74.73 \pm 0.09}$	$\mathbf{0.98 \pm 0.00}$	$\mathbf{5.65 \pm 0.43}$
SSA-LS	$\mathbf{1.14 \pm 0.10}$	74.34 ± 0.10	$\mathbf{0.98 \pm 0.00}$	6.14 ± 0.24

on frozen CLIP ViT-B/32 features, on ImageNet-1k and additionally on CIFAR-100, and sweep $\epsilon \in \{0.01, 0.02, 0.05, 0.10\}$ for every ϵ-dependent method (LS, LS++, ILS, and SSA-LS). Methods without a smoothing strength (CE, OLS, MbLS, CALS, ACLS) are reported once for reference. This lets each method be compared at its own best ϵ rather than a shared one.

With the sparsity factor k selected per ϵ on a held-out validation split, SSA-LS attains the lowest ECE of all methods on both datasets, even when every baseline is evaluated at its own best smoothing strength: 1.14 on ImageNet-1k at ϵ=0.05, below any result attained by LS, LS++, or ILS at any ϵ, and 1.83 on CIFAR-100, compared with a best of 2.50 for uniform LS. The selected neighborhoods follow a consistent pattern across both datasets. At small smoothing strengths ($\epsilon \in \{0.01, 0.02\}$), validation selects large neighborhoods (k=200 on ImageNet-1k, k=25 on CIFAR-100) and SSA-LS performs on par with uniform LS (e.g., 2.17 vs. 2.29 on ImageNet-1k and 2.71 vs. 2.57 on CIFAR-100 when $\epsilon = 0.02$), consistent with our finding that large-k SSA-LS approaches uniform smoothing. At larger strengths ($\epsilon \in \{0.05, 0.10\}$), where substantially more mass is redistributed, validation selects tight neighborhoods (k=10 on ImageNet-1k; k=3 and k=1 on CIFAR-100 for $\epsilon = 0.05$ and 0.1, respectively) and SSA-LS improves markedly over LS (1.14 vs. 2.25 and 6.45 vs. 8.22 on ImageNet-1k; 1.83 vs. 3.01 and 4.57 vs. 9.71 on CIFAR-100). Importantly, SSA-LS remains competitive at every ϵ provided k is chosen appropriately. These calibration gains are not bought with accuracy: at its best-calibrated setting, SSA-LS matches the accuracy of cross-entropy on both datasets (74.34% vs. 74.26% on ImageNet-1k; 79.94% vs. 79.92% on CIFAR-100), and accuracy stays within a narrow band across all smoothing methods. OLS attains the highest accuracy but fails to produce a well-calibrated model: it is outperformed by uniform LS on both datasets and is significantly miscalibrated compared to cross-entropy on ImageNet-1k (13.54 vs. 6.76).

I Sensitivity to the softmax temperature

We set $T = 1.2$ as a reasonable default for a moderately-weighted redistribution. In this section, we conduct a sensitivity analysis on T and investigate the effects of T on calibration and accuracy.

Table 15 shows that calibration is stable across the full range of T: every setting from $T = 0.5$ to 10^6 lands within a tight band, ECE $3.50 - 3.68$ for ResNet-18 and $4.14 - 4.66$ for DenseNet-121. All of these lie well below the CE (4.49 for ResNet-18 and 10.26 for DenseNet-121) and LS (5.84 for ResNet-18 and 5.89 for DenseNet-121) baselines. In practice, this means that T is not a sensitive hyperparameter and can be left at a reasonable default.

The one informative extreme is $T = 10^6$, where the similarity weights collapse to a uniform distribution over the support, reducing SSA-LS to uniform smoothing restricted over the sematic support set. This limit remains among the best-calibrated settings in both tables, reinforcing that *where* the smoothing mass is placed matters more than *how* it is placed.

Table 14: Comparison of methods across ϵ values on ImageNet-1k and CIFAR-100. SSA-LS maintains comparable calibration across the ϵ spectrum and attains the global minimum.

(a) ImageNet-1k

ϵ	Method	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
-	CE	6.76 ± 0.11	74.26 ± 0.09	1.01 ± 0.00	12.86 ± 0.85
	OLS	13.54 ± 0.12	**74.97 ± 0.12**	1.03 ± 0.00	20.14 ± 0.26
	MBLS	4.37 ± 2.50	74.72 ± 0.10	0.99 ± 0.03	9.15 ± 3.23
	CALS	19.66 ± 15.21	74.53 ± 0.40	1.22 ± 0.34	25.32 ± 17.42
	ACLS	2.91 ± 3.15	74.66 ± 0.06	0.97 ± 0.03	7.43 ± 2.58
0.01	LS	3.98 ± 0.09	74.52 ± 0.08	0.97 ± 0.00	7.60 ± 0.57
	LS++	5.74 ± 0.05	74.32 ± 0.03	0.99 ± 0.00	11.09 ± 0.66
	ILS	6.82 ± 0.11	74.21 ± 0.10	1.01 ± 0.00	12.35 ± 0.93
	SSA-LS	4.06 ± 0.05	74.46 ± 0.07	0.97 ± 0.00	7.59 ± 0.46
0.02	LS	2.17 ± 0.07	74.58 ± 0.08	0.96 ± 0.00	13.57 ± 16.87
	LS++	4.48 ± 0.16	74.33 ± 0.14	0.97 ± 0.00	9.14 ± 0.96
	ILS	6.78 ± 0.12	74.24 ± 0.10	1.01 ± 0.00	12.97 ± 0.71
	SSA-LS	2.29 ± 0.06	74.52 ± 0.06	0.97 ± 0.00	5.77 ± 1.13
0.05	LS	2.25 ± 0.08	74.73 ± 0.09	0.98 ± 0.00	**5.65 ± 0.43**
	LS++	1.46 ± 0.06	74.52 ± 0.05	**0.95 ± 0.00**	8.50 ± 10.75
	ILS	6.80 ± 0.14	74.24 ± 0.15	1.01 ± 0.00	12.57 ± 0.71
	SSA-LS	**1.14 ± 0.10**	74.34 ± 0.10	0.98 ± 0.00	6.14 ± 0.24
0.10	LS	8.22 ± 0.09	74.88 ± 0.08	1.03 ± 0.00	12.77 ± 0.36
	LS++	6.57 ± 0.09	74.74 ± 0.11	0.97 ± 0.00	10.72 ± 1.30
	ILS	6.76 ± 0.10	74.23 ± 0.09	1.01 ± 0.00	12.68 ± 0.69
	SSA-LS	6.45 ± 0.05	74.23 ± 0.08	1.02 ± 0.00	12.20 ± 3.66

(b) CIFAR-100

ϵ	Method	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
-	CE	8.87 ± 0.13	79.92 ± 0.07	0.81 ± 0.00	22.46 ± 0.91
	OLS	6.88 ± 0.17	**81.11 ± 0.15**	**0.69 ± 0.00**	14.44 ± 1.60
	MBLS	2.00 ± 0.23	80.82 ± 0.15	0.70 ± 0.00	9.84 ± 2.64
	CALS	15.23 ± 0.15	80.93 ± 0.08	0.82 ± 0.00	23.80 ± 1.16
	ACLS	2.29 ± 0.20	80.86 ± 0.13	0.69 ± 0.00	**9.53 ± 3.00**
0.01	LS	4.84 ± 0.10	80.31 ± 0.07	0.72 ± 0.00	14.06 ± 1.91
	LS++	6.99 ± 0.18	80.05 ± 0.12	0.75 ± 0.00	18.58 ± 1.98
	ILS	8.79 ± 0.16	79.97 ± 0.12	0.81 ± 0.00	22.22 ± 0.54
	SSA-LS	5.22 ± 0.14	80.09 ± 0.19	0.73 ± 0.00	15.98 ± 1.55
0.02	LS	2.50 ± 0.08	80.52 ± 0.11	0.70 ± 0.00	12.81 ± 5.19
	LS++	4.58 ± 0.07	80.17 ± 0.05	0.71 ± 0.00	12.90 ± 1.46
	ILS	8.77 ± 0.15	79.94 ± 0.09	0.81 ± 0.00	22.73 ± 1.01
	SSA-LS	2.71 ± 0.13	80.29 ± 0.17	0.71 ± 0.00	13.58 ± 1.28
0.05	LS	3.01 ± 0.13	80.76 ± 0.13	0.72 ± 0.00	9.94 ± 1.63
	LS++	2.01 ± 0.18	80.49 ± 0.14	0.68 ± 0.00	12.31 ± 0.52
	ILS	8.68 ± 0.17	79.89 ± 0.12	0.80 ± 0.00	22.77 ± 1.77
	SSA-LS	**1.83 ± 0.14**	79.94 ± 0.12	0.72 ± 0.00	18.21 ± 10.59
0.10	LS	9.71 ± 0.10	80.96 ± 0.06	0.78 ± 0.00	18.06 ± 1.78
	LS++	9.36 ± 0.14	80.89 ± 0.08	0.72 ± 0.00	14.83 ± 0.68
	ILS	8.43 ± 0.21	79.95 ± 0.15	0.80 ± 0.00	22.04 ± 0.81
	SSA-LS	4.57 ± 0.07	78.62 ± 0.06	0.81 ± 0.01	11.65 ± 1.49

Table 15: Effect of varying softmax temperature T on calibration and accuracy (ResNet-18 and DenseNet-121 on CIFAR-100, $k = 5$). T can be used to flatten or sharpen the redistribution among the top-k neighbors. Sharpened ($T = 0.5$), flattened ($T = 1.2, 5.0$), original ($T = 1.0$), and unweighted ($T = 10^6$) redistribution perform comparably with no consistent trend among them.

(a) ResNet-18

Method	Temperature (T)	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
CE	–	4.49 ± 0.16	78.45 ± 0.23	0.87 ± 0.01	14.58 ± 5.41
LS	–	5.84 ± 0.31	**79.13 ± 0.10**	0.93 ± 0.00	20.60 ± 1.56
SSA-LS	0.5	3.68 ± 0.22	79.08 ± 0.19	0.83 ± 0.01	**14.50 ± 2.04**
SSA-LS	1.0	3.54 ± 0.17	78.98 ± 0.36	0.83 ± 0.01	14.67 ± 1.06
SSA-LS	1.2	3.55 ± 0.22	79.02 ± 0.23	0.83 ± 0.01	15.15 ± 1.28
SSA-LS	5.0	3.63 ± 0.19	78.90 ± 0.17	**0.83 ± 0.00**	15.61 ± 2.26
SSA-LS	10^6	**3.50 ± 0.15**	78.90 ± 0.31	0.83 ± 0.01	14.75 ± 1.21

(b) DenseNet-121

Method	Temperature (T)	ECE ↓	Accuracy ↑	NLL ↓	MCE ↓
CE	–	10.26 ± 0.33	**67.99 ± 0.41**	1.32 ± 0.01	21.29 ± 0.94
LS	–	5.89 ± 0.53	67.64 ± 0.16	1.35 ± 0.01	15.66 ± 1.92
SSA-LS	0.5	4.44 ± 0.51	67.66 ± 0.41	**1.25 ± 0.01**	**12.06 ± 1.27**
SSA-LS	1.0	4.50 ± 0.35	67.73 ± 0.28	1.26 ± 0.02	12.67 ± 1.82
SSA-LS	1.2	4.66 ± 0.31	67.72 ± 0.49	1.26 ± 0.01	11.30 ± 1.73
SSA-LS	5.0	4.57 ± 0.43	67.75 ± 0.25	1.26 ± 0.01	11.37 ± 0.81
SSA-LS	10^6	**4.14 ± 0.36**	67.57 ± 0.46	1.26 ± 0.01	27.77 ± 36.68