Towards Near-Optimal Regularization in Deep Learning via the Inverse-Rate Regularizer

HU Jiahao [1] Luis A. Ortega [2] and Taous-Meriem Laleg-Kirati [3] Andrés R. Masegosa[4]

[1]King Abdullah University of Science and Technology,Saudi Arabia, [2]Universidad Autónoma de Madrid, Spain
[3]Université Paris-Saclay, Inria, France. [4]Aalborg University,Denmark

Motivation

- **Empirical fact:** Modern architectures often reach near-zero training loss but maintain low test error.
- **Limit of classic theory:** VC / Rademacher complexity is distribution-agnostic and have a *hard* time explaining "benign overfitting."
- **New insight (PAC-Chernoff [Masegosa & Ortega, 2025]):** The *inverse rate function*, which is distribituion dependent, tightly approximates the generalization gap for interpolators.
- **Optimal Regularization:** The *inverse rate function* is an optimal regularizer for interpolators [Masegosa & Ortega, 2025, Theorem 6.1].
- **Problem:** Computing that *inverse rate function* needs unknown distribution information—no oracle at train time.
- **Our solution:** A fast, mini-batch estimator that can be plugged into SGD as the *Inverse-Rate Regularizer*, yielding a theoretically grounded regualizarer with promising results.
- **Our Contribution:** A first step towards operationalise a near-optimal theoretical regularizer with near-optimal performance.

Preliminaries

Step 1: Cumulant-Generating Function (CGF)

$$J_\theta(\lambda) \;=\; \ln\, \mathbb{E}_{(x,y)\sim\nu}\big[e^{\lambda\,(L(\theta)\,-\,\ell(y,x,\theta))}\big]$$

Step 2: Rate Function (Large-Deviation speed)

$$I_\theta(a) \;=\; \sup_{\lambda\geq 0}\big\{\lambda\, a - J_\theta(\lambda)\big\}$$

Step 3: Inverse Rate Function (complexity term)

$$I_\theta^{-1}(s) \;=\; \inf_{\lambda>0}\; \frac{s + J_\theta(\lambda)}{\lambda}$$

PAC–Chernoff bound:

$$L(\theta) \;\leq\; \hat{L}(\theta) \;+\; I_\theta^{-1}\Big(\tfrac{1}{n}\,\ln\tfrac{|\Theta|}{\delta}\Big)$$

Theory (PAC–Chernoff): If a model θ *interpolates* the data, meaning $\hat{L}(\theta)\leq\varepsilon$, then

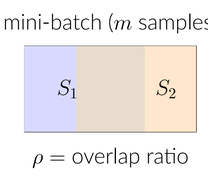
$$\big|\, L(\theta) - I_\theta^{-1}(\tfrac{1}{n}\,\ln\tfrac{|\Theta|}{\delta})\big| \;\leq\; \varepsilon,$$

so the *inverse rate function* $I_\theta^{-1}(\frac{1}{n}\,\ln\frac{|\Theta|}{\delta})$ is an **exact expression for the generalization gap**.

Consequence ⇒ Optimal Regularizer for Interpolators

$$\min_\theta\; \Big[\,\hat{L}(\theta) \;+\; I_\theta^{-1}(\tfrac{1}{n}\,\ln\tfrac{|\Theta|}{\delta})\,\Big] \quad\Longrightarrow\quad \text{near-optimal generalization}$$

Method: Overlapping-Batch Estimator & Adaptive Weights



Overlapping-batch CGF estimator

$$\widehat{J}_\theta(\lambda) = \ln\frac{1}{|S_1|}\sum_{(x,y)\in S_1} e^{-\lambda\,\ell(y,x,\theta)} \;+\; \lambda\,\frac{1}{|S_2|}\sum_{(x,y)\in S_2}\ell(y,x,\theta)$$

Empirical inverse-rate regularizer

$$\widehat{\mathcal{I}}_\theta^{-1}(s) = \min_{\lambda>0}\frac{s+\widehat{J}_\theta(\lambda)}{\lambda}$$

Implicit per-sample weight (log-loss)

$$\widehat{\mathcal{I}}_\theta^{-1}(s) = \widehat{\mathbb{E}}[w_{y,x}\cdot\ell(y,x,\theta)] \qquad \text{with} \qquad w_{y,x} = 1 - \frac{p(y|x,\theta)^{\lambda^\star}}{\widehat{\mathbb{E}}[p(y|x,\theta)^{\lambda^\star}]}, \tag{1}$$

- $w_{y,x} > 0$: under-confident examples.
- $w_{y,x} < 0$: over-confident predictions.
- $\mathbb{E}[w_{y,x}] = 0$ perfectly confident examples.

Connections – popular regularizers through the inverse-rate lens

Method / family	Penalty or objective (typical form)	Relation to inverse-rate I_θ^{-1}
ℓ_2 weight decay	$r(\theta) = \|\theta\|_2$	$I_\theta^{-1}(s) \leq \sqrt{2M}\,\|\theta\|_2$ (Prop. 6.2)
Input-gradient penalty	$r(\theta) = \mathbb{E}\big[\|\nabla_x\ell\|_2^2\big]$	$I_\theta^{-1}(s) \leq \sqrt{sM}\,\sqrt{\mathbb{E}\|\nabla_x\ell\|_2^2}$
Lipschitz constraint	Lip(θ) bounded via spectral norm	If $\|\nabla_x\ell\|_2 \leq$ Lip(θ) the same upper bound on I_θ^{-1} applies
KL-DRO ($D_{KL}\leq s$)	$\sup_{q:\, D_{KL}(q\|\hat{P}_n)\leq s} \hat{L}_q(\theta)$	Dual objective equals $\hat{L}(\theta) + I_\theta^{-1}(s)$ (Thm. 3)
Focal / Meta-Weight	Per-sample weight $(1-p)^\gamma$ or learned $w_{y,x}$	Heuristic approximation of the inverse-rate weight $w_{y,x} = 1 - \frac{p^{\lambda^\star}}{\widehat{\mathbb{E}}[p^{\lambda^\star}]}$

Results

- Empirical inverse-rate regularizer $\widehat{\mathcal{I}}_\theta^{-1}(s)$ was unestable due to problems estimation optimal $\lambda^\star$. We treated $\lambda^\star$ as tuneable hyper-paramter

$$\widehat{\mathcal{I}}_\theta^{-1}(s) \approx \frac{s+\widehat{J}_\theta(\lambda^\star)}{\lambda^\star}$$

- **Inverse-Rate Regularization Yields Better-Calibrated Models** Table 1 present a summary of results for a subset of the 19 network architectures. Last rows summarizes the comparison by reporting, for each metric, how often our method outperforms the baseline across the full set of 19 models. The inverse-rate regularizer slightly lowers Top-1 accuracy but consistently improves NLL, variance, calibration, and Top-5 accuracy, yielding more reliable predictions overall.

Table 1. Comparison of the baseline (**Base**) method (SGD + L2 + data augmentation) versus our inverse-rate regularizer added on top (**Ours**).

CIFAR-10	Top1-Acc (↑)		Top5-Acc (↑)		NLL (↓)		Variance (↓)		ECE (↓)	
	Base	Ours	Base	Ours	Base	Ours	Base	Ours	Base	Ours
mobilenetv2_x0_75	0.935	**0.938**	**0.998**	0.998	0.264	**0.254**	1.341	**1.306**	0.040	**0.037**
repvgg_a0	**0.945**	0.938	0.998	**0.998**	**0.237**	0.247	1.327	**1.206**	**0.036**	0.037
resnet32	0.934	**0.935**	0.997	**0.998**	0.295	**0.250**	1.682	**1.205**	0.042	**0.036**
shufflenetv2_x0_5	**0.904**	0.891	0.996	**0.997**	0.335	**0.329**	1.395	**0.940**	0.048	**0.033**
vgg19_bn	**0.941**	0.934	0.997	**0.997**	0.332	**0.315**	2.187	**1.878**	**0.049**	0.049
All Models		8/19		13/19		14/19		14/19		11/19

CIFAR-100	Top1-Acc (↑)		Top5-Acc (↑)		NLL (↓)		Variance (↓)		ECE (↓)	
	Base	Ours	Base	Ours	Base	Ours	Base	Ours	Base	Ours
mobilenetv2_x0_75	**0.743**	0.698	**0.930**	0.920	1.080	**1.053**	4.414	**2.596**	0.110	**0.044**
repvgg_a0	**0.755**	0.739	0.931	**0.931**	**1.056**	1.075	**4.272**	4.452	**0.094**	0.108
resnet32	0.696	**0.699**	0.910	**0.920**	1.330	**1.106**	5.851	**3.448**	0.140	**0.077**
shufflenetv2_x0_5	**0.682**	0.674	0.902	**0.905**	1.297	**1.199**	4.754	**3.254**	0.118	**0.054**
vgg19_bn	**0.743**	0.697	**0.901**	0.891	1.780	**1.400**	12.966	**6.664**	0.194	**0.142**
All Models		1/19		13/19		16/19		15/19		14/19

- **Sensitivity of the Inverse-Rate regularizer to Hyper-parameters.**The results in table 2 show that $\rho = 0.5$ offers the best trade-off across metrics. It consistently outperforms the baseline in terms of NLL and NLL variance—achieving 12–17 wins out of 19 across datasets—while also maintaining strong Top-5 accuracy and competitive ECE.

Table 2. Aggregate win counts comparing the inverse-rate regularizer to the baseline (SGD + L2 + augmentation) across 19 architectures on CIFAR-10 (top) and CIFAR-100 (bottom). Each row corresponds to a specific setting of the hyper-parameters ρ and $\lambda^\star$. "Best" denotes per-model selection based on validation NLL. For each metric, the table reports how many times the regularized model outperformed the baseline across the 19 model architectures.

Architecture	ρ	$\lambda^\star$	Top1-Acc	Top5-Acc	NLL	Variance	ECE
CIFAR-10	Best	Best	8/19	13/19	14/19	14/19	11/19
	0.0	Best	0/19	6/19	6/19	19/19	17/19
	0.5	Best	0/19	12/19	12/19	19/19	14/19
	1.0	Best	9/19	9/19	10/19	11/19	10/19
CIFAR-100	Best	Best	1/19	13/19	16/19	15/19	14/19
	0.0	Best	0/19	5/19	6/19	17/19	16/19
	0.5	Best	1/19	12/19	17/19	15/19	13/19
	1.0	Best	2/19	13/19	14/19	15/19	13/19

Conclusions, Limitations and Future Works

Contribution:

- Introduced the *inverse-rate regularizer* — a practical estimator of the theoretically optimal inverse rate function — integrated into first-order optimisation.
- Improved probabilistic quality in 19 CNN architectures on CIFAR-10/100; unified and extended existing regularisers.
- Core insight: better estimators of the inverse rate function can lead to principled, near-optimal regularization.

Limitations:

- Mini-batch estimator is biased and has variance issues in small-sample regimes.
- Requires tuning of two hyperparameters.
- Experiments limited to log-loss image classification — transferability to other tasks is untested.

Future Directions:

- Use held-out validation data to reduce selection bias in inverse rate estimation.
- Analyse bias–variance trade-offs and guarantees when n is limited.
- Explore application to regression, structured prediction, and large-scale pre-training.

References

[1] Andrés R. Masegosa and Luis A. Ortega.
Pac-chernoff bounds: Understanding generalization in the interpolation regime.
Journal of Artificial Intelligence Research, 82:503–562, 2025.