OVLR: Efficient, Scalable, and Robust Training via Output-Level Variance-Reduced Likelihood Ratio

Minhao Zou[1,2,3], **Tao Ren**[4], **Jinyang Jiang**[4], **Rui Tao**[4], **Zehao Li**[4], **Jiale Fu**[5], **Hui Shao**[6], **Xianhua Liu**[1], **Yijie Peng**[2,3,4,5,7,8,9]

[1]School of Computer Science, Peking University [2]AdvancedInstitute of Information Technology, Peking University
[3]Hangzhou IndEngine Intelligence Co., Ltd. [4]Guanghua School of Management, Peking University
[5]Dundee International Institute, Central South University [6]International Business School, Zhejiang University
[7]Department of AI for Management, School of Management and Engineering, Nanjing University [8]Xiangjiang Laboratory
[9]Wuhan Institute of Artificial Intelligence, Peking University Correspondence to: Yijie Peng <pengyijie@pku.edu.cn>

ICML International Conference On Machine Learning

Motivation

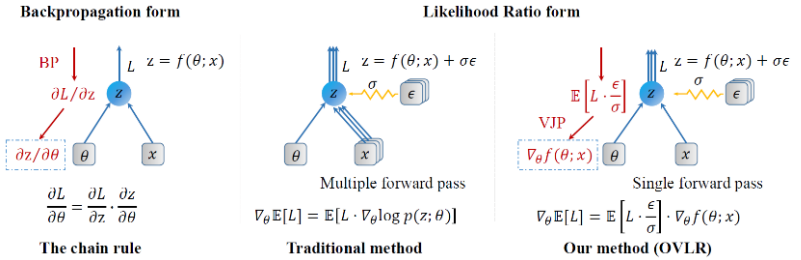


Figure 1. Comparison of Gradient Estimation Paradigms.

• Motivation

- Backpropagation (BP) fails for gradient-agnostic objectives (e.g., 0-1 loss, truncated loss) – gradients are zero, undefined, or uninformative.
- Existing LR / ES methods perturb parameters or hidden layers (high-dimensional) → cost scales with model size → not scalable for deep networks.
- Thus, we need a scalable estimator that: (i) directly optimizes gradient-agnostic targets (where BP fails), and (ii) matches BP's efficiency (where LR/ES fails).

• Contribution

- **OVLR framework** – Shifts Perturbation from parameter space → low-dimensional output space.
  - Single deterministic forward pass + one Vector-Jacobian Product (VJP).
  - Speed & memory on par with BP
- **Variance reduction** – Output-level repetition + antithetic sampling → gradient variance decoupled from $1/\sigma^2$ (near-constant, $R^2 \approx 0$).
- **Theoretical guarantees** – Unbiased estimator, variance upper bound, convergence rate O $(1/\sqrt{K})$.
- **Empirical excellence** – OVLR handles both worlds: standard differentiable tasks (with BP-level efficiency) and gradient-agnostic tasks (where BP fails).
- **Plug-and-play** – Works with any architecture (ResNet, ViT, Mamba). No model modification. Open-source (PyTorch).

Method

• Problem Formulation

$$L(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[L(f(\theta;x),y)]$$

Goal: Minimize $\mathbb{E}_z[L(z,y)]$ by estimating $\nabla_\theta \mathbb{E}_z[L(z,y)]$

• Variance Reduction

◦ **Output-level repetition**

Reuse the same forward pass Z for n noise samples → no n× backbone cost.

◦ **Antithetic sampling (AS)**

Pair ϵ and $-\epsilon$ → cancels odd-order noise, variance nearly constant vs $1/\sigma^2$.

Algorithm 1 Vectorized Training with OVLR

Input: Mini-batch (X, Y), parameters θ, noise scale σ, sample count n, batch size B, feature dimension d
Output: Updated model parameters θ'
Forward Pass:
$Z = f(\theta; X)$ {Single deterministic backbone pass; $Z \in \mathbb{R}^{B\times d}$}
$\tilde{Z} = Z.\text{repeat}(n, 1)$, $\tilde{Y} = Y.\text{repeat}(n)$ {Vectorized expansion to $N = B \times n$ samples}
Symmetric Sampling:
Sample $\epsilon_{1:N/2} \sim \mathcal{N}(0, I_d)$, set $\epsilon_{N/2+1:N} = -\epsilon_{1:N/2}$ {Antithetic sampling}
$Z_{perturbed} = \tilde{Z} + \sigma\epsilon$ {Batch-wise perturbation of features with Gaussian noise}
Loss & Gradient Construction:
$L = l(Z_{perturbed}, \tilde{Y})$ {Batch-wise black-box loss evaluation}
$v = \frac{1}{N\sigma}(L \odot \epsilon)$ {Vectorized output-level gradient signal; $\odot$ is element-wise product}
Backward Pass:
$\nabla_\theta L = \text{VJP}(Z, \theta, v)$ {Single VJP through backbone network (equivalent to Z.backward(v) in PyTorch)}
$\theta' = \text{Optimizer}(\theta, \nabla_\theta L)$ {Update parameters using optimizer}
Return θ'

• For Gaussian (Default)

$$\nabla_\theta \mathbb{E}_z[L(z,y)] = \mathbb{E}_{\epsilon\sim N(0,1)}\left[L(z,y)\cdot\frac{\epsilon}{\sigma}\right]\cdot\nabla_\theta\,\mu(\theta)$$

- $\mu(\theta) = f(\theta; x)$ – deterministic output
- σ – noise scale
- $z = \mu(\theta) + \sigma\epsilon$ – perturbed output
- $L(z, y)$ – general loss function
- $\epsilon \sim N(0,1)$ – Gaussian noise
- y – ground truth label

• For Laplace

$$\nabla_\theta \mathbb{E}_z[L(z,y)] = \mathbb{E}_{\epsilon\sim Laplace(0,1/\sqrt{2})}\left[L(z,y)\cdot\frac{\sqrt{2}\cdot sign(\epsilon)}{\sigma}\right]\cdot\nabla_\theta\,\mu(\theta)$$

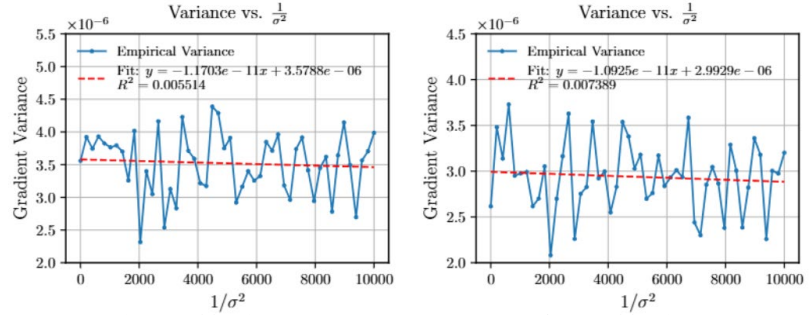
• For Student-t

$$\nabla_\theta \mathbb{E}_z[L(z,y)] = \mathbb{E}_{\epsilon\sim t_\nu}\left[L(z,y)\cdot\frac{(v+1)\epsilon}{(v-2+\epsilon^2)\sigma}\right]\cdot\nabla_\theta\,\mu(\theta)$$

Table 1. Computational and memory profiles during training.

Method	Forward Cost	Backward Cost	Memory Usage
Standard Backpropagation	$T_\mu + T_L$	$\tilde{T}_\mu + \tilde{T}_L$	$M_\mu + M_L + M_\theta$
Input-level Repetition	$n \cdot T_\mu + n \cdot T_L$	$n \cdot \tilde{T}_\mu + T_{\text{vec}}$	$n \cdot M_\mu + n \cdot M_{\text{noise}} + M_\theta$
OVLR (Ours)	$T_\mu + n \cdot T_L$	$\tilde{T}_\mu + T_{\text{vec}}$	$M_\mu + n \cdot M_{\text{noise}} + M_\theta$

Experiments



(a) Repetition (100) + AS (b) Repetition (1000) + AS

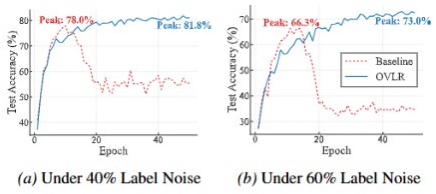
Figure 2. Variance Reduction Efficacy of Different Strategies.

Table 2. Ablation Study on Variance Reduction Strategies and Noise Robustness

Dataset	Mode		Repetition (n)								Noise Scale σ					
	Vanilla	AS	1	20	50	100	200	300	400	500	0.01	0.1	0.5	1	2	5
CIFAR10	88.15	88.47	30.27	86.86	88.06	88.16	88.47	88.34	87.35	88.40	87.96	88.33	87.21	88.47	87.76	88.20
CIFAR100	36.62	60.06	1.85	38.43	54.82	59.77	60.06	61.20	61.73	62.45	60.94	59.32	59.92	60.06	60.26	58.98
TinyImageNet	18.78	46.90	0.71	22.05	37.91	44.94	46.90	45.68	45.69	46.39	46.92	46.36	45.24	46.90	45.99	47.31

Table 3. OVLR Achieves Competitive Accuracy with Negligible Overhead.

Model	CIFAR100		TinyImageNet	
	Accuracy (BP/OVLR)	Time (s) (BP/OVLR)	Accuracy (BP/OVLR)	Time (s) (BP/OVLR)
ResNet-18	63.10 / 62.10 -1.00	48.85 / 49.19 +0.34	47.27 / 46.83 -0.44	97.87 / 98.22 +0.35
DenseNet-121	65.92 / 65.31 -0.61	153.62 / 153.53 -0.09	53.88 / 52.86 -1.02	307.10 / 306.92 -0.18
ViT-B/16	86.79 / 88.17 +1.38	373.06 / 373.75 +0.69	79.72 / 79.89 +0.17	746.74 / 748.05 +1.31



(a) Under 40% Label Noise (b) Under 60% Label Noise

Figure 3. OVLR Filters Noisy Gradients for Robust Convergence.

Table 4. Memory Profile

Model	CIFAR100		TinyImageNet	
	Allocated (BP/OVLR)	Reserved (BP/OVLR)	Allocated (BP/OVLR)	Reserved (BP/OVLR)
ResNet-18	1,565 / 1,587 +22	3,466 / 3,466 0	1,565 / 1,625 +60	3,468 / 3,454 -14
DenseNet-121	8,112 / 8,132 +20	9,334 / 9,338 +4	8,113 / 8,174 +61	9,348 / 9,418 +70
ViT-B/16	10,082 / 10,102 +20	10,394 / 10,406 +12	10,083 / 10,144 +61	10,402 / 10,504 +102

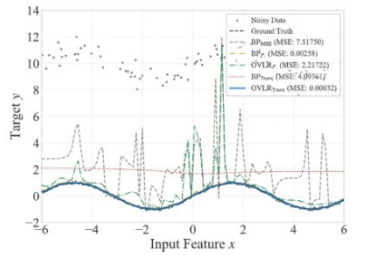


Figure 4. OVLR Overcomes Gradient Vanishing in Truncated Loss Optimization.