SASSHA: Sharpness-aware Adaptive Second-order Optimization with Stable Hessian Approximation

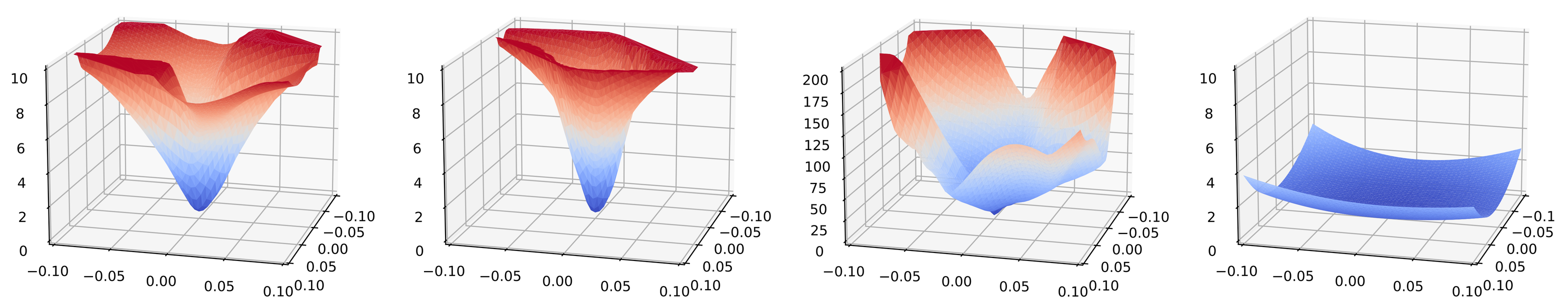
Dahun Shin* [1] Dongyeop Lee* [1] Jinseok Chung [1] Namhoon Lee [1]

[1]Pohang University of Science and Technology

Main Contribution

- observe that approximate second-order methods tend to converge to sharp minima, which may explain their degraded generalization performance.
- propose SASSHA, an adaptive second-order method designed to stably converge to flat solutions, thereby improving generalization.
- SASSHA demonstrates strong performance across diverse deep learning tasks, including vision, language, and robustness to label noise.
- SASSHA achieves this efficiently via lazy Hessian updates without performance degradation, due to its trajectory staying within regions of slowly changing curvature.

Second-order methods converge to sharp minima



(a) AdaHessian (b) Sophia-H (c) Shampoo (d) SASSHA

	Sharpness				Generalization	
	$\lambda_{max}(H)$	$\mathrm{tr}(H)_{\times 10^3}$	δL_{grad}	$\delta L_{\text{avg}\times 10^{-3}}$	L_{val}	Acc$_{\text{val}}$ (%)
SGD	$265_{\pm 25.00}$	$7.290_{\pm 0.300}$	$0.703_{\pm 0.132}$	$1.310_{\pm 1.030}$	$1.260_{\pm 0.001}$	$69.32_{\pm 0.19}$
AdaHessian	$11992_{\pm 5779}$	$46.94_{\pm 17.60}$	$4.119_{\pm 1.136}$	$12.50_{\pm 6.080}$	$1.377_{\pm 0.070}$	$68.06_{\pm 0.22}$
Sophia-H	$22797_{\pm 10857}$	$68.15_{\pm 20.19}$	$8.130_{\pm 3.082}$	$19.19_{\pm 6.380}$	$1.463_{\pm 0.022}$	$67.76_{\pm 0.37}$
Shampoo	$436374_{\pm 9017}$	$6823_{\pm 664.7}$	$73.27_{\pm 12.51}$	$49307489_{\pm 56979794}$	$1.386_{\pm 0.010}$	$64.08_{\pm 0.46}$
SASSHA	$107_{\pm 40.00}$	$1.870_{\pm 0.650}$	$0.238_{\pm 0.088}$	$0.650_{\pm 0.860}$	$0.961_{\pm 0.005}$	$72.14_{\pm 0.16}$

- Approximate second-order optimizers tend to yield minima of high sharpness and worse generalization compared to SGD. However, SASSHA effectively recovers this [a].
- provide theoretical explanation for this phenomenon through linear stability analysis:

Corollary 4.6: The linearly stable fixed point $x^\star$ of SASSHA satisfies the following necessary conditions:

$$0 \leq a(1+\rho a) \leq \frac{2\epsilon}{\eta}, \quad 0 \leq s_2^2 \leq \frac{\epsilon^2}{\eta^2 - 2\eta\rho\epsilon}, \quad 0 \leq s_3^2 \leq \frac{\epsilon^2}{2\eta^2\rho}, \quad 0 \leq s_4^2 \leq \frac{\epsilon^2}{\eta^2\rho^2},$$

where $a = \lambda_{max}(\mathbb{E}[H_\xi])$ and $s_k = \lambda_{max}((\mathbb{E}[H_\xi^k] - \mathbb{E}[H_\xi]^k)^{1/k})$ are the sharpness and the non-uniformity of the stochastic Hessian measured with the k-th moment, respectively.

In contrast, standard approximate second-order methods with $P \approx H^{-1}$ impose no constraint on sharpness.

Method: SASSHA

Update rule: We propose SASSHA, which performs the following update:

$$x_{t+1} = x_t - \widehat{H}_t(\tilde{x}_t)^{-1} \nabla f_t(\tilde{x}_t)$$

where $\widehat{H}$ denotes the diagonal of the Hessian, approximated via Hutchinson method.

[a]We find that the same trend holds for other workloads.

1. **Sharpness minimization:** $\tilde{x}_t = x_t + \rho \frac{\nabla f(x_t)}{\|\nabla f(x_t)\|_2}$.
   - encourage convergence to flat minima.
   - pose a risk of divergence by penalizing Hessian entries on average.
2. **Stable Hessian approximation:** $\widehat{H} \to |\widehat{H}|^{1/2}$
   - alleviate divergence and preserve the relative scale of the Hessian.
   - can be interpreted as a geometrical interpolation $H^\alpha (0 \leq \alpha \leq 1)$ which can balance between bias and variance of the population risk.
   - avoid saddle points or local maxima.
3. **Lazy Hessian update:** reusing previously computed Hessian for several iterations without performance degradation.

Algorithm 1 SASSHA algorithm

Input: Initial parameter x_0, learning rate $\{\eta_t\}$, moving average parameters β_1, β_2, Hessian update interval k, weight decay parameter λ
Set $m_{-1} = 0$, $D_{-1} = 0$
for $t = 1$ **to** T **do**
  $g_t = \nabla f_{\mathcal{B}}(x_t)$
  $\epsilon_t^\star = \rho g_t / \|g_t\|_2$
  $\tilde{g}_t = \nabla f_{\mathcal{B}}(x_t + \epsilon_t^\star)$
  $m_t = \beta_1 m_{t-1} + (1-\beta_1)\tilde{g}_t$
  $\overline{m}_t = m_t/(1-\beta_1^t)$
  if $t \bmod k = 1$ **then**
    $\tilde{H}_t = \widehat{H}(x_t + \epsilon_t^\star)$ ▷ Sharpness minimization
    $D_t = \beta_2 D_{t-1} + (1-\beta_2)|\tilde{H}_t|$
    $\overline{D}_t = \sqrt{D_t/(1-\beta_2^t)}$ ▷ Stable Hessian approximation
  else
    $\overline{D}_t = \overline{D}_{t-1}$ ▷ Lazy Hessian update
  end if
  $x_{t+1} = x_t - \eta_t \overline{D}_t^{-1} \overline{m}_t - \eta_t \lambda x_t$
end for

Theorem 4.4 (Convergence guarantee): Under standard smoothness and convexity assumptions, let $\{x_t\}$ be generated by SASSHA from given any initial point $x_0 \in \mathbb{R}^d$, with step sizes η_t and perturbation radii ρ_t satisfying $\sum_{t=1}^\infty \eta_t = \infty, \quad \sum_{t=1}^\infty \eta_t^2 < \infty, \quad \sum_{t=1}^\infty \rho_t^2 \eta_t < \infty$. Then,

$$\liminf_{t\to\infty} \|\nabla f(x_t)\| = 0$$

Experiments

Table 1. Image classification results.

Category	Method	CIFAR-10 ResNet-20	CIFAR-10 ResNet-32	CIFAR-100 ResNet-32	CIFAR-100 WRN-28-10	ImageNet ResNet-50	ImageNet ViT-s-32
First-order	SGD	$92.03_{\pm 0.32}$	$92.69_{\pm 0.06}$	$69.32_{\pm 0.19}$	$80.06_{\pm 0.15}$	$75.58_{\pm 0.05}$	$62.90_{\pm 0.36}$
	AdamW	$92.04_{\pm 0.11}$	$92.42_{\pm 0.13}$	$68.78_{\pm 0.22}$	$79.09_{\pm 0.35}$	$75.38_{\pm 0.08}$	$66.46_{\pm 0.15}$
	SAM SGD	$92.85_{\pm 0.07}$	$93.89_{\pm 0.13}$	$71.99_{\pm 0.20}$	$83.14_{\pm 0.13}$	$76.36_{\pm 0.16}$	$64.54_{\pm 0.63}$
	SAM AdamW	$92.77_{\pm 0.29}$	$93.45_{\pm 0.24}$	$71.15_{\pm 0.37}$	$82.88_{\pm 0.31}$	$76.35_{\pm 0.16}$	$68.31_{\pm 0.17}$
Second-order	AdaHessian	$92.00_{\pm 0.17}$	$92.48_{\pm 0.15}$	$68.06_{\pm 0.22}$	$76.92_{\pm 0.26}$	$73.64_{\pm 0.16}$	$66.42_{\pm 0.23}$
	Sophia-H	$91.81_{\pm 0.27}$	$91.99_{\pm 0.08}$	$67.76_{\pm 0.37}$	$79.35_{\pm 0.24}$	$72.06_{\pm 0.49}$	$62.44_{\pm 0.36}$
	Shampoo	$88.55_{\pm 0.83}$	$90.23_{\pm 0.24}$	$64.08_{\pm 0.46}$	$74.06_{\pm 1.28}$	*	*
	SASSHA	**92.98**$_{\pm 0.05}$	**94.09**$_{\pm 0.24}$	**72.14**$_{\pm 0.16}$	**83.54**$_{\pm 0.08}$	**76.43**$_{\pm 0.18}$	**69.20**$_{\pm 0.30}$

- SASSHA consistently outperforms the other methods for all workloads.

Table 2. Language pertraining and finetuning results.

	Pretrain / GPT1-mini	**Finetune** / SqeezeBERT						
	Wikitext-2 Perplexity	SST-2 Acc	MRPC Acc / F1	STS-B S/P corr.	QQP F1 / Acc	MNLI mat/m.mat	QNLI Acc	RTE Acc
AdamW	175.06	$90.29_{\pm 0.52}$	$84.56_{\pm 0.25}$ / $88.99_{\pm 0.11}$	$88.34_{\pm 0.15}$ / $88.48_{\pm 0.20}$	$89.92_{\pm 0.05}$ / $86.58_{\pm 0.11}$	$81.22_{\pm 0.07}$ / $82.26_{\pm 0.05}$	$89.93_{\pm 0.14}$	$68.95_{\pm 0.72}$
SAM AdamW	158.06	**90.52**$_{\pm 0.27}$	$83.25_{\pm 2.79}$ / $87.90_{\pm 2.21}$	$88.38_{\pm 0.01}$ / $88.79_{\pm 0.99}$	$90.26_{\pm 0.28}$ / $86.99_{\pm 0.31}$	$81.56_{\pm 0.18}$ / **82.46**$_{\pm 0.19}$	**90.38**$_{\pm 0.05}$	$68.83_{\pm 1.46}$
AdaHessian	407.69	$89.64_{\pm 0.13}$	$79.74_{\pm 4.00}$ / $85.26_{\pm 3.50}$	$86.08_{\pm 4.04}$ / $86.46_{\pm 4.06}$	$90.37_{\pm 0.05}$ / $87.07_{\pm 0.05}$	$81.33_{\pm 0.17}$ / $82.08_{\pm 0.02}$	$89.94_{\pm 0.12}$	$71.00_{\pm 1.04}$
Sophia-H	157.60	$90.44_{\pm 0.46}$	$85.78_{\pm 1.07}$ / $89.90_{\pm 0.82}$	$88.17_{\pm 1.07}$ / $88.53_{\pm 1.13}$	$90.70_{\pm 0.04}$ / $87.60_{\pm 0.06}$	**81.77**$_{\pm 0.18}$ / $82.36_{\pm 0.22}$	$90.12_{\pm 0.14}$	$70.76_{\pm 1.44}$
SASSHA	**122.40**	$90.44_{\pm 0.98}$	**86.28**$_{\pm 0.28}$ / **90.13**$_{\pm 0.161}$	**88.72**$_{\pm 0.75}$ / **89.10**$_{\pm 0.70}$	**90.91**$_{\pm 0.06}$ / **87.85**$_{\pm 0.09}$	$81.61_{\pm 0.25}$ / $81.71_{\pm 0.11}$	$89.85_{\pm 0.20}$	**72.08**$_{\pm 0.55}$

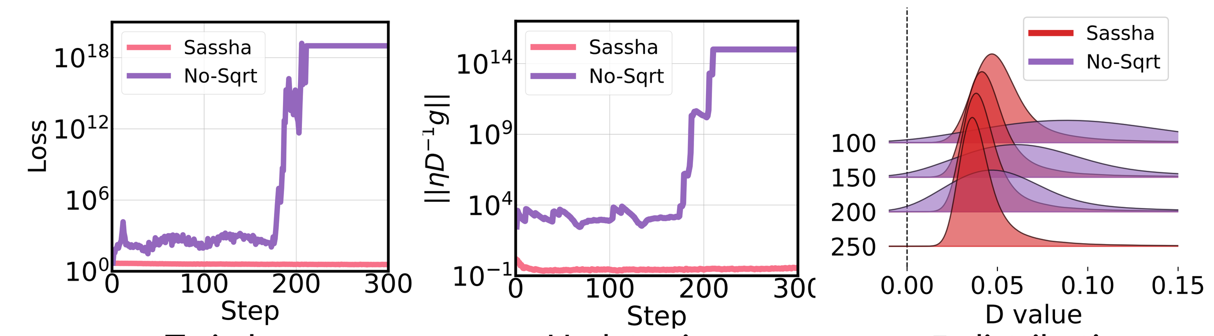
- (pretraining; left) SASSHA achieves the lowest perplexity.
- (finetuning; right) better than others, but competitive to Sophia-H.

Table 3. Robustness to label noise.

	CIFAR-10				CIFAR-100			
Noise level	0%	20%	40%	60%	0%	20%	40%	60%
SGD	$92.69_{\pm 0.06}$	$89.91_{\pm 0.87}$	$87.26_{\pm 0.40}$	$82.72_{\pm 1.59}$	$69.32_{\pm 0.19}$	$62.18_{\pm 0.06}$	$55.78_{\pm 0.55}$	$45.53_{\pm 0.78}$
SAM SGD	$93.89_{\pm 0.13}$	$92.27_{\pm 0.14}$	$90.11_{\pm 0.25}$	$85.79_{\pm 0.30}$	$71.99_{\pm 0.20}$	$65.53_{\pm 0.11}$	$61.20_{\pm 0.17}$	$51.93_{\pm 0.47}$
AdaHessian	$92.48_{\pm 0.15}$	$90.11_{\pm 0.01}$	$86.88_{\pm 0.04}$	$83.25_{\pm 0.01}$	$68.06_{\pm 0.22}$	$63.06_{\pm 0.25}$	$58.37_{\pm 0.13}$	$46.02_{\pm 1.96}$
Sophia-H	$91.99_{\pm 0.08}$	$89.93_{\pm 0.01}$	$87.30_{\pm 0.51}$	$82.78_{\pm 1.43}$	$67.76_{\pm 0.37}$	$62.34_{\pm 0.47}$	$56.54_{\pm 0.28}$	$45.37_{\pm 0.27}$
Shampoo	$90.23_{\pm 0.83}$	$88.14_{\pm 0.29}$	$85.15_{\pm 0.61}$	$81.16_{\pm 0.30}$	$64.08_{\pm 0.46}$	$58.85_{\pm 0.66}$	$53.82_{\pm 0.71}$	$42.91_{\pm 0.99}$
SASSHA	**94.09**$_{\pm 0.24}$	**92.49**$_{\pm 0.11}$	**90.29**$_{\pm 0.11}$	**86.50**$_{\pm 0.08}$	**72.14**$_{\pm 0.16}$	**66.78**$_{\pm 0.47}$	**61.97**$_{\pm 0.27}$	**53.98**$_{\pm 0.57}$

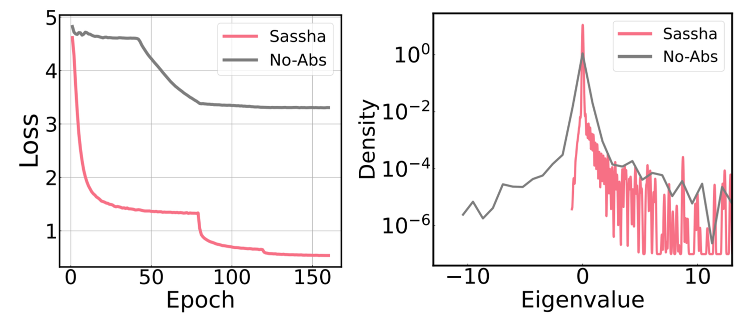
- SASSHA shows much robust performance under label noise.

Ablation: stable Hessian approximation



(a) Train loss (b) Update size (c) D distribution

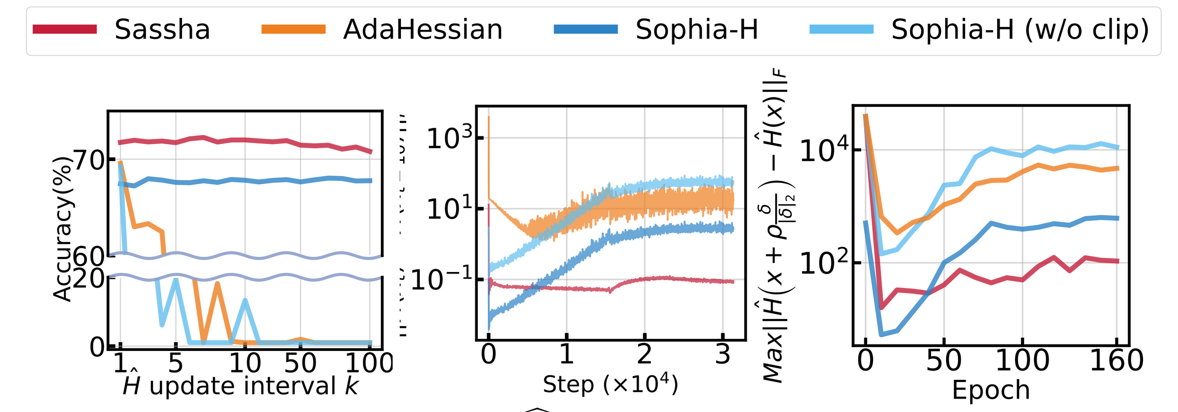
- (a) without sqrt training diverges when (b) the update size starts to spike.
- (c) individual entries in the diagonal Hessian; they gradually shifts toward zero values, as a result, D^{-1} becomes too large, creating overly large steps.



(a) Train loss (b) Hess. eigenspectrum

- (left) without absolute SASSHA does not converge well.
- (right) negative values in the Hessian distribution driving toward saddle points.

Robust to Lazy Hessian updates



(a) Lazy Hessian (b) $\widehat{H}$ change (c) Local sensitivity

- (a) SASSHA remains relatively effective with prolonged Hessian reusing
- (b) The difference between the current and previous Hessian is small.
- (c) Optimization is biased toward regions with less curvature change.

Cost analysis

Table 4. Average wall-clock time per epoch (s) and the theoretical cost of different methods, with the lazy Hessian interval is set to 10.

Method	Cost				CIFAR10	CIFAR100	ImageNet
	Descent	Sharpness	Hessian	Total	ResNet32	WRN28-10	ViT-small
AdamW	1 GC	0 GC	0 HVP	1 GC	5.03	59.29	976.56
SAM	1 GC	1 GC	0 HVP	2 GC	9.16	118.46	1302.08
AdaHessian	1 GC	0 GC	1 HVP	4 GC	33.75	296.63	2489.07
SASSHA	1 GC	1 GC	0.1 HVP	2.3 GC	12.00	142.06	1377.20

- SASSHA is faster than other approximate second-order methods.