Accelerated Gradient Clipping

Optimization Methods in Machine Learning

Danil Andreev Vladimir Makharev

Innopolis University

Fall 2023

Gradient clipping

$$\text{clip}\left(\nabla f(x,\xi), \lambda\right) = \min\left\{1, \frac{\lambda}{||\nabla f(x,\xi)||}\right\} \nabla f(x,\xi)$$

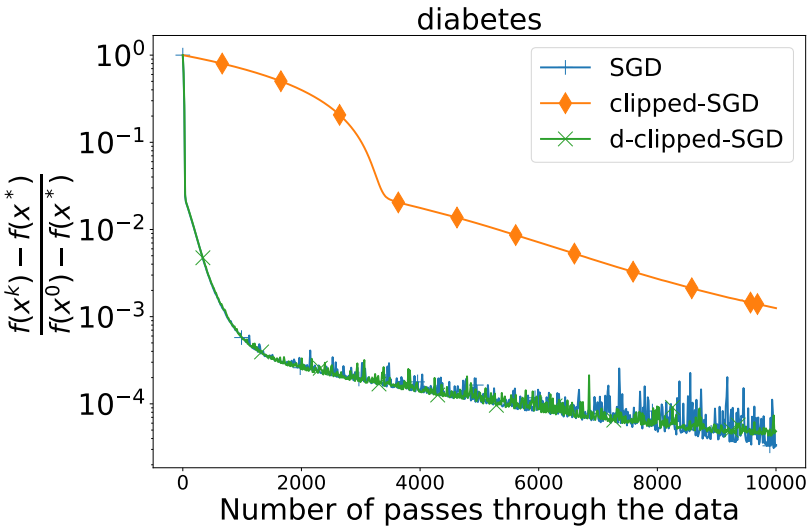
- $\nabla f(x,\xi)$ - stochastic gradient of the objective function at point x
- λ - clipping parameter, larger λ means less aggressive clipping
- $||\nabla f(x,\xi)||$ - Euclidean norm of the gradient vector

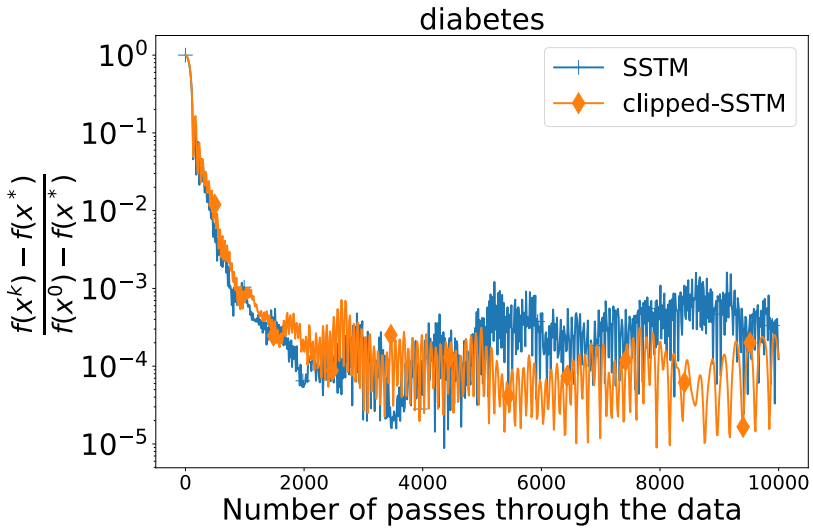
As a result, the gradient vector is projected on the Euclidean ball with radius λ with center at the origin.

Gradient clipping

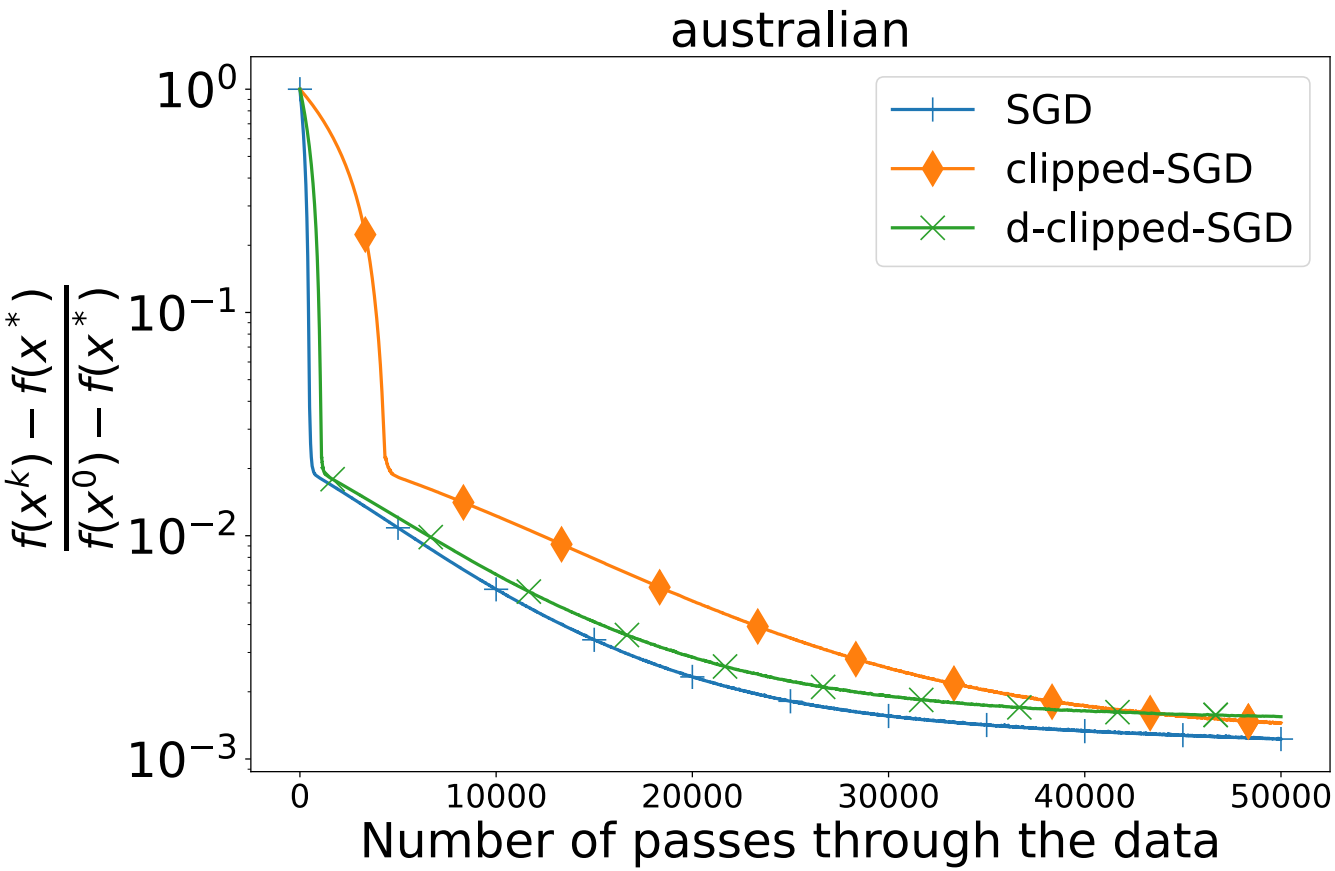
- In SGD, mini-batches are used for gradient computation.
- These mini-batches introduce randomness, leading to high-variance gradient estimates.
- Gradient clipping helps stabilize training by preventing extreme updates caused by these fluctuations.

SGD and SSTM trajectories on `diabetes` dataset

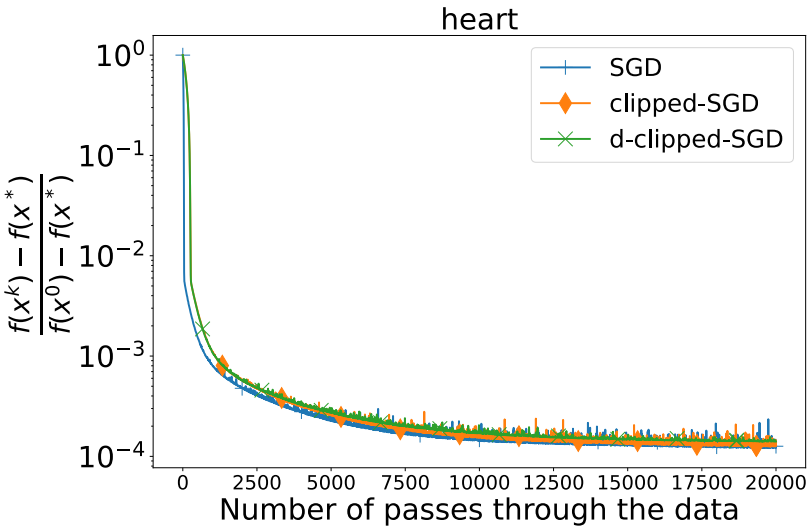


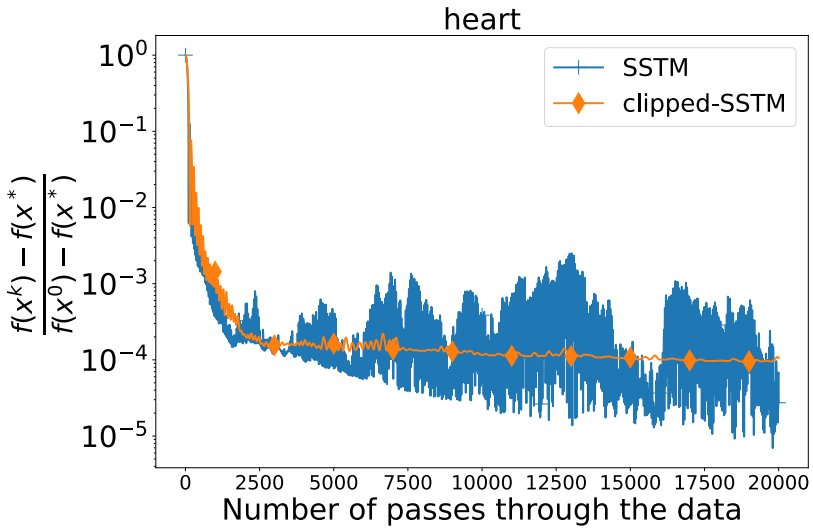


SGD trajectories on australian dataset



SGD and SSTM trajectories on heart dataset





Fine-tuning BERT on Yelp reviews

We built a pipeline to fine-tune a pretrained model based on HuggingFace docs

- Pretrained BERT (`bert-base-cased`) model
- Yelp reviews dataset of text reviews and score between 1 and 5
- Dataset splits: `(train, val, eval) = (1000, 300, 500)`

Fine-tuning BERT on Yelp reviews

We built a pipeline to fine-tune a pretrained model based on HuggingFace docs

- Pretrained BERT (`bert-base-cased`) model
- Yelp reviews dataset of text reviews and score between 1 and 5
- Dataset splits: `(train, val, eval) = (1000, 300, 500)`

Hyperparameters:

- `batch_size = 8`
- `num_epochs = 3`
- `learning_rate = 0.00005`

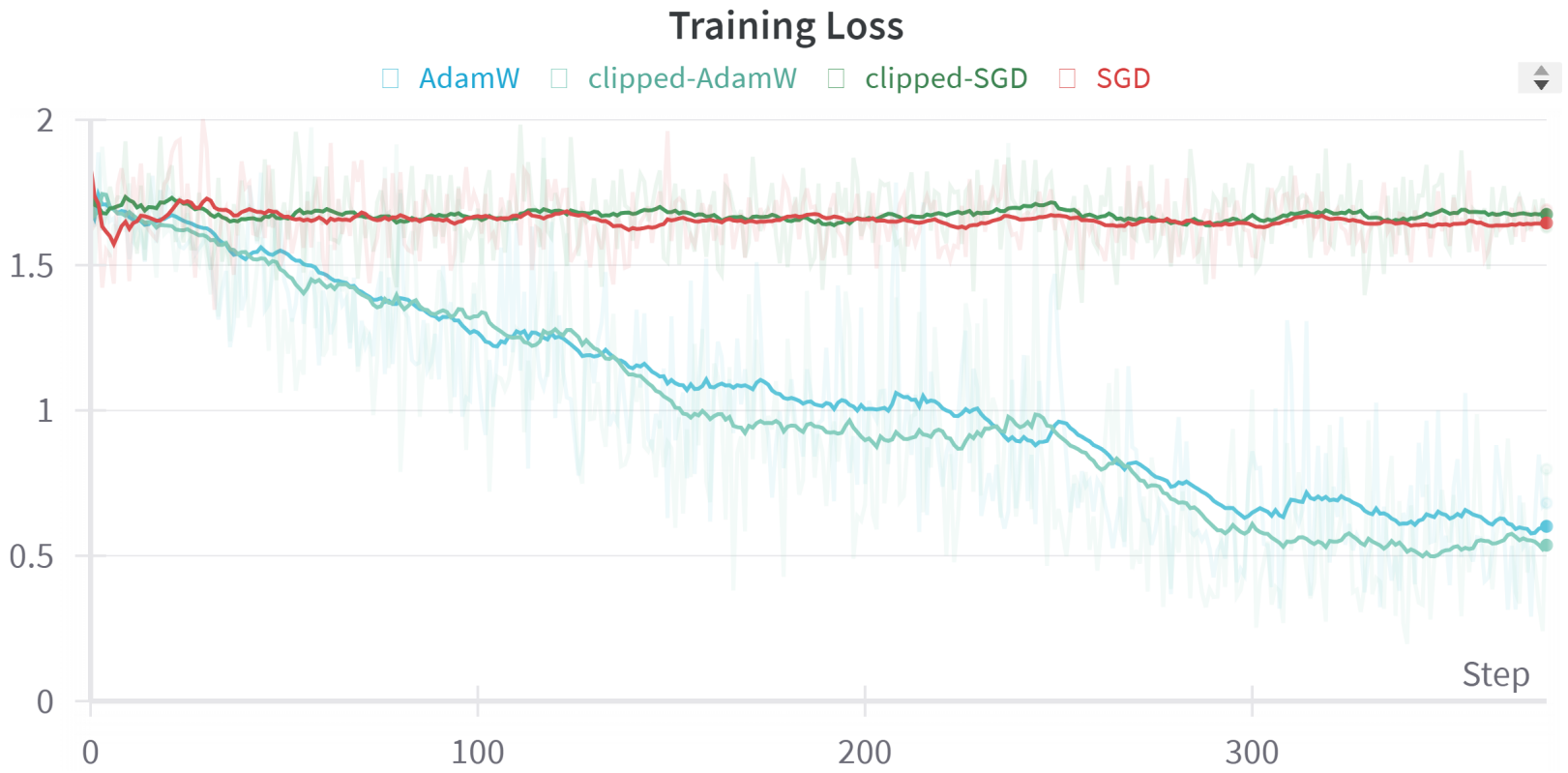
Experiments with BERT fine-tuning on optimizers

We experimented with PyTorch optimizers SGD and AdamW. For clipping, `clip_grad_norm_` function with `max_norm` $= \lambda = 1.0$ was used. We used WandD to track loss and accuracy. For loss `smoothing = 0.9` was applied. We denoted momentum as μ.

- `SGD`
- `clipped-SGD`
- `SGD` ($\mu = 0.9$)
- `clipped-SGD` ($\mu = 0.9$)
- `SGD-Nesterov` ($\mu = 0.9$)
- `clipped-SGD-Nesterov` ($\mu = 0.9$)
- `AdamW`
- `clipped-AdamW`

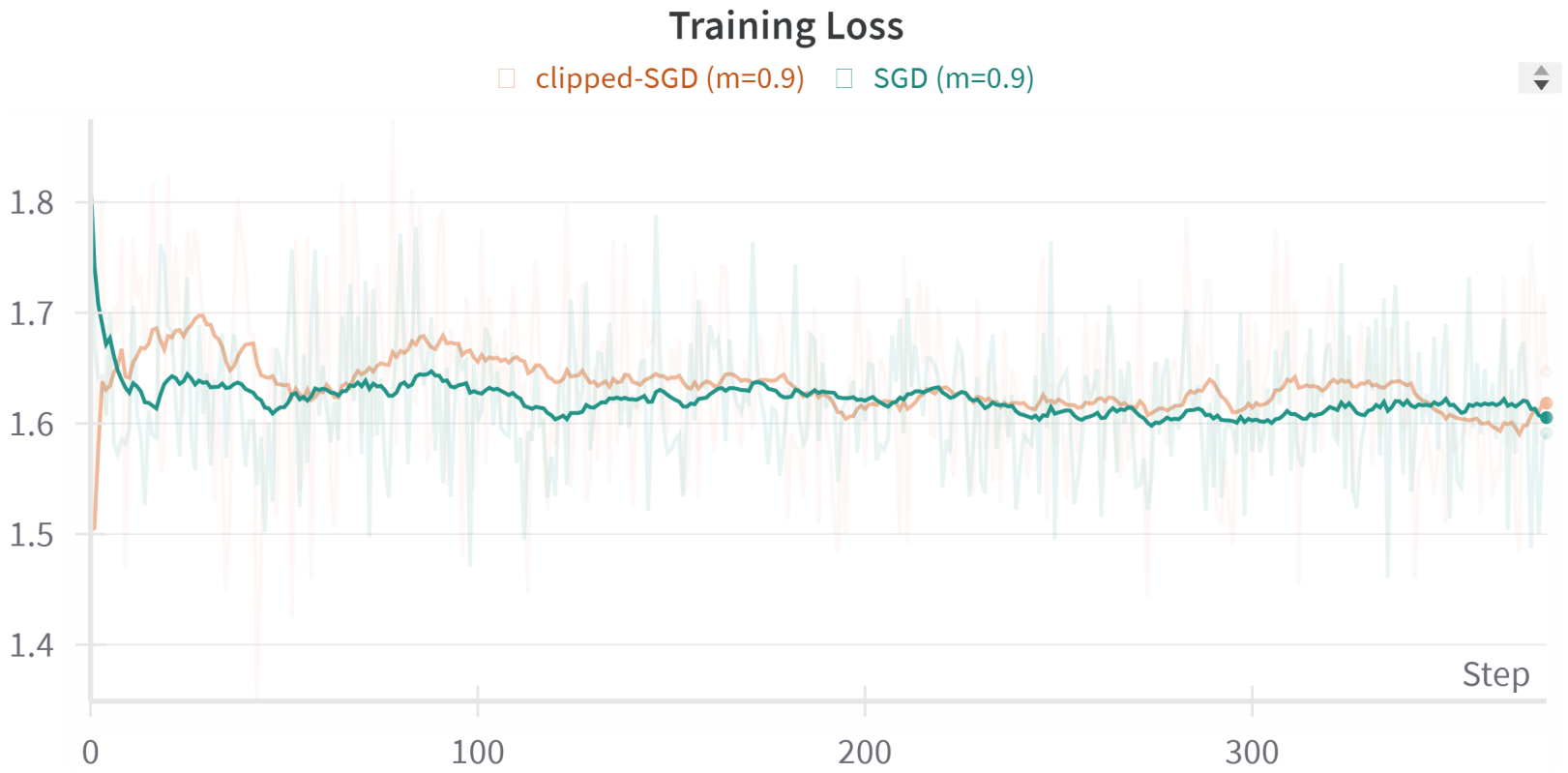
Training loss trajectories (`train` split)

`SGD`, `clipped-SGD`, `AdamW`, `clipped-AdamW`



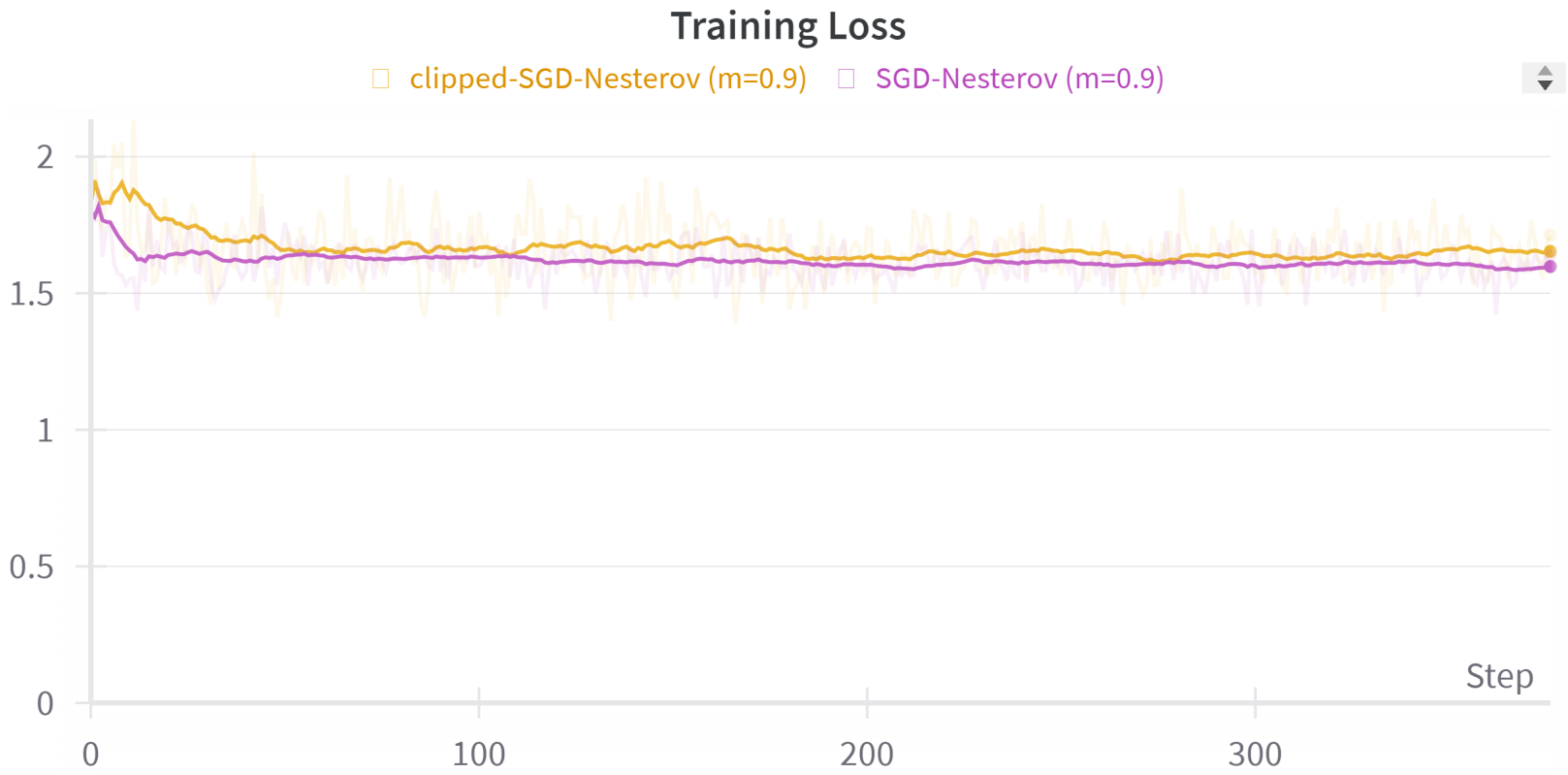
Training loss trajectories (`train` split)

With $\mu = 0.9$: `SGD`, `clipped-SGD`



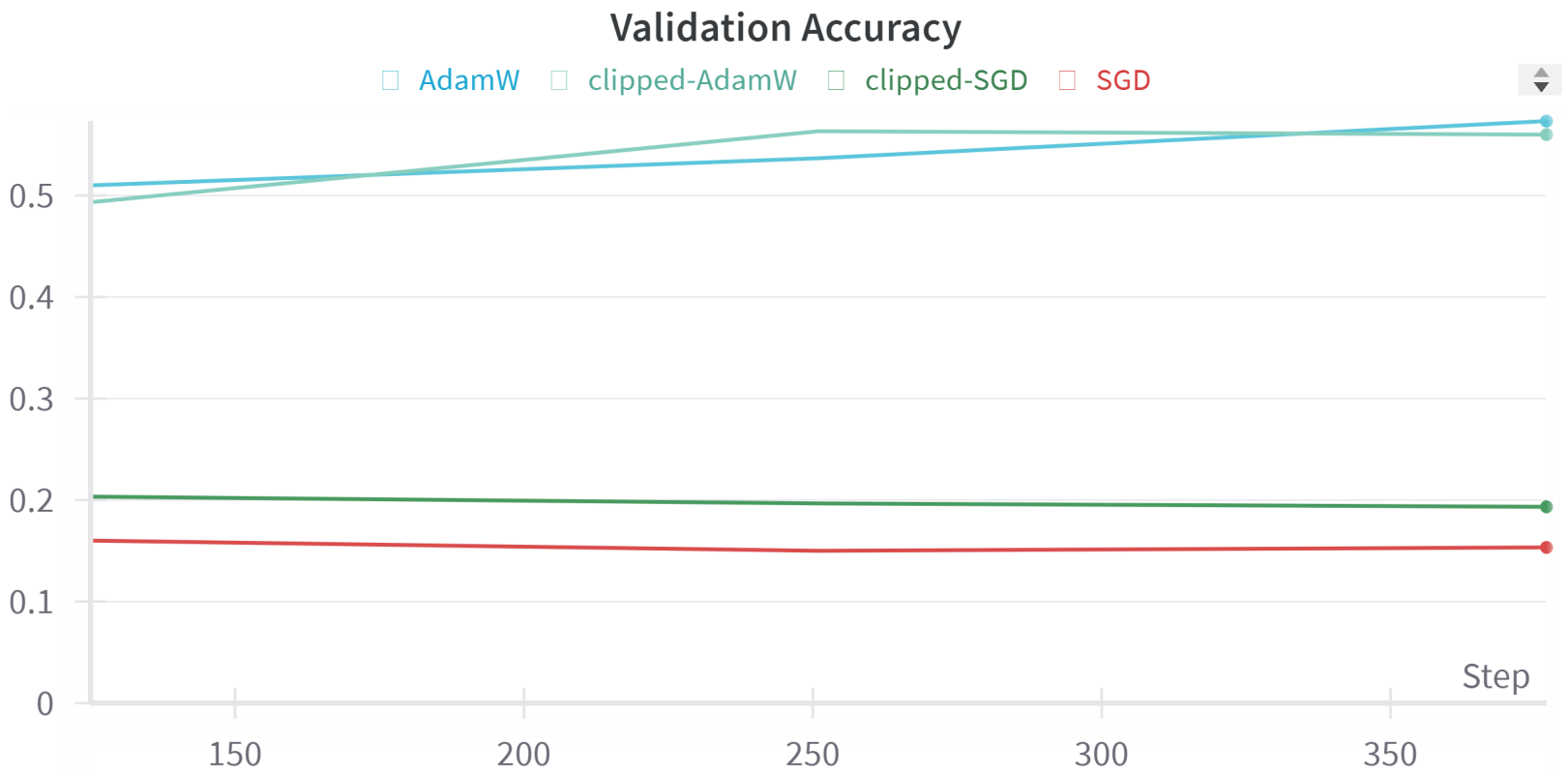
Training loss trajectories (`train` split)

With $\mu = 0.9$: `SGD-Nesterov`, `clipped-SGD-Nesterov`



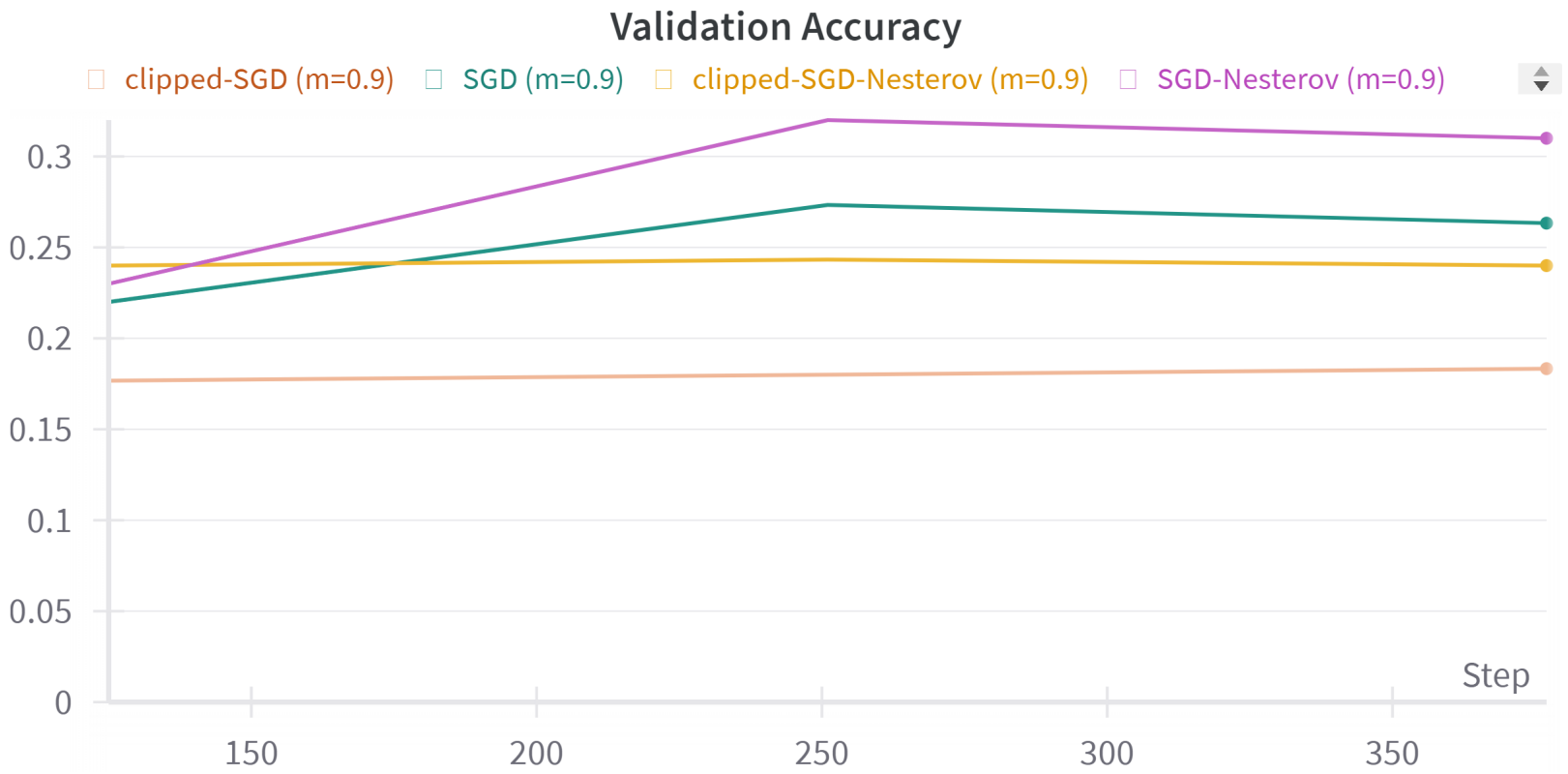
Validation accuracy trajectories (`val` split)

`SGD`, `clipped-SGD`, `AdamW`, `clipped-AdamW`

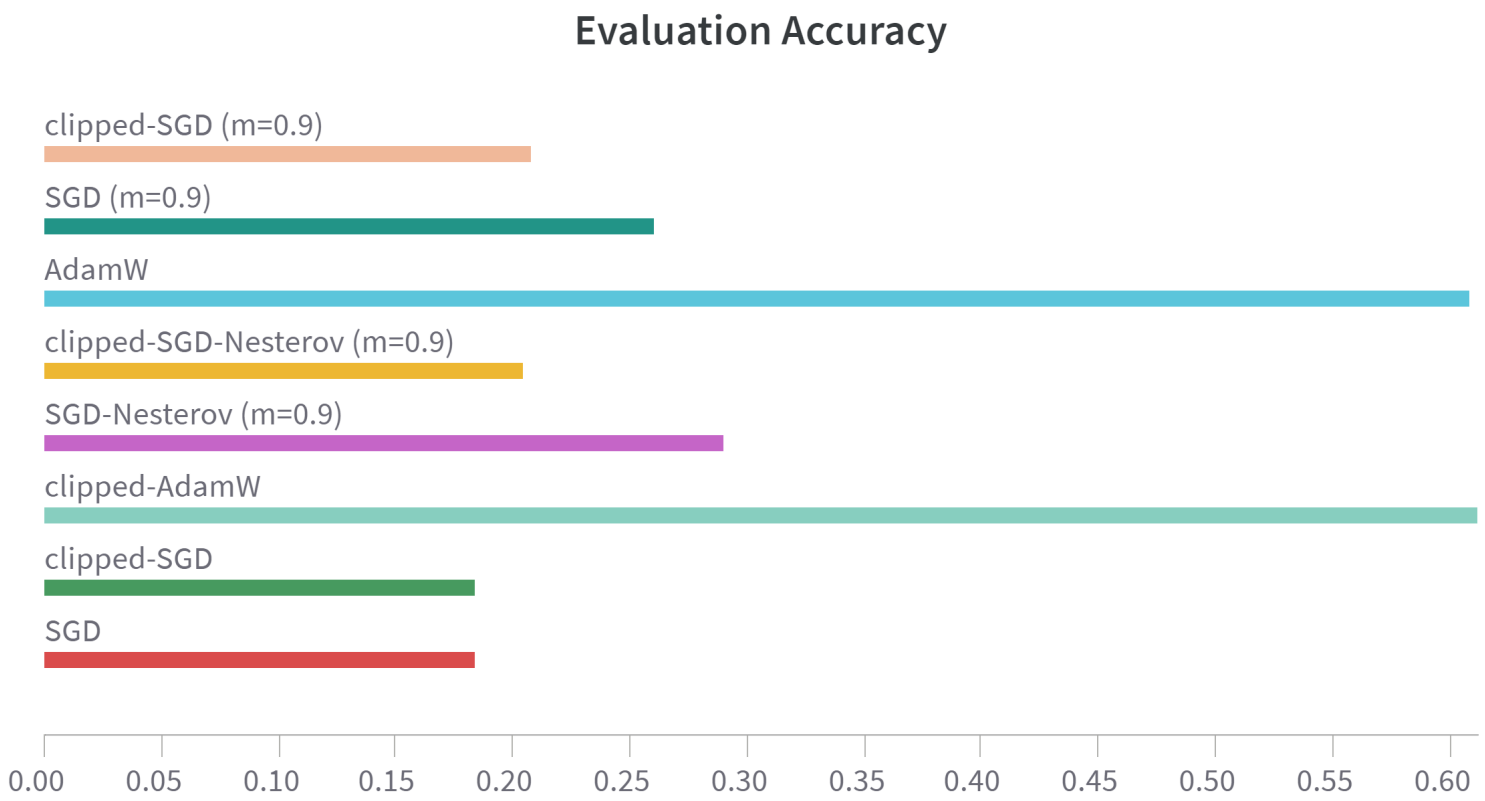


Validation accuracy trajectories (`val` split)

With $\mu = 0.9$: `SGD`, `clipped-SGD`, `SGD-Nesterov`, `clipped-SGD-Nesterov`



Evaluation accuracy trajectories (test split)



Conclusion

- Accelerated gradient clipping in stochastic optimization significantly enhances performance in the presence of heavy-tailed noise
- First high-probability complexity bounds (on a number of oracle calls) were derived by authors for `clipped-SSTM` and `clipped-SGD` methods
- Experiments from paper are easily reproducible
- Our experiments in NLP domain shows noticeable improvement in loss convergence for AdamW optimizer with clipping, while for SGD variants optimizers clipping effect is not significant

Preliminaries

In this section we introduce the main part of notations, assumption and definitions. The rest is classical for optimization literature and stated in the appendix (see Section A). Throughout the paper we assume that at each point $x \in \mathbb{R}^n$ function f is accessible only via stochastic gradients $\nabla f(x, \xi)$ such that

$$\mathbb{E}_\xi[\nabla f(x,\xi)] = \nabla f(x), \quad \mathbb{E}_\xi\left[\|\nabla f(x,\xi) - \nabla f(x)\|_2^2\right] \leq \sigma^2, \tag{2}$$

i.e. we have an access to the unbiased estimator of $\nabla f(x)$ with uniformly bounded by σ^2 variance where σ is some non-negative number. These assumptions on the stochastic gradient are standard in the stochastic optimization literature [18, 20, 31, 38, 49]. Below we introduce one of the most important definitions in this paper.

Definition 1.1 (light-tailed random vector)**.** We say that random vector η has a light-tailed distribution, i.e. satisfies "light-tails" assumption, if there exist $\mathbb{E}[\eta]$ and $\mathbb{P}\{\|\eta - \mathbb{E}[\eta]\|_2 > b\} \leq 2\exp\left(-\frac{b^2}{2\sigma^2}\right)$ for all $b > 0$

Such distributions are often called sub-Gaussian ones (see [30] and references therein). One can show (see Lemma 2 from [30]) that this definition is equivalent to

$$\mathbb{E}\left[\exp\left(\|\eta - \mathbb{E}[\eta]\|_2^2/\sigma^2\right)\right] \leq \exp(1) \tag{3}$$

up to absolute constant difference in σ. Due to Jensen's inequality and convexity of $\exp(\cdot)$ one can easily show that inequality (3) implies $\mathbb{E}[\|\eta - \mathbb{E}[\eta]\|_2^2] \leq \sigma^2$. However, the reverse implication does not hold in general. Therefore, in the rest of the paper by stochastic gradient with heavy-tailed distribution, we mean such a stochastic gradient that satisfies (2) but not necessarily (3).

Complexity bounds for convex objectives

Table 1: Comparison of existing high-probability convergence results for stochastic optimization under assumptions (2) for convex and L-smooth objectives. The second column contains an overall number of stochastic first-order oracle calls needed to achieve ε-solution with probability at least $1-\beta$. In the third column "light" means that $\nabla f(x,\xi)$ satisfies (3) and "heavy" means that the result holds even in the case when (3) does not hold. Column "Domain" describes the set where the optimization problem is defined. For RSMD Θ is a diameter of the set where the optimization problem is defined. We use red color to emphasize the restrictions we eliminate.

Method	Complexity	Tails	Domain
SGD [9]	$O\left(\max\left\{\frac{LR_0^2}{\varepsilon}, \frac{\sigma^2 R_0^2}{\varepsilon^2}\ln^2(\beta^{-1})\right\}\right)$	light	bounded
AC-SA [18, 38]	$O\left(\max\left\{\sqrt{\frac{LR_0^2}{\varepsilon}}, \frac{\sigma^2 R_0^2}{\varepsilon^2}\ln(\beta^{-1})\right\}\right)$	light	arbitrary
RSMD [47]	$O\left(\max\left\{\frac{L\Theta^2}{\varepsilon}, \frac{\sigma^2\Theta^2}{\varepsilon^2}\right\}\ln(\beta^{-1})\right)$	heavy	bounded
clipped-SGD [This work]	$O\left(\max\left\{\frac{LR_0^2}{\varepsilon}, \frac{\sigma^2 R_0^2}{\varepsilon^2}\right\}\ln(\beta^{-1})\right)$	heavy	$\mathbb{R}^n$
clipped-SSTM [This work]	$O\left(\max\left\{\sqrt{\frac{LR_0^2}{\varepsilon}}, \frac{\sigma^2 R_0^2}{\varepsilon^2}\right\}\ln\frac{LR_0^2+\sigma R_0}{\varepsilon\beta}\right)$	heavy	$\mathbb{R}^n$

Complexity bounds for μ-strongly convex objectives

Table 2: Comparison of existing high-probability convergence results for stochastic optimization under assumptions (2) for μ-strongly convex and L-smooth objectives. The second column contains an overall number of stochastic first-order oracle calls needed to achieve ε-solution with probability at least $1-\beta$. In the third column "light" means that $\nabla f(x,\xi)$ satisfies (3) and "heavy" means that the result holds even in the case when (3) does not hold. Column "Domain" describes the set where the optimization problem is defined. For RSMD Θ is a diameter of the set where the optimization problem is defined and $R = \sqrt{2(f(x^0)-f(x^*))/\mu}$, $r_0 = f(x^0) - f(x^*)$. We use red color to emphasize the restrictions we eliminate.

Method	Complexity	Tails	Domain
SIGM [11]	$O\left(\max\left\{\frac{L}{\mu}\ln\frac{\mu R_0^2}{\varepsilon}, \frac{\sigma^2}{\mu\varepsilon}\ln\left(\beta^{-1}\ln\frac{\mu R_0^2}{\varepsilon}\right)\right\}\right)$	light	arbitrary
MS-AC-SA [19]	$O\left(\max\left\{\sqrt{\frac{L}{\mu}}\ln\frac{LR_0^2}{\varepsilon}, \frac{\sigma^2}{\mu\varepsilon}\ln\left(\beta^{-1}\ln\frac{LR_0^2}{\varepsilon}\right)\right\}\right)$	light	arbitrary
restarted-RSMD [47]	$O\left(\max\left\{\frac{L}{\mu}\ln\left(\frac{\mu\Theta^2}{\varepsilon}\right), \frac{\sigma^2}{\mu\varepsilon}\right\}\ln\left(\beta^{-1}\ln\frac{\mu\Theta^2}{\varepsilon}\right)\right)$	heavy	bounded
proxBoost [7]	$O\left(\max\left\{\sqrt{\frac{L}{\mu}}\ln\left(\frac{LR_0^2\ln\frac{L}{\mu}}{\varepsilon}\right), \frac{\sigma^2\ln\frac{L}{\mu}}{\mu\varepsilon}\right\}\cdot C\right)$, where $C = \ln\left(\frac{L}{\mu}\right)\ln\left(\frac{\ln\frac{L}{\mu}}{\beta}\right)$	heavy	arbitrary
clipped-SGD [This work]	$O\left(\max\left\{\frac{L}{\mu}, \frac{\sigma^2}{\mu\varepsilon}\cdot\frac{L}{\mu}\right\}\ln\left(\frac{r_0}{\varepsilon}\right)\ln\left(\frac{L}{\mu\beta}\ln\frac{r_0}{\varepsilon}\right)\right)$	heavy	$\mathbb{R}^n$
R-clipped-SGD [This work]	$O\left(\max\left\{\frac{L}{\mu}\ln\frac{\mu R^2}{\varepsilon}, \frac{\sigma^2}{\mu\varepsilon}\right\}\ln\left(\frac{L}{\mu\beta}\ln\frac{\mu R^2}{\varepsilon}\right)\right)$	heavy	$\mathbb{R}^n$
R-clipped-SSTM [This work]	$O\left(\max\left\{\sqrt{\frac{L}{\mu}}\ln\frac{\mu R^2}{\varepsilon}, \frac{\sigma^2}{\mu\varepsilon}\right\}\ln\left(\frac{L}{\mu\beta}\ln\frac{\mu R^2}{\varepsilon}\right)\right)$	heavy	$\mathbb{R}^n$