OPT2025: 17th Annual Workshop on Optimization for Machine Learning 1–32

Aligning Distributionally Robust Optimization with Practical Deep Learning Needs

Dmitrii Feoktistov[1,2,3] FEOKTISTOVDD@MY.MSU.RU
Igor Ignashin[2,4] IGNASHIN.I@MIRIAI.ORG
Andrey Veprikov[2,4,5] VEPRIKOV.A@MIRIAI.ORG
Nikita Borovko[1] BOROVKONA@MY.MSU.RU
Alexander Bogdanov[2,4] BOGDANOV.A@MIRIAI.ORG
Savelii Chezhegov[2,4,5] SAVACHEZHEGOV2017@GMAIL.COM
Aleksandr Beznosikov[2,4] ANBEZNOSIKOV@GMAIL.COM
[1]*Lomonosov Moscow State University, Moscow, Russia*
[2]*Basic Research of Artificial Intelligence Laboratory (BRAIn Lab), Moscow, Russia*
[3]*Yandex Research, Moscow, Russia*
[4]*Moscow Independent Research Institute of Artificial Intelligence (MIRAI), Moscow, Russia*
[5]*SB AI Lab, Moscow, Russia*

Abstract

While traditional Deep Learning (DL) optimization methods treat all training samples equally, Distributionally Robust Optimization (DRO) adaptively assigns importance weights to different samples. However, a significant gap exists between DRO and current DL practices. Modern DL optimizers require adaptivity and the ability to handle stochastic gradients, as these methods demonstrate superior performance. This paper aims to bridge this gap by introducing ALSO – Adaptive Loss Scaling Optimizer – an adaptive DRO algorithm suitable for DL. We prove the convergence of our proposed algorithm for non-convex objectives, the standard setting for DL models. Empirical evaluation demonstrates that ALSO outperforms baselines.

1. Introduction

Deep Learning (DL) has long been centered around the empirical risk minimization problem:

$$\min_{\theta\in\mathbb{R}^d}\left\{\frac{1}{n}\sum_{i=1}^{n} f_i(\theta)+\frac{\tau}{2}\|\theta\|_2^2\right\}, \tag{1}$$

where θ are the parameters of the DL model, $f_i(\theta) := L(\text{model}(\theta,\mathbf{x}_i),\ \mathbf{y}_i)$ is the loss function on the i-th element $(\mathbf{x}_i,\mathbf{y}_i)\in\mathbf{X}\times\mathbf{Y}$ of the training data, n is the number of the training samples and $\frac{\tau}{2}\|\theta\|_2^2$ is a regularizer.

However, the problem (1) has a natural limitation: it assumes that all samples in the training dataset are equally important. Additionally, DL models performance is measured on test samples, which are not used in (1). This uniform treatment can lead to poor generalization, especially when a distributional shift exists between the training and test sets – a common cause of overfitting [43]. Distributionally Robust Optimization (DRO) [11; 27; 42] minimizes the expected loss with respect to the "worst" training data distribution at the moment. One formalization of this idea leads to the following minimax problem:

$$\min_{\theta \in \mathbb{R}^d} \max_{\pi \in \Delta_{n-1} \cap U} \left\{ h(\theta, \pi) := \sum_{i=1}^{n} \pi_i f_i(\theta) + \frac{\tau}{2} \|\theta\|_2^2 - \tau \mathrm{KL}\left[\pi \| \hat{\pi}\right] \right\}, \tag{2}$$

where U is an uncertainty set, i.e. constraint on π. For example, one can use KL-divergence ball to prevent significant deviations from some prior distribution $\hat{\pi}$: $U = \{\pi \in \Delta_{n-1} : \mathrm{KL}\left[\pi \| \hat{\pi}\right] \leq r\}$. To additionally restrict deviation from the starting distribution, a regularization using KL-divergence is used, where $\tau > 0$ is the regularization parameter. It is worth highlighting that if we substitute $\pi = \hat{\pi} = \mathcal{U}\left(\overline{1, n}\right)$ into (2), the resulting equation is exactly the same as (1).

Despite that DRO has successful applications in separate DL fields such as Reinforcement Learning [31; 21; 30] and Semi-Supervised Learning [3], we identify several challenges in applying existing methods for general DL:

• **Non-adaptive θ-update**. Most general DRO methods use simple SGD updates [5] or apply Variance Reduction (VR) techniques [33; 32; 25; 38], while the most successful DL optimizers are adaptive [22; 8].

• **Focus on methods for classical Machine Learning.** Despite the success of the existing DRO methods in the convex domain (e.g. logistic regression) [33; 32; 25], neural networks are inherently non-convex, presenting additional challenges, and VR methods are usually ineffective in DL [10]. Furthermore, Variance Reduction techniques necessitate additional memory to store historical gradients – a significant limitation for large Deep Learning models with millions of parameters.

• **Heuristic methods.** Several attempts have been made to develop DRO methods specifically for Deep Learning. For instance, in [29] the authors propose a heuristic algorithm without theoretical guarantees that requires two separate training phases to produce the final model. An alternative approach is presented in [40], where the authors propose an algorithm with convergence guarantees for the convex case and apply it to neural network training with replacement of GD step with Adam step without proper theoretical analysis.

• **Exact solution of the inner maximization problem in** (2). Another approach is proposed by [38], where authors address the non-convex scenario. They solve the inner maximization problem exactly, resulting in the following formulation:

$$\min_{\theta \in \mathbb{R}^d} \left\{ \frac{1}{n} \sum_{i=1}^{n} \exp\left[\tau^{-1} f_i(\theta)\right] + \frac{\tau}{2} \|\theta\|_2^2 \right\}. \tag{3}$$

Although this eliminates the need to manage π, the reformulation (3) has several drawbacks. The exponential term is numerically unstable for small τ. Moreover, computing the globally optimal $\pi^*(\theta)$ for an undertrained model can be problematic, as it can cause the model to overfit outliers and prematurely focus on samples that are only difficult due to the model's immaturity.

Motivated by these limitations, we design general-purpose DRO methods for the problem (2), which utilize adaptive θ-update, and analyze it in the non-convex, stochastic case. Our contributions are summarized as follows.

• **Deep Learning optimizer.** We present `ALSO` – `A`daptive `L`oss `S`caling `O`ptimizer – a novel algorithm designed to solve the problem (2) in Deep Learning context (Algorithm 1).

• **Theory.** We establish a convergence of `ALSO` in the stochastic, non-convex, L-smooth case.

• **Experiments**. We experimentally demonstrate that `ALSO` outperforms `Adam` [22] and DRO algorithms.

2. ALSO – Adaptive Loss Scaling Optimizer

The development of our algorithm is motivated by the evolution of optimization methods for saddle point problems. The easiest option to obtain methods for saddle point problems is to adapt gradient schemes from minimization tasks. In this way, it is possible to obtain the Stochastic Gradient Descent Ascent (SGDA) method. However, this scheme is inadequate from the theoretical perspective. Therefore, it is suggested to use more advanced algorithms such as Extragradient [23]. For our non-Euclidean geometry, it makes sense to consider an appropriate modification of Extragradient – Mirror-Prox [19]. However, both Extragradient and Mirror-Prox require two oracle calls per iteration. To address this, so-called Optimistic version of these algorithms can be applied [37]. It requires only one oracle call per iteration. It turns out that the Extragradient and Optimistic updates outperform SGDA not only in the theory, but also in DL, particularly in training GANs [9; 12; 34; 6; 26; 36]. Building upon the foundation of Optimistic Mirror-Prox, we introduce `ALSO` (Algorithm 1) – the `A`daptive `L`oss `S`caling `O`ptimizer which effectively addresses DL requirements. Since for θ euclidean norm is used, Optimistic Mirror-Prox utilizes GD-like step over θ. To enhance adaptivity, we replace this GD step with Adam [22], resulting in our proposed `ALSO` algorithm (Algorithm 1) for solving the problem (2).

In practice, nearly all works that employ Euclidean Optimistic method for DL tasks do not use its theoretical version, but rather an adaptive variant (typically with Adam-style stepsizes) [9; 12; 34; 6; 26; 36]. This substitution is often justified as a standard procedure in DL. However, we question this approach, as establishing theoretical guarantees for adaptive methods is a nontrivial and technically demanding task (see Appendix C). In this work, we do not follow this simplified route; instead, we provide a rigorous analysis of the adaptive method (see Theorem 5).

Algorithm 1 `ALSO`

Input: $\gamma_\theta, \gamma_\pi$ – stepsize for θ and π; β_1, β_2, ε from Adam; momentum α; τ_π, τ_θ – regularization parameters for π and θ; number of iterations N; $\hat{\pi}$ – prior distribution.

Initialization: $m^0 = g^0 = p^0 = \mathbf{0}$, $v_0 = 0$, $\pi^0 = \hat{\pi}$, $\hat{\gamma}_\pi = \gamma_\pi/(1 + \gamma_\pi \tau_\pi)$, $n = \mathrm{len}(\hat{\pi})$

for $k = 0, 1, 2, \ldots, N$ **do**

Sample B indexes of object: $\{i_1^k, ..., i_B^k\}$

$g^{k+1} = \frac{n}{B} \sum_{j=1}^{B} \pi_{i_j^k} \nabla_\theta f_{i_j^k}(\theta^k)$

$\hat{g}^{k+1} = (1+\alpha) g^{k+1} - \alpha g^k + \tau_\theta \theta^k$

$p^{k+1} = \frac{n}{B} \sum_{j=1}^{B} e_{i_j^k} \cdot f_{i_j^k}(\theta^k)$, where e_i is vector with 1 in i-th position and zeros in others

$\hat{p}^{k+1} = (1+\alpha) p^{k+1} - \alpha p^k$

$\theta^{k+1} = \theta^k - \gamma_\theta \cdot \mathrm{Adam}(\hat{g}^{k+1}, \beta_1, \beta_2, \varepsilon)$

$\pi^{k+1} = \arg\min_{\pi \in U \cap \Delta_{c-1}} \left\{ \langle \hat{p}^{k+1} + \log \frac{\pi}{\hat{\pi}}, \pi \rangle + \mathrm{KL}[\pi || \pi^k] \right\}$

end

Assumption 1 *The admissible domain $\mathcal{D}_\pi := \Delta_{c-1} \cap U$ is nonempty, closed, and convex. Moreover, regularizer $\hat{\pi} \in \mathrm{Int}(\mathcal{D}_\pi)$*

Assumption 2 *For all i the functions f_i from (2) are K_i-Lipschitz and L_i-Lipschitz continuous on Θ with respect to the Euclidean norm $\|\cdot\|_2$, i.e., for any $\theta^1, \theta^2 \in \Theta$ the following inequality holds:*

$$\|\nabla f_i(\theta^1) - \nabla f_i(\theta^2)\|_2 \leq L_i \|\theta^1 - \theta^2\|_2 \quad \text{and} \quad |f_i(\theta^1) - f_i(\theta^2)|_2 \leq K_i \|\theta^1 - \theta^2\|_2.$$

Assumption 3 *At each iteration of Algorithm 1 we have access to $g = g(\theta, \pi)$ and $p = p(\theta, \pi)$, which provide unbiased estimates of the gradients for the problem (2). Moreover,*

$$\mathbb{E}\big\|g(\theta,\pi) - \nabla_\theta h(\theta,\pi)\big\|_2^2 \leq \sigma^2; \quad \mathbb{E}\big\|p(\theta,\pi) - \nabla_\pi h(\theta,\pi)\big\|_2^2 \leq \sigma^2.$$

Definition 4 (Stationary point, cf. [28]) *A point θ is called an ε-stationary point ($\varepsilon \geq 0$) of a differentiable function Φ if $\|\nabla \Phi(\theta)\| \leq \varepsilon$. If $\varepsilon = 0$, then θ is a stationary point.*

In our setting, the primal objective is $\Phi(\theta) := \max_{\pi \in \mathcal{D}_\pi} h(\theta, \pi)$, which is differentiable since $h(\theta, \pi)$ is smooth with respect to θ and the maximization is over a compact convex set. Therefore, following [28], it is sufficient to measure convergence of Algorithm 1 by the gradient norm $\|\nabla \Phi(\theta)\|$, as small gradients certify approximate stationarity of the original min–max problem (2). Moreover, due to stochasticity in the updates, it is natural to adopt the criterion $\mathbb{E}\|\nabla \Phi(\theta)\|^2 \leq \varepsilon^2$.

Now we are ready to present the following main theorem, which establishes the complexity bounds of Algorithm 1.

Theorem 5 *Under Assumptions 1, 2, 3, the required number of iterations to achieve ε-stationarity 4 ($\mathbb{E}\|\nabla \Phi(\theta)\|^2 \leq \varepsilon^2$) for the problem (2) by `ALSO` (Algorithm 1) with $\gamma_\theta = \mathcal{O}(\frac{\lambda^4}{L^4})$, $\gamma_\pi = \frac{\lambda}{8L^2}$, $\beta_1 = \mathcal{O}(\frac{\varepsilon \lambda^2}{L^2})$, $\beta_2 = 1 - \mathcal{O}(\varepsilon^2)$, $B = \mathcal{O}(\frac{\sigma^2}{\varepsilon^2})$ is*

$$T = \mathcal{O}\left(\frac{L^4}{\lambda^4 \varepsilon^2} \cdot \max\{\Delta_\Phi \cdot (K + \sigma),\ D_0\}\right),$$

where $\Delta_\Phi = \Phi(\theta^0) - \min_{\theta \in \mathbb{R}^d} \Phi(\theta)$, $D_0 = KL(\pi^*(\theta^0) \| \pi^0), \pi^*(\theta) = \arg\max_{\pi \in \mathcal{D}_\pi} h(\theta, \pi)$ and $L^2 = \mathcal{O}\left(\left(\frac{c}{n} \max_i \sum_{j=1}^{n_i} L_{i,j} + \tau + \frac{c}{n} \max_i \sum_{j=1}^{n_i} K_{i,j}\right)^2 + \lambda^2\right)$, $K = \frac{c}{n} \max_i \sum_{j=1}^{n_i} K_{i,j}$.

Appendix C provides a detailed derivation of the constants and discusses parameter tuning.

Discussion. This convergence result matches the convergence guarantees established for the standard SGDA method in [28]. Unlike SGDA, our method leverages a non-Euclidean geometry and incorporates adaptivity by performing an Adam-type update on the θ variable. At the same time, the accurate decomposition of heavy-ball terms and boundaries for the scaled factors, similarly to [7] for the nonconvex scenario, allows to obtain the GD-behavior in our analysis, which leads to the same theoretical guarantees as for SGDA.

3. Experiments

We compare `ALSO` with standard DL baselines, including `AdamW` and `Static Weights` (see Appendix A.1 for details), as well as DRO methods that tackle close to our problem. We use both classical DRO methods like `Spectral Risk` [32], and state-of-the-art methods such as `DRAGO` [33] (noted for fast convergence), `FastDRO` [25] (a scalable method), `RECOVER` [38] (a non-convex method). Baselines were implemented using official code when available, or based on the original papers otherwise. We use established metrics that effectively capture performance under heterogeneity. All methods were tuned for the same number of iterations using the Optuna package [1] (see Appendix A). To reduce the hyperparameter search space, we fix $\alpha = 1$. This decision is supported by theory (see [37]) and prior empirical studies, which have shown that setting α near 1 is an effective choice [34; 9; 2]. We use $U = \Delta_{n-1}$ for `ALSO`. Our code is available at https://github.com/brain-lab-research/ALSO.

3.1. Learning from Unbalanced Data

This experiment demonstrates `ALSO`'s effectiveness on training datasets with significant class imbalance. We consider a classification task on the CIFAR-10 dataset [24] using the ResNet-18 model [17]. Class imbalance was created by grouping the 10 original classes into two by parity. One of these groups was then undersampled in the training and validation sets, while the test set remained balanced for evaluation. To quantify the class imbalance, we introduce the unbalanced coefficient (uc), which specifies the ratio of samples between the first and second classes as: $\#\text{ 1 class}/\#\text{ 2 class} = \text{uc}$, where # is the number of samples in the corresponding classes. We consider the values $\text{uc} \in \{1, 2, 5, 10, 20, 30, 40, 50\}$. The results of the experiment are presented in Figure 1. Analyzing the results, we observe that the proposed method `ALSO` outperforms all the compared baselines. The performance advantage is particularly noticeable for large values of the unbalanced coefficient (≥ 30), where one class significantly outweighs the other. Additional details can be found in Appendix A.1.

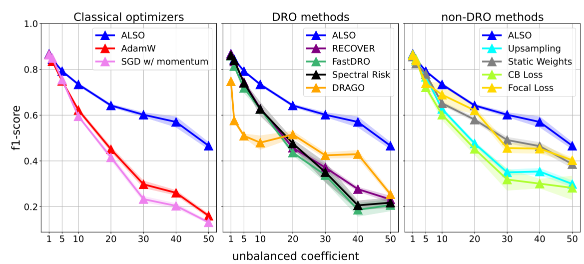

Figure 1: Performance comparison of optimization techniques designed for training in the presence of class imbalance: `ALSO`, different `AdamW` and DRO techniques. The final f1-score was averaged over 20 runs, see Appendix A.1 for details.

3.2. Tabular Deep Learning

We choose Tabular DL for `ALSO` evaluation as tabular data is not only central to many real-world, industrial problems but is also characterized by complex and challenging data heterogeneity: heavy-tailed, non-symmetric target, extreme distributional shift, class imbalance, e.t.c. (see Table 3 for details). As a model, we choose MLP-PLR [13] as it is a strong baseline in the tabular DL field. We evaluate the training procedure over 14 datasets from [15; 39]. Detailed dataset characteristics and training specifications can be found in Appendix A.2. The results of the algorithms comparison are presented in Table 1. `ALSO` demonstrates the best performance on the most datasets and can be considered as alternative to both conventional DL optimizers and specialized DRO methods.

Table 1: Performance comparison of `ALSO` and baselines on tabular deep learning datasets. Bold entries represent the best method on each dataset according to mean, underlined entries represent methods, which performance is best with standard deviations over 15 runs. Metric is written near dataset name, ↑ means that higher values indicate better performance, ↓ means otherwise.

Dataset	ALSO	AdamW	DRAGO	Spectral Risk	FastDRO	RECOVER	Static Weights
Weather (RMSE ↓)	**1.4928 ± 0.0042**	1.5208 ± 0.0037	1.5803 ± 0.0103	1.5189 ± 0.0047	1.5184 ± 0.0041	1.5547 ± 0.0034	1.5161 ± 0.0046
Ecom Offers (ROC-AUC ↑)	0.5976 ± 0.0020	0.5810 ± 0.0039	**0.5983 ± 0.0019**	0.5796 ± 0.0034	0.5900 ± 0.0126	0.5859 ± 0.0031	0.5803 ± 0.0033
Cooking Time (RMSE ↓)	**0.4806 ± 0.0003**	0.4813 ± 0.0003	0.4843 ± 0.0008	0.4810 ± 0.0004	0.4809 ± 0.0004	0.4813 ± 0.0006	0.4818 ± 0.0006
Maps Routing (RMSE ↓)	**0.1612 ± 0.0001**	0.1618 ± 0.0002	0.1651 ± 0.0005	0.1619 ± 0.0003	0.1620 ± 0.0003	0.1621 ± 0.0003	0.1617 ± 0.0002
Homesite Insurance (ROC-AUC ↑)	**0.9632 ± 0.0003**	0.9621 ± 0.0005	0.9536 ± 0.0018	0.9609 ± 0.0005	0.9614 ± 0.0008	0.9612 ± 0.0005	0.9619 ± 0.0003
Delivery ETA (RMSE ↓)	**0.5513 ± 0.0020**	0.5519 ± 0.0017	0.5555 ± 0.0016	0.5528 ± 0.0013	0.5528 ± 0.0017	0.5551 ± 0.0035	0.5555 ± 0.0031
Homecredit Default (ROC-AUC ↑)	**0.8585 ± 0.0012**	0.8579 ± 0.0012	0.8463 ± 0.0013	0.8575 ± 0.0012	0.8579 ± 0.0014	0.8576 ± 0.0011	0.8557 ± 0.0012
Sberbank Housing (RMSE ↓)	**0.2424 ± 0.0024**	0.2434 ± 0.0027	0.2694 ± 0.0070	0.2453 ± 0.0036	0.2458 ± 0.0044	0.2589 ± 0.0093	0.2465 ± 0.0080
Black Friday (RMSE ↓)	**0.6842 ± 0.0004**	0.6864 ± 0.0005	0.7011 ± 0.0040	0.6861 ± 0.0004	0.6861 ± 0.0003	0.6963 ± 0.0012	0.6870 ± 0.0008
Microsoft (RMSE ↓)	**0.7437 ± 0.0004**	0.7442 ± 0.0003	0.7496 ± 0.0010	0.7441 ± 0.0003	0.7448 ± 0.0004	0.7486 ± 0.0002	0.7467 ± 0.0004
California Housing (RMSE ↓)	**0.4495 ± 0.0046**	0.4602 ± 0.0042	0.6326 ± 0.2073	0.4681 ± 0.0050	0.4639 ± 0.0024	0.4787 ± 0.0042	0.4651 ± 0.0040
Churn Modeling (ROC-AUC ↑)	**0.8666 ± 0.0027**	0.8616 ± 0.0015	0.7960 ± 0.0010	0.8626 ± 0.0020	0.8622 ± 0.0020	0.8604 ± 0.0033	0.8249 ± 0.0073
Adult (ROC-AUC ↑)	0.8699 ± 0.0001	0.8688 ± 0.0012	0.7640 ± 0.0014	0.8687 ± 0.0009	**0.8702 ± 0.0009**	0.8683 ± 0.0013	0.8498 ± 0.0051
Higgs Small (ROC-AUC ↑)	0.7280 ± 0.0009	0.7274 ± 0.0017	0.6263 ± 0.0573	**0.7282 ± 0.0021**	**0.7282 ± 0.0009**	0.7267 ± 0.0013	0.7222 ± 0.0022

4. Ablation Study Summary

Due to space constraints, our full ablation studies are presented in Appendix B. This section provides a summary of the key findings: we demonstrate that `ALSO`'s computational overhead is insignificant compared to training with `AdamW` (Section B.1); `ALSO` demonstrates stable performance across a wide range of hyperparameter values (Section B.2); we validate our key design choices, such as the use of momentum (α) and a non-adaptive π update, showing that the version of `ALSO` presented in Algorithm 1 is optimal (Section B.3).

5. Discussion and Future work

In this work, we introduce `ALSO`, an adaptive optimizer that successfully bridges the gap between Distributionally Robust Optimization and the practical needs of modern deep learning. Our theoretical analysis establishes convergence in the challenging non-convex setting, and extensive empirical evaluations demonstrate its superior performance. For future work, we would like to explore alternative adaptive mechanisms and methods for automatically tuning the regularization parameters could further enhance the algorithm's usability and performance.

6. Acknowledgements

The work was supported by the Ministry of Economic Development of the Russian Federation (agreement No. 139-15-2025-013, dated June 20, 2025, IGK 000000C313925P4B0002).

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] Kimon Antonakopoulos, Veronica Belmega, and Panayotis Mertikopoulos. An adaptive mirror-prox method for variational inequalities with singular operators. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Jose Blanchet and Yang Kang. Semi-supervised learning based on distributionally robust optimization. *Data Analysis and Applications 3: Computational, Classification, Financial, Statistical and Stochastic Methods*, 5:1–33, 2020.

[4] Dmitry Bylinkin, Mikhail Aleksandrov, Savelii Chezhegov, and Aleksandr Beznosikov. Enhancing stability of physics-informed neural network training through saddle-point reformulation, 2025. URL https://arxiv.org/abs/2507.16008.

[5] Yair Carmon and Danielle Hausler. Distributionally robust optimization via ball oracle acceleration. *Advances in Neural Information Processing Systems*, 35:35866–35879, 2022.

[6] Tatjana Chavdarova, Gauthier Gidel, François Fleuret, and Simon Lacoste-Julien. Reducing noise in gan training with variance reduced extragradient. *Advances in Neural Information Processing Systems*, 32, 2019.

[7] Savelii Chezhegov, Yaroslav Klyukin, Andrei Semenov, Aleksandr Beznosikov, Alexander Gasnikov, Samuel Horváth, Martin Takáč, and Eduard Gorbunov. Gradient clipping improves adagrad when the noise is heavy-tailed. *arXiv preprint arXiv:2406.04443*, 2024.

[8] Dami Choi, Christopher J Shallue, Zachary Nado, Jaehoon Lee, Chris J Maddison, and George E Dahl. On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446*, 2019.

[9] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

[10] Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[11] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

[12] Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. *arXiv preprint arXiv:1802.10551*, 2018.

[13] Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35:24991–25004, 2022.

[14] Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. Tabm: Advancing tabular deep learning with parameter-efficient ensembling. *arXiv preprint arXiv:2410.24210*, 2024.

[15] Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. Tabr: Tabular deep learning meets nearest neighbors. In *The Twelfth International Conference on Learning Representations*, 2024.

[16] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Zixuan Jiang, Jiaqi Gu, Mingjie Liu, Keren Zhu, and David Z Pan. Optimizer fusion: Efficient training with better locality and parallelism. *arXiv preprint arXiv:2104.00237*, 2021.

[19] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[20] Herman Kahn and Andy W Marshall. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.

[21] Nathan Kallus, Xiaojie Mao, Kaiwen Wang, and Zhengyuan Zhou. Doubly robust distributionally robust off-policy evaluation and learning. In *International Conference on Machine Learning*, pages 10598–10632. PMLR, 2022.

[22] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[25] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33:8847–8860, 2020.

[26] Tengyuan Liang and James Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 907–915. PMLR, 2019.

[27] Fengming Lin, Xiaolei Fang, and Zheming Gao. Distributionally robust optimization: A review on theory and applications. *Numerical Algebra, Control and Optimization*, 12(1):159–212, 2022.

[28] Tianyi Lin, Chi Jin, and Michael Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International conference on machine learning*, pages 6083–6093. PMLR, 2020.

[29] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

[30] Zijian Liu, Qinxun Bai, Jose Blanchet, Perry Dong, Wei Xu, Zhengqing Zhou, and Zhengyuan Zhou. Distributionally robust q-learning. In *International Conference on Machine Learning*, pages 13623–13643. PMLR, 2022.

[31] Kyriakos Lotidis, Nicholas Bambos, Jose Blanchet, and Jiajin Li. Wasserstein distributionally robust linear-quadratic estimation under martingale constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 8629–8644. PMLR, 2023.

[32] Ronak Mehta, Vincent Roulet, Krishna Pillutla, Lang Liu, and Zaid Harchaoui. Stochastic optimization for spectral risk measures. In *International Conference on Artificial Intelligence and Statistics*, pages 10112–10159. PMLR, 2023.

[33] Ronak Mehta, Jelena Diakonikolas, and Zaid Harchaoui. Drago: Primal-dual coupled variance reduction for faster distributionally robust optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[34] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. *arXiv preprint arXiv:1807.02629*, 2018.

[35] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[36] Wei Peng, Yu-Hong Dai, Hui Zhang, and Lizhi Cheng. Training gans with centripetal acceleration. *Optimization Methods and Software*, 35(5):955–973, 2020.

[37] Leonid Denisovich Popov. A modification of the arrow-hurwitz method of search for saddle points. *Mat. Zametki*, 28(5):777–784, 1980.

[38] Qi Qi, Zhishuai Guo, Yi Xu, Rong Jin, and Tianbao Yang. An online method for a class of distributionally robust optimization with non-convex objectives. *Advances in Neural Information Processing Systems*, 34:10067–10080, 2021.

[39] Ivan Rubachev, Nikolay Kartashev, Yury Gorishniy, and Artem Babenko. Tabred: Analyzing pitfalls and filling the gaps in tabular deep learning benchmarks. *arXiv preprint arXiv:2406.19380*, 2024.

[40] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[41] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Data augmentation using random image cropping and patching for deep cnns. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(9):2917–2931, 2019.

[42] Wolfram Wiesemann. Distributionally robust optimization. *arXiv preprint arXiv:2411.02549*, 2024.

[43] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pages 7184–7193. PMLR, 2019.

Appendix A. Missing Experiment Details

A.1. Unbalanced Data Details (Section 3.1)

Baselines description. Now, let us discuss described basic unbalance handling techniques. The first of these techniques is known as *upsampling* [16; 20], the idea is to sample objects for gradient calculation at the current optimization step not uniformly, but proportionally to the class ratio of each object in the training dataset. For the $\hat{\pi}$ regularizer in the problem (2), we utilize this modified distribution instead of the vanilla uniform distribution $\mathcal{U}(\overline{1,n})$. This choice results in a significant improvement in the performance. The second technique is called *static weights* [16]. Its idea is similar to the previous method, however, instead of modifying the sampling distribution, objects are sampled uniformly. The class imbalance is then addressed by multiplying the loss function for each object by a weight equal to the inverse ratio of the number of objects belonging to that class in the training dataset.

Data preprocessing. For all optimizers the same preprocessing was used for fair comparison. We modified the images from CIFAR-10 train dataset with Normalizing and classical computer vision augmentations: Random Crop [41], Random Horizontally Flip.

Training neural networks. We use cross-entropy as the loss function. We do not apply learning rate schedules since we tune hyperparameters. We use a predefined batch size equal to 64 and maximum number of epochs equal to 20.

Hyperparameter tuning. Hyperparameter tuning is performed with the TPE sampler (200 iterations) with 5 epoch from the Optuna package [1]. Hyperparameter tuning spaces for experiment are provided in Table 2.

Parameter	Distribution
Learning rate	LogUniform$[1e\text{-}4, 1e\text{-}2]$
Weight decay	LogUniform$[1e\text{-}6, 1e\text{-}2]$
π-Learning rate (γ_π from `ALSO`, used for `ALSO`, `DRAGO`)	LogUniform$[1e\text{-}5, 1e\text{-}3]$
π-regularization (τ_π from `ALSO`, used for `ALSO`, `DRAGO`, `RECOVER`, `Spectral Risk`)	LogUniform$[1e\text{-}3, 1]$

Table 2: The hyperparameter tuning space for unbalanced data experiment.

Evaluation. The tuned hyperparameters are evaluated under 20 random seeds. The mean test metric and its standard deviation over these random seeds are then used to compare algorithms as described in Section 3.1.

A.2. Tabular Deep Learning Details (Section 3.2)

Name	# Train	# Validation	# Test	# Num	# Bin	# Cat	Task type	Metric	Heterogeniety	Batch size
Sberbank Housing	18 847	4 827	4 647	365	17	10	Regression	RMSE	Heavy-tailed	256
Ecom Offers	109 341	24 261	26 455	113	6	0	Binclass	ROC AUC	Extreme shift	1024
Maps Routing	160 019	59 975	59 951	984	0	2	Regression	RMSE	-	1024
Homesite Insurance	224 320	20 138	16 295	253	23	23	Binclass	ROC AUC	Class imbalance	1024
Cooking Time	227 087	51 251	41 648	186	3	3	Regression	RMSE	Heavy-tailed	1024
Homecredit Default	267 645	58 018	56 001	612	2	82	Binclass	ROC AUC	High uncertainty	1024
Delivery ETA	279 415	34 174	36 927	221	1	1	Regression	RMSE	Non-symmetric	1024
Weather	106 764	42 359	40 840	100	3	0	Regression	RMSE	Non-symmetric	1024
Churn Modelling	6 400	1 600	2 000	10	3	1	Binclass	ROC AUC	Noisy data	128
California Housing	13 209	3 303	4 128	8	0	0	Regression	RMSE	Heavy-tailed	256
Adult	26 048	6 513	16 281	6	1	8	Binclass	ROC AUC	High uncertainty	256
Higgs Small	62 751	15 688	19 610	28	0	0	Binclass	ROC AUC	-	512
Black Friday	106 764	26 692	33 365	4	1	4	Regression	RMSE	Heavy-tailed	512
Microsoft	723 412	235 259	241 521	131	5	0	Regression	RMSE	-	1024

Table 3: Properties of the datasets from [15; 39]. "# Num", "# Bin", and "# Cat" denote the number of numerical, binary, and categorical features, respectively

We mostly follow the experiment setup from [14]. As such, most of the text below is copied from [14].

Data preprocessing. For each dataset, for all optimizers, the same preprocessing was used for fair comparison. For numerical features, by default, we used a slightly modified version of the quantile normalization from the Scikit-learn package [35] (see the source code), with rare exceptions when it turned out to be detrimental (for such datasets, we used the standard normalization or no normalization). For categorical features, we used one-hot encoding. Binary features (i.e. the ones that take only two distinct values) are mapped to $\{0, 1\}$ without any further preprocessing.

Training neural networks. We use cross-entropy for classification problems and mean squared error for regression problems as loss function. We do not apply learning rate schedules. We do not use data augmentations. We apply global gradient clipping to 1.0. For each dataset, we used a predefined dataset-specific batch size. We continue training until there are `patience` consecutive epochs without improvements on the validation set; we set `patience` $= 16$.

Hyperparameter tuning. In most cases, hyperparameter tuning is performed with the TPE sampler (100 iterations) from the Optuna package [1]. Hyperparameter tuning spaces for experiment are provided in Table 4.

Evaluation. On a given dataset, for a given model, the tuned hyperparameters are evaluated under multiple (in most cases, 15) random seeds. The mean test metric and its standard deviation over these random seeds are then used to compare algorithms as described in Table 3.

Parameter	Distribution
# layers	UniformInt$[1, 5]$
Width (hidden size)	UniformInt$[64, 1024]$
Dropout rate	$\{0.0, \text{Uniform}[0.0, 0.5]\}$
n_frequencies	UniformInt$[16, 96]$
d_embedding	UniformInt$[16, 32]$
frequency_init_scale	LogUniform$[1e\text{-}2, 1e1]$
Learning rate	LogUniform$[3e\text{-}5, 1e\text{-}3]$
Weight decay	$\{0, \text{LogUniform}[1e\text{-}4, 1e\text{-}1]\}$
π-Learning rate (γ_π from `ALSO`, used for `ALSO`, `DRAGO`)	LogUniform$[1e\text{-}5, 1e\text{-}3]$
π-regularization (τ_π from `ALSO`, used for `ALSO`, `DRAGO`, `RECOVER`, `Spectral Risk`)	LogUniform$[1e\text{-}3, 1]$
Size (used for `FastDRO`)	Uniform$[0, 1]$
n_draws (used for `Spectral Risk`)	LogUniform$[1e\text{-}3, 1]$

Table 4: The hyperparameter tuning space for tabular Deep Learning experiment.

Appendix B. Ablation Study

B.1. `ALSO` Step Time Analysis

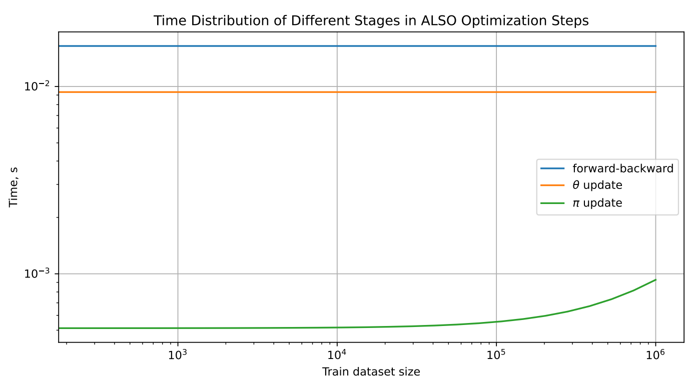

Figure 2: Time distribution over dataset size of three main parts of optimization process with `ALSO`: gradient computation (forward-backward), θ update and π update. The trained model is ResNet-18 with batch size. Time of each part is averaged across 25 training steps. We want to highlight, that gradient computations are required for all first order optimization methods, and this measurement is used only for comparison.

To analyze the time consumption of each component in the optimization process with `ALSO`, we conduct an experiment training ResNet-18 [17] with a fixed batch size of 64 across various dataset sizes, measured time is averaged across 25 iterations. This approach is chosen because while π updates depend on dataset size, gradient computation and θ updates do not. We test dataset sizes up to 1 million samples, which exceeds our largest experimental dataset, which contains approximately 800000 samples. The experiment was conducted on one NVIDIA GeForce RTX 2080 Ti GPU. We want to highlight, that gradient computations are required for all first order optimization methods, and this measurement is used only for comparison.

The results, presented in Figure 2, reveal a clear hierarchy in computational demands. Gradient computation (forward-backward passes) consistently requires significantly more time than both θ and π updates across all dataset sizes, which is consistent with [18]. Furthermore, θ updates consistently demand more computational time than π updates. This experiment leads to conclusion that the explicit weight vector update (π update) is computationally negligible relative to the overall training step time.

B.2. Hyperparameters sensitivity

This ablation study examines `ALSO`'s sensitivity to its π-specific hyperparameters: the π-learning rate (γ_π) and π-regularization (τ_π). We conducted full 2D sweeps for both parameters, fixing model weight learning rates and regularization to isolate their impact. Results from the imbalanced data setting (Section 3.1) show consistent performance across varying imbalance coefficients (Figure 3). Across all experiments, `ALSO` proves largely insensitive to γ_π and τ_π settings, suggesting strong performance is achievable without extensive tuning.

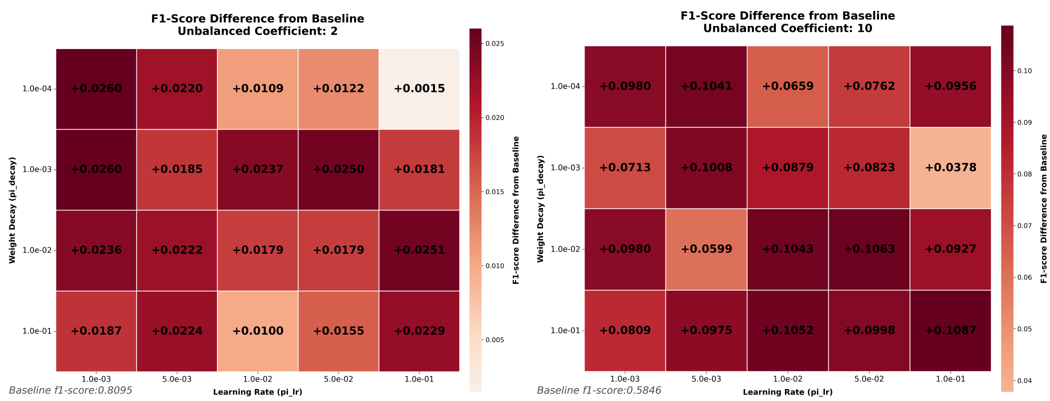

Figure 3: Robustness of `ALSO` to π-hyperparameters (ΔF1-score vs. AdamW baseline). Each cell shows the F1-score difference between `ALSO` and AdamW with static weights (baseline), over a full 2D grid of π-learning rate (γ_π) and π-regularization (τ_π). All cells are red (positive ΔF1), indicating that `ALSO` consistently outperforms the baseline across the entire grid and for different imbalance coefficients (2 and 10).

B.3. Design choices

This section presents an empirical evaluation of key design choices in the proposed algorithm, focusing on the optimistic step and the non-adaptive update rule for the parameter π. We compare the performance of three algorithm variants:

1. Vanilla `ALSO`: The standard implementation of the proposed algorithm (Algorithm 1).

2. Descent-Ascent `ALSO` ($\alpha = 0$): A variant where the optimistic step is removed by setting the optimistic coefficient α to zero.

3. $\mathtt{A}^{\pi}\mathtt{LSO}$: A modified version of `ALSO` that employs the Adam optimizer for updating the weight vector π.

The algorithms were evaluated across three distinct experimental settings: Learning from Unbalanced Data (Section 3.1), Tabular Deep Learning (Section 3.2). The results are summarized in Figure 4 and Table 5.

The Descent-Ascent variant has a significantly lower performance compared to the other two algorithms, indicating the importance of the optimistic step. The $\mathtt{A}^{\pi}\mathtt{LSO}$ algorithm achieves comparable performance to vanilla `ALSO` in some scenarios (Table 5). However, in the Unbalanced Data experiment, $\mathtt{A}^{\pi}\mathtt{LSO}$ demonstrates degraded performance when the unbalanced coefficient is large (≥ 10).

Considering both performance and ease of implementation, we recommend vanilla `ALSO` as a robust baseline. While $\mathtt{A}^{\pi}\mathtt{LSO}$ can provide competitive results in certain settings, it introduces additional hyperparameters and computational overhead associated with the Adam optimizer for π. Therefore, $\mathtt{A}^{\pi}\mathtt{LSO}$ may be considered when sufficient computational resources are available for hyperparameter tuning and multiple experimental runs.

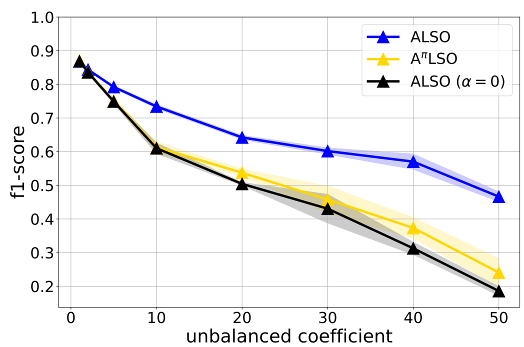

Figure 4: Performance comparison of `ALSO`, `ALSO` with $\alpha = 0$ (descent-ascent), and $\mathtt{A}^{\pi}\mathtt{LSO}$ (adaptive step over π) on the unbalanced CIFAR experiment from Section 3.1. Hyperparameter tuning is performed in the same manner as in the main experiment.

Appendix C. Theory for ALSO

C.1. Definitions

Let $h(\theta, \pi)$ be a differentiable function defined in 2. In our analysis, we will consider Assumptions 2, 3, and 1 to provide theoretical guarantees.

In fact, we apply 3 to estimate the norms of stochastic gradients and we add batch size B to control the variance of noise that occurs due to stochastics in gradient oracle. Also in 2 we require the $K_{i,j}$-Lipschitz continuity of $f_{i,j}(\theta)$ and their $L_{i,j}$-smoothness. In the sequel, assumption 7 is useful several times in calculations, but it has a different form, however, we can estimate this constant L through our existing $L_{i,j}$ and $K_{i,j}$.

We use asssumption 1 with set U because this notation is adopted in the related paper [33]. Namely, we define the domain for π as the set $U \cap \Delta$, which is usually used to truncate

Dataset	`ALSO`	`ALSO` $\alpha = 0$	`A`$^{\pi}$`LSO`
Weather (RMSE ↓)	$\mathbf{1.4928 \pm 0.0042}$	1.5209 ± 0.0036	$\underline{1.4967 \pm 0.0066}$
Ecom Offers (ROC-AUC ↑)	$\mathbf{0.5976 \pm 0.0020}$	$\underline{0.5975 \pm 0.0020}$	0.5915 ± 0.0087
Cooking Time (RMSE ↓)	$\mathbf{0.4806 \pm 0.0003}$	0.4810 ± 0.0003	$\mathbf{0.4806 \pm 0.0004}$
Maps Routing (RMSE ↓)	$\underline{0.1612 \pm 0.0001}$	0.1613 ± 0.0002	$\mathbf{0.1611 \pm 0.0001}$
Homesite Insurance (ROC-AUC ↑)	$\mathbf{0.9632 \pm 0.0003}$	$\underline{0.9630 \pm 0.0004}$	0.9626 ± 0.0003
Delivery ETA (RMSE ↓)	$\underline{0.5513 \pm 0.0020}$	0.5536 ± 0.0030	$\mathbf{0.5507 \pm 0.0011}$
Homecredit Default (ROC-AUC ↑)	$\mathbf{0.8587 \pm 0.0012}$	0.8587 ± 0.0008	$\mathbf{0.8587 \pm 0.0011}$
Sberbank Housing (RMSE ↓)	$\underline{0.2424 \pm 0.0024}$	0.2457 ± 0.0044	$\mathbf{0.2401 \pm 0.0073}$
Black Friday (RMSE ↓)	$\underline{0.6842 \pm 0.0004}$	$\underline{0.6843 \pm 0.0013}$	$\mathbf{0.6838 \pm 0.0005}$
Microsoft (RMSE ↓)	$\underline{0.7437 \pm 0.0003}$	$\underline{0.7435 \pm 0.0003}$	$\mathbf{0.7438 \pm 0.0003}$
California Housing (RMSE ↓)	0.4495 ± 0.0046	0.4533 ± 0.0043	$\mathbf{0.4455 \pm 0.0032}$
Churn Modeling (ROC-AUC ↑)	$\mathbf{0.8666 \pm 0.0027}$	0.8597 ± 0.0076	$\underline{0.8646 \pm 0.0019}$
Adult (ROC-AUC ↑)	$\mathbf{0.8699 \pm 0.0001}$	$\underline{0.8698 \pm 0.0002}$	$\underline{0.8698 \pm 0.0014}$
Higgs Small (ROC-AUC ↑)	$\underline{0.7280 \pm 0.0009}$	$\underline{0.7279 \pm 0.0013}$	$\mathbf{0.7288 \pm 0.0012}$

Table 5: Performance comparison of `ALSO`, `ALSO` $\alpha = 0$ (descent-ascent) and `A`$^{\pi}$`LSO` (adaptive step over π). The trained model is MLP-PLR [13]. Bold entries represent the best method on each dataset according to mean, underlined entries represent methods, which performance is best with standard deviations over 15 seeds taken into account. Metric is written near dataset name, ↑ means that higher values indicate better performance, ↓ means that lower values indicate better performance. Hyperparameter tuning is performed in the same manner as in the main experiment.

corners of Δ to ensure that the KL divergence remains bounded on $\Delta \cap U$. However, in our theory we do not require that the simplex must be with truncated corners.

In this section, we consider a more general case of assumptions for our algorithm. So we now introduce several definitions and lemmas proven in [4], which will be used in the convergence analysis.

We consider more general problem than (2):

$$\min_{\theta \in \mathbb{R}^d} \max_{\pi \in S} \left[\mathcal{L}(\theta, \pi) = \sum_{i=1}^{c} \pi_i \left(\frac{c}{n} \sum_{j=1}^{n_i} f_{i,j}(\theta) \right) + \frac{\tau}{2} \|\theta\|_2^2 - \lambda D_{\psi}(\pi || \hat{\pi}) \right], \tag{4}$$

where we replace KL-divergence with general D_{Ψ}-divergence (Bregman divergence).

Assumption 6 *The domain $S \subseteq \mathbb{R}^c$ is nonempty, closed, convex, with $\hat{\pi} \in \operatorname{Int}(S)$.*

Assumption 7 *The function $\mathcal{L}(\theta, \pi)$ is L-smooth, i.e. for all $(\theta_1, \pi_1), (\theta_2, \pi_2) \in \mathbb{R}^d \times S$ it satisfies*

$$\|\nabla \mathcal{L}(\theta_1, \pi_1) - \nabla \mathcal{L}(\theta_2, \pi_2)\|^2 \leq L^2 \left(\|\theta_1 - \theta_2\|^2 + \|\pi_1 - \pi_2\|^2 \right).$$

Lemma 8 *Under Assumptions 2, and 6, the function $\mathcal{L}(\theta, \pi)$ in (4) is L-smooth (i.e. Assumption 7), i.e. for all $(\theta^1, \pi^1), (\theta^2, \pi^2) \in \mathbb{R}^d \times S$ it holds*

$$\|\nabla \mathcal{L}(\theta^1, \pi^1) - \nabla \mathcal{L}(\theta^2, \pi^2)\|^2 \leq L^2 \Big(\|\theta^1 - \theta^2\|^2 + \|\pi^1 - \pi^2\|^2 \Big),$$

where the Lipschitz constant L can be chosen as

$$L^2 \;=\; \Big(\tfrac{c}{n}\max_{i\in[c]}\sum_{j=1}^{n_i} L_{i,j} + \tau + \tfrac{c}{n}\max_{i\in[c]}\sum_{j=1}^{n_i} K_{i,j}\Big)^2 + (\lambda L_\psi)^2,$$

with $L_{i,j}$ and $K_{i,j}$ being the smoothness and Lipschitz constants of $f_{i,j}$ from Assumption 2, and L_ψ the Lipschitz constant of $\nabla_\pi D_\psi(\cdot\|\hat{\pi})$.

Proof We decompose the gradient into its θ- and π-parts:

$$\nabla_\theta \mathcal{L}(\theta,\pi) = \sum_{i=1}^{c}\pi_i\Big(\tfrac{c}{n}\sum_{j=1}^{n_i}\nabla f_{i,j}(\theta)\Big) + \tau\theta, \quad \nabla_\pi \mathcal{L}(\theta,\pi) = \Big(\tfrac{c}{n}\sum_{j=1}^{n_i} f_{i,j}(\theta)\Big)_{i=1}^{c} - \lambda\nabla_\pi D_\psi(\pi\|\hat{\pi}).$$

For the θ-part we obtain

$$\begin{aligned}
&\|\nabla_\theta \mathcal{L}(\theta^1,\pi^1) - \nabla_\theta \mathcal{L}(\theta^2,\pi^2)\| \\
&\le \sum_{i=1}^{c}|\pi_i^1 - \pi_i^2|\Big(\tfrac{c}{n}\sum_{j=1}^{n_i}\|\nabla f_{i,j}(\theta^1)\|\Big) + \tfrac{c}{n}\sum_{i=1}^{c}\pi_i^2\sum_{j=1}^{n_i}\|\nabla f_{i,j}(\theta^1) - \nabla f_{i,j}(\theta^2)\| + \tau\|\theta^1 - \theta^2\| \\
&\le \tfrac{c}{n}\max_i\sum_{j=1}^{n_i} K_{i,j}\,\|\pi^1 - \pi^2\| + \Big(\tfrac{c}{n}\max_i\sum_{j=1}^{n_i} L_{i,j} + \tau\Big)\|\theta^1 - \theta^2\|.
\end{aligned}$$

For the π-part we analogously have

$$\|\nabla_\pi \mathcal{L}(\theta^1,\pi^1) - \nabla_\pi \mathcal{L}(\theta^2,\pi^2)\| \le \tfrac{c}{n}\max_i\sum_{j=1}^{n_i} K_{i,j}\,\|\theta^1 - \theta^2\| + \lambda L_\psi\|\pi^1 - \pi^2\|.$$

Combining both estimates yields

$$\|\nabla\mathcal{L}(\theta^1,\pi^1) - \nabla\mathcal{L}(\theta^2,\pi^2)\|^2 \;\le\; \Big(\tfrac{c}{n}\max_i\sum_j L_{i,j} + \tau + \tfrac{c}{n}\max_i\sum_j K_{i,j}\Big)^2\|\theta^1 - \theta^2\|^2 + (\lambda L_\psi)^2\|\pi^1 - \pi^2\|^2,$$

which completes the proof. ∎

Lemma 9 *Under Assumption 2, with $\tau = 0$, the function $\mathcal{L}(\theta,\pi)$ in (4) is K-lipschitz with respect to θ, i.e. for all $\theta^1,\theta^2 \in \mathbb{R}^d$ and $\pi \in S$ it holds*

$$|\mathcal{L}(\theta^1,\pi) - \mathcal{L}(\theta^2,\pi)| \;\le\; L\|\theta^1 - \theta^2\|,$$

where the K can be chosen as

$$K \;=\; \frac{c}{n}\max_{i\in[c]}\sum_{j=1}^{n_i} K_{ij}$$

with $K_{i,j}$ being Lipschitz constant of $f_{i,j}$ from Assumption 2.

Proof

$$|\mathcal{L}(\theta^1, \pi) - \mathcal{L}(\theta^2, \pi)| = |\sum_{i=1}^{c} \pi_i \frac{c}{n} \sum_{j=1}^{n_i} (f_{ij}(\theta^1) - f_{ij}(\theta^2))| \leq$$

$$\sum_{i=1}^{c} \pi_i \frac{c}{n} \sum_{i=1}^{n} |f_{ij}(\theta^1) - f_{ij}(\theta^2)| \leq \sum_{i=1}^{c} \pi_i \frac{c}{n} \sum_{j=1}^{n_i} K_{ij} \|\theta^1 - \theta^2\| \leq$$

$$\leq \frac{c}{n} \|\theta^1 - \theta^2\| \sum_{i=1}^{c} \pi_i \sum_{j=1}^{n_i} K_{ij} \leq \frac{c}{n} \max_{i \in [c]} \sum_{j=1}^{n_i} K_{ij}$$

The last inequality holds, since $\pi \in \Delta_{c-1}$. ■

Assumption 10 *The function ψ, which produce D_ψ, is **1-strongly convex**, i.e. for all $\pi_1, \pi_2 \in S$ it satisfies*

$$\psi(\pi_1) \geqslant \psi(\pi_2) + \langle \nabla \psi(\pi_2), \pi_1 - \pi_2 \rangle + \frac{1}{2} \|\pi_2 - \pi_1\|^2.$$

Lets formulate lemma from [4]

Lemma 11 ([4]) *Consider the problem (4) under Assumption 10. Then, for every $\theta \in \mathbb{R}^d$ the function $\mathcal{L}(\theta, \pi)$ is λ-**strongly concave**, i.e. for all $\pi_1, \pi_2 \in S$ it satisfies*

$$\mathcal{L}(\theta, \pi_1) \leq \mathcal{L}(\theta, \pi_2) + \langle \nabla_\psi \mathcal{L}(\theta, \pi_2), \pi_1 - \pi_2 \rangle - \frac{\lambda}{2} \left(D_\psi(\pi_1, \pi_2) + D_\psi(\pi_2, \pi_1) \right).$$

C.2. Auxiliary lemmas

Notation 1 *For the saddle-point problem (4) and Algorithm 1, we use the following notation, aligned with [4]:*

$$g_\theta^t \equiv \frac{c}{B} \sum_{j=1}^{B} \pi_{c_j^t} \nabla_\theta f_{c_j^t, i_j^t}(\theta^t), \qquad \textit{stochastic gradient w.r.t. } \theta,$$

$$g_\pi^t \equiv \frac{c}{B} \sum_{j=1}^{B} e_{c_j^t} f_{c_j^t, i_j^t}(\theta^t) - \lambda \nabla_\pi D_\psi(\pi^t \| \hat{\pi}), \qquad \textit{stochastic gradient w.r.t. } \pi,$$

γ_θ — *stepsize for* θ, $\qquad$ γ_π — *stepsize for* π,

$$\mathcal{L}(\theta, \pi) \equiv \sum_{i=1}^{c} \pi_i \Big(\tfrac{c}{n} \sum_{j=1}^{n_i} f_{i,j}(\theta) \Big) + \tfrac{\tau}{2} \|\theta\|_2^2 - \lambda D_\psi(\pi \| \hat{\pi}), \qquad S \textit{ — feasible set for } \pi.$$

Here e_i denotes the i-th standard basis vector in $\mathbb{R}^c$, $\hat{\pi}$ is the reference distribution in the regularization term, and $\nabla_\pi D_\psi(\pi^t \| \hat{\pi})$ denotes the gradient (or subgradient) of the divergence D_ψ with respect to π.

According to the notation, Algorithm 1 can be formulated in a simpler form:

$$\theta^{t+1} = \theta^t \; - \; \gamma_\theta \, d_\theta^t,$$

$$\pi^{t+1} = \arg\min_{\pi \in S} \Big\{ \langle -\gamma_\pi g_\pi^t, \; \pi \rangle \; + \; D_\psi(\pi \,\|\, \pi^t) \Big\},$$

where d_θ^t is classsical Adam step.

We begin by noting that our convergence analysis is based on the Adam estimator. Let us introduce the main Adam Estimator process:

$$\theta^{t+1} = \theta^t - \gamma_\theta d_\theta^t \; = \; \theta^t - \gamma_\theta \frac{m_\theta^t}{b_t}, \tag{5}$$

$$\pi^{t+1} = \arg\min_{\pi \in S} \Big\{ \langle -\gamma_\pi g_\pi^t, \; \pi \rangle + D_\psi(\pi \,\|\, \pi^t) \Big\}. \tag{6}$$

We also introduce a copy of the main process, which behaves identically to the original algorithm but is used to generate the scaling constant b_t for the main process:

$$\theta_{\text{copy}}^{t+1} = \theta_{\text{copy}}^t - \gamma_\theta \frac{m_{\theta,\text{copy}}^t}{b_t},$$

$$\pi_{\text{copy}}^{t+1} = \arg\min_{\pi \in S} \Big\{ \langle -\gamma_\pi \tilde{g}_\pi^t, \; \pi \rangle + D_\psi(\pi \,\|\, \pi_{\text{copy}}^t) \Big\}.$$

The update rules for the copy and main processes are:

$$m_{\theta,\text{copy}}^t = \beta_1 m_{\theta,\text{copy}}^{t-1} + (1-\beta_1)\tilde{g}_\theta^t,$$

$$b_t^2 = \beta_2 b_{t-1}^2 + (1-\beta_2)\|\tilde{g}_\theta^t\|^2,$$

$$m_\theta^t = \beta_1 m_\theta^{t-1} + (1-\beta_1) g_\theta^t,$$

where g_θ^t is the stochastic gradient with respect to θ at the point (θ^t, π^t), and $\tilde{g}_\theta^t$ is the stochastic gradient at the point $(\theta_{\text{copy}}^t, \pi_{\text{copy}}^t)$.

The first moment m_θ^t admits a closed-form expression:

$$m_\theta^t \; = \; (1-\beta_1) \sum_{k=0}^{t} \beta_1^{t-k} \, g_\theta^k.$$

We initialize

$$m_{\theta,\text{copy}}^{-1} = m_\theta^{-1} = 0, \qquad b_{-1}, b_0 > 0.$$

The purpose of introducing the copy process is to decouple the randomness of the estimator: in the original process, products of random variables inside expectations are dependent, while in the proposed estimator the corresponding quantities can be treated as independent, which allows us to move products under the expectation in the convergence analysis.

According to the above, the next lemma holds.

Lemma 12 ([7], Lemma 13) *For a reference step $r \le t$, and letting $\beta_2 = 1 - \frac{1}{K}$ for some $K \ge t - r$, the following lower bound holds:*

$$b_t^2 \ge \beta_2^{t-r} b_r^2 = \left(1 - \frac{1}{K}\right)^{t-r} b_r^2 \ge \left(1 - \frac{1}{K}\right)^{K} b_r^2 \ge c_m^2 b_r^2,$$

where for our Adam-type estimator, we can choose $c_m = \frac{1}{2}$.

Now let us formulate a technical lemma, which we will need in the future to evaluate the resulting sums:

Lemma 13 *Let $a_t = -\langle \nabla\Phi(\theta^t), d_\theta^t \rangle$ and $\xi_t = -\langle \nabla\Phi(\theta^t), g_\theta^t \rangle$, where d_θ^t is the Adam estimator step and g_θ^t is the stochastic gradient used for the momentum term in the Adam estimator 5, and θ^t is the iterate of the main process at step t. Then, the following inequality holds:*

$$\sum_{t=0}^{T} a_t \;\le\; \sum_{k=0}^{T} C_k \xi_k \;+\; 3\gamma_\theta \kappa L \sum_{k=0}^{T-1} A_k \|d_\theta^k\|^2,$$

where

$$C_k = (1-\beta_1) \sum_{t=k}^{T} \frac{\beta_1^{t-k}}{b_t}, \qquad A_k = b_k \sum_{t=k+1}^{T} \frac{\beta_1^{t-k}}{b_t}.$$

Proof According to the update rule, we have

$$a_t = \frac{1}{b_t}\left((1-\beta_1)\xi_t - \langle \nabla\Phi(\theta^t), \beta_1 m_\theta^{t-1} \rangle\right).$$

Hence, we get

$$\begin{aligned} a_t &= \frac{1}{b_t}\left((1-\beta_1)\xi_t + \langle \nabla\Phi(\theta^{t-1}) - \nabla\Phi(\theta^t) - \nabla\Phi(\theta^{t-1}), \beta_1 m_\theta^{t-1} \rangle\right) \\ &= \frac{1}{b_t}\left((1-\beta_1)\xi_t + \beta_1 b_{t-1} a_{t-1} + \langle \nabla\Phi(\theta^{t-1}) - \nabla\Phi(\theta^t), \beta_1 m_\theta^{t-1} \rangle\right). \end{aligned}$$

Using $3\kappa L$-Lipschitzness of Φ, the last term can be decomposed as follows:

$$\begin{aligned} \langle \nabla\Phi(\theta^{t-1}) - \nabla\Phi(\theta^t), \beta_1 m_\theta^{t-1} \rangle &\le 3\beta_1 \kappa L \|\theta^t - \theta^{t-1}\| \|m_\theta^{t-1}\| \\ &\le 3\gamma_\theta \kappa L \beta_1 b_{t-1} \|d_\theta^{t-1}\|^2, \end{aligned}$$

where in the second inequality we apply the property of the proximal operator. Thus, one can obtain

$$a_t \le \frac{1}{b_t}(1-\beta_1)\xi_t + \beta_1 \frac{b_{t-1}}{b_t} a_{t-1} + 3\gamma_\theta \kappa L \beta_1 \frac{b_{t-1}}{b_t} \|d_\theta^{t-1}\|^2.$$

Running the recursion over a_t, we have

$$a_t \le \frac{1}{b_t} \sum_{k=0}^{t} (1-\beta_1)\, \beta_1^{t-k} \xi_k \;+\; 3\gamma_\theta \kappa L \sum_{k=0}^{t-1} \beta_1^{t-k} \frac{b_k}{b_t} \|d_\theta^k\|^2.$$

Summing over $t = 0$ to T, we get:

$$\sum_{t=0}^{T} a_t \le \sum_{t=0}^{T} \frac{1}{b_t} \sum_{k=0}^{t} (1-\beta_1)\beta_1^{t-k}\,\xi_k \;+\; 3\gamma_\theta \kappa L \sum_{t=0}^{T}\sum_{k=0}^{t-1} \frac{\beta_1^{t-k} b_k}{b_t} \|d_\theta^k\|^2.$$

Switching the order of sums in the second term leads to

$$\sum_{t=0}^{T} a_t = \sum_{t=0}^{T} \frac{1}{b_t} \sum_{k=0}^{t} (1-\beta_1)\beta_1^{t-k}\,\xi_k \;+\; 3\gamma_\theta \kappa L \sum_{k=0}^{T-1} b_k \|d_\theta^k\|^2 \sum_{t=k+1}^{T} \frac{\beta_1^{t-k}}{b_t}.$$

Thus, the overall summed inequality becomes:

$$\sum_{t=0}^{T} a_t \le \sum_{k=0}^{T} C_k \xi_k + 3\gamma_\theta \kappa L \sum_{k=0}^{T-1} A_k \|d_\theta^k\|^2,$$

where:

$$C_k = (1-\beta_1) \sum_{t=k}^{T} \frac{\beta_1^{t-k}}{b_t}, \qquad A_k = b_k \sum_{t=k+1}^{T} \frac{\beta_1^{t-k}}{b_t}.$$

This finishes the proof. ■

The next lemma, that is useful for us, help us to upper bound distance between momentum and stochastic gradient:

Lemma 14 *Let g_t is stochastic gradient, and m_t is momentum of the Adam estimator 5 then distance between them such as folowing:*

$$\|g_t - m_t\|^2 \le \beta_1^2 \cdot G_t, \tag{7}$$

where β_1 is parameter in Adam and $G_t = 2\left(\|g_t\|^2 + (1-\beta_1)\sum_{k=0}^{t-1} \beta_1^{t-k}\|g_k\|^2\right)$.

Proof

$$\begin{aligned}\|g_t - m_t\|^2 = \|g_t - (1-\beta_1)g_t - \beta_1 m_{t-1}\|^2 &= \beta_1^2\|g_t - m_{t-1}\|^2 \\ &\le 2\beta_1^2\left(\|g_t\|^2 + \|m_{t-1}\|^2\right)\end{aligned}$$

We know that recursion on momentum m_t is revealed in the following:

$$m_{t-1} = (1-\beta_1)g_{t-1} + m_{t-2} = (1-\beta_1)\sum_{k=0}^{t-1} \beta_1^{t-k} g_k$$

Using convexity of $\|\cdot\|^2$ we have:

$$\begin{aligned}\|m_{t-1}\|^2 = \|(1-\beta_1)\sum_{k=0}^{t-1}\beta_1^{t-k} g_k\|^2 &\le (1-\beta_1)^2 \frac{1}{1-\beta_1^t} \sum_{k=0}^{t-1} \beta_1^{t-k}\|g_k\|^2 \\ &\le (1-\beta_1)\sum_{k=0}^{t-1}\beta_1^{t-k}\|g_k\|^2\end{aligned}$$

■

Now we can move on to the main theorem.

C.3. Main lemmas and theorem

C.3.1. Main lemma

Lemma 15 (Stochastic distance recursion) *Consider the problem (4) under Assumptions 7, 10, and 3. Let $g_t = \nabla_\pi \mathcal{L}(\theta^t, \pi^t; \zeta_t)$ be the stochastic gradient computed using a mini-batch of size B, and let $\xi_t := g_t - \nabla_\pi \mathcal{L}(\theta^t, \pi^t)$ be the noise term. Then, Algorithm 5 with tuning*

$$\gamma_\pi = \frac{\lambda}{8L^2}, \qquad \gamma_\theta \leq \frac{c_m b_0}{1048\, L\, \kappa^4},$$

produces a sequence $\{(\theta^t, \pi^t)\}_{t=1}^T$ such that

$$\begin{aligned}\mathbb{E}[D_\psi(\pi^*(\theta^{t+1}), \pi^{t+1})] \leq \left(1 - \tfrac{1}{128\kappa^2}\right) \mathbb{E}[D_\psi(\pi^*(\theta^t), \pi^t)] \\ + \gamma_\theta^2\, C_\Phi\, \mathbb{E}\|\nabla \Phi(\theta^t)\|^2 + \gamma_\theta^2\, C_B\, \tfrac{\sigma^2}{B} + \gamma_\theta^2\, \beta_1^2 C_\beta,\end{aligned}$$

where the constants are

$$C_\Phi = \frac{2080\, \kappa^6}{c_m^2 b_0^2}, \qquad C_B = \frac{1040\, \kappa^6}{c_m^2 b_0^2} + \frac{\lambda^2}{32L^4}, \qquad C_\beta = \frac{8320\, \kappa^6}{c_m^2 b_0^2}\Big(K^2 + \tfrac{\sigma^2}{B}\Big).$$

Proof To begin, we use three-point identity:

$$\begin{aligned}D_\psi(\pi^*(\theta^{t+1}), \pi^{t+1}) =& D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t)) + D_\psi(\pi^*(\theta^t), \pi^{t+1}) \\ &+ \langle \nabla\psi(\pi^*(\theta^t)) - \nabla\psi(\pi^{t+1}), \pi^*(\theta^{t+1}) - \pi^*(\theta^t)\rangle.\end{aligned} \tag{8}$$

Further, we write the optimality condition for the stochastic mirror-ascent step:

$$\left\langle -\gamma_\pi g_t + [\nabla\psi(\pi^{t+1}) - \nabla\psi(\pi^t)], \pi^*(\theta^t) - \pi^{t+1} \right\rangle \geq 0.$$

Applying (8), we obtain

$$-\gamma_\pi \left\langle g_t, \pi^*(\theta^t) - \pi^{t+1} \right\rangle + D_\psi(\pi^*(\theta^t), \pi^t) - D_\psi(\pi^*(\theta^t), \pi^{t+1}) - D_\psi(\pi^{t+1}, \pi^t) \geq 0.$$

Substituting $g_t = \nabla_\pi \mathcal{L}(\theta^t, \pi^t) + \xi_t$, we get:

$$\begin{aligned}-\gamma_\pi \left\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^t), \pi^*(\theta^t) - \pi^{t+1} \right\rangle - \gamma_\pi \left\langle \xi_t, \pi^*(\theta^t) - \pi^{t+1} \right\rangle \\ + D_\psi(\pi^*(\theta^t), \pi^t) - D_\psi(\pi^*(\theta^t), \pi^{t+1}) - D_\psi(\pi^{t+1}, \pi^t) \geq 0.\end{aligned}$$

After re-arranging the terms, we get

$$D_\psi(\pi^*(\theta^t), \pi^{t+1}) \leq D_\psi(\pi^*(\theta^t), \pi^t) - D_\psi(\pi^{t+1}, \pi^t) - \tag{9}$$

$$\gamma_\pi \left\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^t), \pi^*(\theta^t) - \pi^{t+1} \right\rangle - \gamma_\pi \left\langle \xi_t, \pi^*(\theta^t) - \pi^{t+1} \right\rangle. \tag{10}$$

Since $\pi^*(\theta^t)$ is the exact maximum of $\mathcal{L}(\theta^t, \pi)$ in π, there is another optimility condition

$$\gamma_\pi \left\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)), \pi^*(\theta^t) - \pi \right\rangle \geq 0.$$

Substituting $\pi = \pi^{t+1}$ and summing it with (9), we derive

$$\begin{aligned}
D_\psi(\pi^*(\theta^t), \pi^{t+1}) \leq & D_\psi(\pi^*(\theta^t), \pi^t) - D_\psi(\pi^{t+1}, \pi^t) \\
& + \gamma_\pi \left\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)) - \nabla_\pi \mathcal{L}(\theta^t, \pi^t), \pi^*(\theta^t) - \pi^{t+1} \right\rangle - \gamma_\pi \left\langle \xi_t, \pi^*(\theta^t) - \pi^{t+1} \right\rangle \\
\leq & D_\psi(\pi^*(\theta^t), \pi^t) - D_\psi(\pi^{t+1}, \pi^t) \\
& + \gamma_\pi \left\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)) - \nabla_\pi \mathcal{L}(\theta^t, \pi^t), \pi^*(\theta^t) - \pi^t \right\rangle \\
& + \gamma_\pi \left\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)) - \nabla_\pi \mathcal{L}(\theta^t, \pi^t), \pi^t - \pi^{t+1} \right\rangle - \\
& \gamma_\pi \left\langle \xi_t, \pi^*(\theta^t) - \pi^t \right\rangle - \gamma_\pi \left\langle \xi_t, \pi^t - \pi^{t+1} \right\rangle .
\end{aligned}$$

Now, we are going to utilize the strong concavity of $\mathcal{L}(\theta, \pi)$ in π:

$$\gamma_\pi \left\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)) - \nabla_\pi \mathcal{L}(\theta^t, \pi^t), \pi^*(\theta^t) - \pi^t \right\rangle \leq \frac{-\gamma_\pi \lambda}{2} D_\psi(\pi^*(\theta^t), \pi^t).$$

Thus, we have

$$\begin{aligned}
D_\psi(\pi^*(\theta^t), \pi^{t+1}) \leq & \left(1 - \frac{\gamma_\pi \lambda}{2}\right) D_\psi(\pi^*(\theta^t), \pi^t) - D_\psi(\pi^{t+1}, \pi^t) \\
& + \gamma_\pi \left\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)) - \nabla_\pi \mathcal{L}(\theta^t, \pi^t), \pi^t - \pi^{t+1} \right\rangle - \gamma_\pi \left\langle \xi_t, \pi^*(\theta^t) - \pi^{t+1} \right\rangle .
\end{aligned}$$

Next, we apply Cauchy-Schwartz inequality to the scalar product and obtain

$$\begin{aligned}
D_\psi(\pi^*(\theta^t), \pi^{t+1}) \leq & \left(1 - \frac{\gamma_\pi \lambda}{2}\right) D_\psi(\pi^*(\theta^t), \pi^t) - D_\psi(\pi^{t+1}, \pi^t) \\
& + \frac{\gamma_\pi \alpha}{2} \|\nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)) - \nabla_\pi \mathcal{L}(\theta^t, \pi^t)\|^2 + \frac{\gamma_\pi}{2\alpha} \|\pi^t - \pi^{t+1}\|^2 \\
& - \gamma_\pi \left\langle \xi_t, \pi^*(\theta^t) - \pi^t \right\rangle - \gamma_\pi \left\langle \xi_t, \pi^t - \pi^{t+1} \right\rangle .
\end{aligned}$$

For the stochastic noise terms, we apply Young's inequality in Bregman geometry:

$$-\gamma_\pi \left\langle \xi_t, \pi^t - \pi^{t+1} \right\rangle \ \leq \ \gamma_\pi^2 \|\xi_t\|_*^2 + \tfrac{1}{2} D_\psi(\pi^{t+1}, \pi^t).$$

Using L-smoothness of $\mathcal{L}$ (see Assumption 7) and ψ is 1-strongly convex (see Assumption 10), we obtain

$$\begin{aligned}
D_\psi(\pi^*(\theta^t), \pi^{t+1}) \leq & \left(1 - \frac{\gamma_\pi \lambda}{2}\right) D_\psi(\pi^*(\theta^t), \pi^t) - D_\psi(\pi^{t+1}, \pi^t) + \frac{1}{2} D_\psi(\pi^{t+1}, \pi^t) \\
& + \gamma_\pi \alpha L^2 D_\psi(\pi^*(\theta^t), \pi^t) + \frac{\gamma_\pi}{\alpha} D_\psi(\pi^{t+1}, \pi^t) \\
& - \gamma_\pi \left\langle \xi_t, \pi^*(\theta^t) - \pi^t \right\rangle \ + \ \gamma_\pi^2 \|\xi_t\|_*^2.
\end{aligned}$$

Choose $\alpha = 2\gamma_\pi$. Substituting this into the previous inequality and reducing terms $D_\psi(\pi^{t+1}, \pi^t)$, we get

$$\begin{aligned}
D_\psi(\pi^*(\theta^t), \pi^{t+1}) \leq & \left(1 - \frac{\gamma_\pi \lambda}{2}\right) D_\psi(\pi^*(\theta^t), \pi^t) \\
& + 2\gamma_\pi^2 L^2 D_\psi(\pi^*(\theta^t), \pi^t) \\
& - \gamma_\pi \left\langle \xi_t, \pi^*(\theta^t) - \pi^t \right\rangle \ + \ \gamma_\pi^2 \|\xi_t\|_*^2.
\end{aligned}$$

Taking conditional expectation $\mathbb{E}[\cdot \mid \mathcal{F}_t]$ and using $\mathbb{E}[\langle \xi_t, \pi^*(\theta^t) - \pi^t \rangle \mid \mathcal{F}_t] = 0$, we obtain

$$\mathbb{E}\big[D_\psi(\pi^*(\theta^t), \pi^{t+1}) \mid \mathcal{F}_t\big] \leq \left(1 - \frac{\gamma_\pi \lambda}{2} + 2\gamma_\pi^2 L^2\right) D_\psi(\pi^*(\theta^t), \pi^t) \ + \ \gamma_\pi^2 \frac{\sigma^2}{B}. \tag{11}$$

The stepsize that minimizes the quadratic factor is

$$\gamma_\pi = \frac{\lambda}{8L^2}.$$

Substituting this choice and applying full expectation yields

$$\mathbb{E}\big[D_\psi(\pi^*(\theta^t), \pi^{t+1})\big] \leq \left(1 - \frac{1}{32\kappa^2}\right) \mathbb{E}\big[D_\psi(\pi^*(\theta^t), \pi^t)\big] + \frac{\lambda^2}{64L^4} \frac{\sigma^2}{B}, \tag{12}$$

where $\kappa = \frac{L}{\lambda}$ is the condition number.

Let us return to (8). Note that

$$\nabla \psi(\pi^*(\theta^t)) - \nabla \psi(\pi^{t+1}) = \frac{1}{\lambda} \left(\nabla_\pi \mathcal{L}(\theta^t, \pi^{t+1}) - \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t))\right).$$

Thus, there is

$$\begin{aligned}
D_\psi(\pi^*(\theta^{t+1}), \pi^{t+1}) =& D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t)) + D_\psi(\pi^*(\theta^t), \pi^{t+1}) \\
&+ \frac{1}{\lambda} \langle \nabla_\pi \mathcal{L}(\theta^t, \pi^{t+1}) - \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)), \pi^*(\theta^{t+1}) - \pi^*(\theta^t) \rangle \\
\leq& D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t)) + D_\psi(\pi^*(\theta^t), \pi^{t+1}) \\
&+ \frac{\alpha L^2}{\lambda} D_\psi(\pi^*(\theta^t), \pi^{t+1}) + \frac{1}{\lambda \alpha} D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t)).
\end{aligned}$$

Let us choose $\alpha = \lambda^3/64L^4$. With such a choice and using fact that $\kappa \geq 1$, we have

$$D_\psi(\pi^*(\theta^{t+1}), \pi^{t+1}) \leq 65\kappa^4 D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t)) + \left(1 + \frac{1}{64\kappa^2}\right) D_\psi(\pi^*(\theta^t), \pi^{t+1}).$$

To deal with $D_\psi(\pi^*(\theta^t), \pi^{t+1})$, we utilize (12). Using $(1 + \frac{1}{64\kappa^2})(1 - \frac{1}{32\kappa^2}) \leq 1 - \frac{1}{64\kappa^2}$ and $1 + \frac{1}{64\kappa^2} \leq 2$ we obtain

$$\mathbb{E}\big[D_\psi(\pi^*(\theta^{t+1}), \pi^{t+1})\big] \leq 65\kappa^4 \mathbb{E}\big[D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t))\big] + \tag{13}$$

$$\left(1 - \frac{1}{64\kappa^2}\right) \mathbb{E}\big[D_\psi(\pi^*(\theta^t), \pi^t)\big] + \frac{\lambda^2}{32L^4} \frac{\sigma^2}{B}. \tag{14}$$

The remaining task is to prove that the descent step does not dramatically change the distance between the optimal values of weights. Let us write down two optimality conditions:

$$\begin{aligned}
&\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)), \pi - \pi^*(\theta^t) \rangle \leq 0, \\
&\langle \nabla_\pi \mathcal{L}(\theta^{t+1}, \pi^*(\theta^{t+1})), \pi - \pi^*(\theta^{t+1}) \rangle \leq 0.
\end{aligned}$$

Let us substitute $\pi = \pi^*(\theta^{t+1})$ into the first inequality and $\pi = \pi^*(\theta^t)$ into the second one. When summing them up, we have

$$\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)) - \nabla_\pi \mathcal{L}(\theta^{t+1}, \pi^*(\theta^{t+1})), \pi^*(\theta^{t+1}) - \pi^*(\theta^t) \rangle \le 0. \tag{15}$$

On the other hand, we can take advantage of the strong concavity of the objective (see Lemma 11) and write

$$\langle \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^{t+1})) - \nabla_\pi \mathcal{L}(\theta^t, \pi^*(\theta^t)), \pi^*(\theta^{t+1}) - \pi^*(\theta^t) \rangle \tag{16}$$

$$\le -\frac{\lambda}{2} \left[D_\psi(\pi^*(\theta^t), \pi^*(\theta^{t+1})) + D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t)) \right]. \tag{17}$$

Combining (15) and (16), we obtain

$$\frac{\lambda^2}{4} \left[D_\psi(\pi^*(\theta^t), \pi^*(\theta^{t+1})) + D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t)) \right]^2 \le L^2 \|\pi^*(\theta^{t+1}) - \pi^*(\theta^t)\|^2 \|\theta^{t+1} - \theta^t\|^2.$$

Re-arranging the terms and substituting Adam estimator step, we derive

$$\left[D_\psi(\pi^*(\theta^t), \pi^*(\theta^{t+1})) + D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t)) \right] \le 4\kappa^2 \|\theta^{t+1} - \theta^t\|^2 \equiv 4\gamma_\theta^2 \kappa^2 \left\| d_\theta^t \right\|^2.$$

After simplifying, we have

$$D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t)) \le 4\gamma_\theta^2 \kappa^2 \left\| d_\theta^t \right\|^2.$$

Using lemma 14 and lemma 12:

$$\|d_\theta^t\|^2 = \|\frac{m_\theta^t}{b_t}\|^2 \le \frac{1}{c_m^2 b_0^2} \|m_\theta^t\|^2 \le \frac{4}{c_m^2 b_0^2} \left(\|g_\theta^t - m_\theta^t\|^2 + \|\nabla_\theta \mathcal{L}(\theta^t, \pi^t)\|^2 + \|\xi_t\|^2 \right) \tag{18}$$

$$\le \frac{4}{c_m^2 b_0^2} \left(\beta_1^2 \cdot G_t + \|\nabla_\theta \mathcal{L}(\theta^t, \pi^t)\|^2 + \|\xi_t\|^2 \right), \tag{19}$$

where $\xi_t = \nabla_\theta \mathcal{L}(\theta^t, \pi^t) - g_\theta^t$ is the stochastic gradient noise,

$$G_t = 2 \left(\|g_\theta^t\|^2 + (1 - \beta_1) \sum_{k=0}^{t-1} \beta_1^{t-k} \|g_\theta^k\|^2 \right)$$

.

Using L-smoothness of $\mathcal{L}$ (see Assumption 7) and ψ is 1-strongly convex (see Assumption 10), we obtain

$$\begin{aligned} \|\nabla_\theta \mathcal{L}(\theta^t, \pi^t)\|^2 &\le 2 \left(\|\nabla \Phi(\theta^t)\|^2 + \|\nabla_\theta \mathcal{L}(\theta^t, \pi^t) - \nabla \Phi(\theta^t)\|^2 \right) \\ &\le 2\|\nabla \Phi(\theta^t)\|^2 + 4L^2 D_\psi(\pi^*(\theta^t), \pi^t) \end{aligned}$$

Applying expectation and using assumption 3 we have:

$$\mathbb{E}\|d_\theta^t\|^2 \le \frac{4}{c_m^2 b_0^2} \Big(\beta_1^2 \cdot \mathbb{E}[G_t] + 2\,\mathbb{E}\|\nabla \Phi(\theta^t)\|^2 + 4L^2\, \mathbb{E}\big[D_\psi(\pi^*(\theta^t), \pi^t)\big] + \frac{\sigma^2}{B} \Big). \tag{20}$$

Setting $\tau = 0$ and using K-Lipschitzness 9 of $\mathcal{L}$ and boundess of variance 3, we have

$$\|g_\theta^k\|^2 \le 2K^2 + \tfrac{2\sigma^2}{B} \quad \Rightarrow \quad \mathbb{E}[G_t] \le 8K^2 + \tfrac{8\sigma^2}{B}. \tag{21}$$

After substituting inequality 21 into 20 we obtain

$$\mathbb{E}\|d_\theta^t\|^2 = \frac{4}{c_m^2 b_0^2}\Big(\beta_1^2 \cdot 8(K^2 + \frac{\sigma^2}{B}) + 2\,\mathbb{E}\|\nabla\Phi(\theta^t)\|^2 + 4L^2\,\mathbb{E}[D_\psi(\pi^*(\theta^t), \pi^t)] + \frac{\sigma^2}{B}\Big). \tag{22}$$

Let us take an expectation and derive

$$\mathbb{E}\,D_\psi(\pi^*(\theta^{t+1}), \pi^*(\theta^t)) \le \frac{16\gamma_\theta^2\kappa^2}{c_m^2 b_0^2}\Big(8\beta_1^2\Big(K^2 + \tfrac{\sigma^2}{B}\Big) + 2\,\mathbb{E}\,\|\nabla\Phi(\theta^t)\|^2 + 4L^2\,\mathbb{E}[D_\psi(\pi^*(\theta^t), \pi^t)] + \tfrac{\sigma^2}{B}\Big).$$

Substituting this into (13) we have

$$\begin{aligned}\mathbb{E}[D_\psi(\pi^*(\theta^{t+1}), \pi^{t+1})] \le{}& \frac{1040\,\gamma_\theta^2\kappa^6}{c_m^2 b_0^2}\Big(8\beta_1^2(K^2 + \tfrac{\sigma^2}{B}) + 2\,\mathbb{E}\|\nabla\Phi(\theta^t)\|^2 + 4L^2\,\mathbb{E}[D_\psi(\pi^*(\theta^t), \pi^t)] + \tfrac{\sigma^2}{B}\Big)\\ &+ \Big(1 - \tfrac{1}{64\kappa^2}\Big)\mathbb{E}[D_\psi(\pi^*(\theta^t), \pi^t)] + \tfrac{\lambda^2}{32L^4}\,\tfrac{\sigma^2}{B}.\end{aligned}$$

Using $\gamma_\theta \le \frac{c_m b_0}{1048\,L\,\kappa^4}$ and substituting (22) into (13), we have

$$\begin{aligned}\mathbb{E}\big[D_\psi(\pi^*(\theta^{t+1}), \pi^{t+1})\big] \le{}& \Big(1 - \tfrac{1}{128\kappa^2}\Big)\,\mathbb{E}\big[D_\psi(\pi^*(\theta^t), \pi^t)\big]\\ &+ \frac{1040\,\gamma_\theta^2\kappa^6}{c_m^2 b_0^2}\Big(8\beta_1^2(K^2 + \tfrac{\sigma^2}{B}) + 2\,\mathbb{E}\|\nabla\Phi(\theta^t)\|^2 + \tfrac{\sigma^2}{B}\Big)\\ &+ \tfrac{\lambda^2}{32L^4}\,\tfrac{\sigma^2}{B}.\end{aligned}$$

Collecting terms, we obtain

$$\begin{aligned}\mathbb{E}[D_\psi(\pi^*(\theta^{t+1}), \pi^{t+1})] \le{}& \Big(1 - \tfrac{1}{128\kappa^2}\Big)\,\mathbb{E}[D_\psi(\pi^*(\theta^t), \pi^t)]\\ &+ \gamma_\theta^2\,C_\Phi\,\mathbb{E}\|\nabla\Phi(\theta^t)\|^2 + \gamma_\theta^2\,C_B\,\tfrac{\sigma^2}{B} + \gamma_\theta^2\,\beta_1^2 C_\beta,\end{aligned}$$

where the constants are

$$C_\Phi = \frac{2080\,\kappa^6}{c_m^2 b_0^2}, \qquad C_B = \frac{1040\,\kappa^6}{c_m^2 b_0^2} + \frac{\lambda^2}{32L^4}, \qquad C_\beta = \frac{8320\,\kappa^6}{c_m^2 b_0^2}\Big(K^2 + \tfrac{\sigma^2}{B}\Big).$$

This completes the proof of the stochastic version of the main lemma. ■

C.3.2. Main theorem

Now let us proceed to the convergence proof for Algorithm 1.

Proof 5 One can note that Φ is $3\kappa L$-smooth. Indeed,

$$\begin{aligned}\|\nabla\Phi(\theta_1) - \nabla\Phi(\theta_2)\|^2 ={}& \|\nabla_\theta\mathcal{L}(\theta_1, \pi^*(\theta_1)) - \nabla_\theta\mathcal{L}(\theta_2, \pi^*(\theta_2))\|^2\\ \le{}& L^2\left[\|\theta_1 - \theta_2\|^2 + 2D_\psi(\pi^*(\theta_1), \pi^*(\theta_2))\right] \le L^2\left(1 + 4\kappa^2\right)\|\theta_1 - \theta_2\|^2\\ \le{}& 9\kappa^2 L^2\|\theta_1 - \theta_2\|^2.\end{aligned}$$

Thus, we can write

$$\begin{aligned}\Phi(\theta^{t+1}) \leq& \Phi(\theta^t) + \langle \nabla\Phi(\theta^t), \theta^{t+1} - \theta^t \rangle + 3\kappa L \|\theta^{t+1} - \theta^t\|^2 \\ =& \Phi(\theta^t) - \gamma_\theta \left\langle \nabla\Phi(\theta^t), d_\theta^t \right\rangle + 3\gamma_\theta^2 \kappa L \left\| d_\theta^t \right\|^2\end{aligned}$$

Summing from $t = 0$ to T yields

$$\Phi(\theta^{T+1}) \leq \Phi(\theta^0) - \gamma_\theta \sum_{t=0}^{T} \langle \nabla\Phi(\theta^t), d_\theta^t \rangle + 3\gamma_\theta^2 \kappa L \sum_{t=0}^{T} \|d_\theta^t\|^2.$$

Applying lemma 13 with $a_t = -\left\langle \nabla\Phi(\theta^t), d_\theta^t \right\rangle$ we have:

$$\Phi(\theta^{T+1}) \leq \Phi(\theta^0) + \gamma_\theta \sum_{k=0}^{T} C_k \xi_k + 3\gamma_\theta^2 \kappa L \sum_{k=0}^{T} (1 + A_k) \|d_\theta^k\|^2,$$

where $\xi_k = -\langle \nabla\Phi(\theta^k), g_\theta^k \rangle$ and g_θ^k is the stochastic gradient in the Adam estimator 5.

By decomposing the stochastic gradient into the true gradient and the noise $g_\theta^k = \nabla_\theta \mathcal{L}(\theta^k, \pi^k) + \eta_k$, we have

$$\begin{aligned}\Phi(\theta^{T+1}) \leq \Phi(\theta^0) - \gamma_\theta \sum_{k=0}^{T} C_k \left\langle \nabla\Phi(\theta^k), \nabla_\theta \mathcal{L}(\theta^k, \pi^k) \right\rangle \\ - \gamma_\theta \sum_{k=0}^{T} C_k \left\langle \nabla\Phi(\theta^k), \eta_k \right\rangle + 3\gamma_\theta^2 \kappa L \sum_{k=0}^{T} (1 + A_k) \|d_\theta^k\|^2.\end{aligned}$$

Rearranging the terms and dividing by γ_θ yields

$$\sum_{k=0}^{T} C_k \left\langle \nabla\Phi(\theta^k), \nabla_\theta \mathcal{L}(\theta^k, \pi^k) \right\rangle \leq \tag{23}$$

$$\frac{\Phi(\theta^0) - \Phi(\theta^{T+1})}{\gamma_\theta} - \sum_{k=0}^{T} C_k \left\langle \nabla\Phi(\theta^k), \eta_k \right\rangle + 3\gamma_\theta \kappa L \sum_{k=0}^{T-1} (1 + A_k) \|d_\theta^k\|^2. \tag{24}$$

Applying Young's inequality to the scalar product:

$$\left\langle \nabla\Phi(\theta^k),\, \nabla_\theta \mathcal{L}(\theta^k, \pi^k) \right\rangle \;\geq\; \frac{1}{2} \|\nabla\Phi(\theta^k)\|^2 \;-\; \frac{1}{2} \left\| \nabla_\theta \mathcal{L}(\theta^k, \pi^k) - \nabla\Phi(\theta^k) \right\|^2.$$

$$\begin{aligned}\frac{1}{2} \sum_{k=0}^{T} C_k \|\nabla\Phi(\theta^k)\|^2 - \frac{1}{2} \sum_{k=0}^{T} C_k \left\| \nabla_\theta \mathcal{L}(\theta^k, \pi^k) - \nabla\Phi(\theta^k) \right\|^2 \;\leq\; \frac{\Phi(\theta^0) - \Phi(\theta^{T+1})}{\gamma_\theta} \\ - \sum_{k=0}^{T} C_k \langle \nabla\Phi(\theta^k), \eta_k \rangle \;+\; 3\gamma_\theta \kappa L \sum_{k=0}^{T-1} (1 + A_k) \|d_\theta^k\|^2.\end{aligned} \tag{25}$$

Let $\mathcal{F}_k$ denote the history of the main process up to time k, and let the coefficients $C_k = (1 - \beta_1) \sum_{j=k}^{T} \beta_1^{j-k} / b_j$ be generated by an auxiliary (copy) sequence $\{b_j\}_{j \geq 0}$. Since C_k

depends only on future $\{b_j\}_{j\geq k}$ from the copy process, while $r_k := \langle \nabla\Phi(\theta^k), \eta_k\rangle$ is generated by the main process at time k, we have the conditional independence of C_k and r_k with respect to $(\mathcal{F}_k, \text{copy})$. Using the unbiasedness $\mathbb{E}[\eta_k \mid \mathcal{F}_k] = 0$, the tower property gives

$$\mathbb{E}[C_k\, r_k] = \mathbb{E}[\,\mathbb{E}[C_k\, r_k \mid \mathcal{F}_k, \text{copy}]] = \mathbb{E}[\,\mathbb{E}[C_k \mid \mathcal{F}_k, \text{copy}]\,\mathbb{E}[r_k \mid \mathcal{F}_k]] = 0.$$

Taking conditional expectation of (24) and then applying the tower property, we obtain

$$\begin{aligned}\frac{1}{2}\sum_{k=0}^{T}\mathbb{E}\Big[C_k\|\nabla\Phi(\theta^k)\|^2\Big] - \frac{1}{2}\sum_{k=0}^{T}\mathbb{E}\Big[C_k\big\|\nabla_\theta\mathcal{L}(\theta^k,\pi^k) - \nabla\Phi(\theta^k)\big\|^2\Big] \;&\leq\; \frac{\Phi(\theta^0) - \mathbb{E}\,\Phi(\theta^{T+1})}{\gamma_\theta} \\ &+ 3\gamma_\theta\kappa L\sum_{k=0}^{T-1}\mathbb{E}\Big[(1+A_k)\|d_\theta^k\|^2\Big]. \end{aligned} \tag{26}$$

To separate the factors on the left, use conditional independence as above:

$$\mathbb{E}\Big[C_k\|\nabla\Phi(\theta^k)\|^2 \mid \mathcal{F}_k, \text{copy}\Big] = \mathbb{E}[C_k \mid \text{copy}]\cdot\|\nabla\Phi(\theta^k)\|^2.$$

Hence

$$\mathbb{E}\Big[C_k\|\nabla\Phi(\theta^k)\|^2\Big] = \mathbb{E}\Big[\mathbb{E}[C_k \mid \text{copy}]\,\|\nabla\Phi(\theta^k)\|^2\Big].$$

Let us get the bound of the scaling parameter b_t in the Adam estimator 5:

$$\mathbb{E}\Big[\|g_\theta^t\|^2 \,\Big|\, \theta^k_{\text{copy}}, \pi^k_{\text{copy}}\Big] \;\leq\; 2\,(K^2 + \frac{\sigma^2}{B}), \tag{27}$$

$$\begin{aligned}\mathbb{E}\Big[b_i \,\Big|\, \theta^k_{\text{copy}}, \pi^k_{\text{copy}}\Big] \;&\leq\; \mathbb{E}\Big[\sqrt{\beta_2 b_{i-1}^2 + (1-\beta_2)\|\tilde{g}_\theta^t\|^2} \,\Big|\, \theta^k_{\text{copy}}, \pi^k_{\text{copy}}\Big] \\ &\leq\; \mathbb{E}\Big[\max\{b_{i-1}, \|\tilde{g}_\theta^t\|\} \,\Big|\, \theta^k_{\text{copy}}, \pi^k_{\text{copy}}\Big] \\ &\leq\; \max_i\sqrt{2K^2 + 2\frac{\sigma^2}{B}} \;=\; \sqrt{2K^2 + 2\frac{\sigma^2}{B}}.\end{aligned} \tag{28}$$

Using 12 we have

$$\mathbb{E}[C_k \mid \theta^k_{\text{copy}}, \pi^k_{\text{copy}}] = (1-\beta_1)\sum_{j=k}^{T}\frac{\beta_1^{\,j-k}}{\mathbb{E}[b_j \mid \theta^k_{\text{copy}}, \pi^k_{\text{copy}}]} \;\geq\; (1-\beta_1)\min_{j\in\{0,\dots,T\}}\frac{1}{\mathbb{E}[b_j \mid \theta^k_{\text{copy}}, \pi^k_{\text{copy}}]} \;\geq\; \frac{1-\beta_1}{\sqrt{2K^2 + 2\frac{\sigma^2}{B}}}$$

and

$$\mathbb{E}[C_k \mid \theta^k_{\text{copy}}, \pi^k_{\text{copy}}] \leq \frac{1}{c_m b_0}.$$

Therefore,

$$\sum_{k=0}^{T}\mathbb{E}\Big[C_k\|\nabla\Phi(\theta^k)\|^2\Big] \;\geq\; \frac{1-\beta_1}{\sqrt{2K^2 + 2\frac{\sigma^2}{B}}}\sum_{k=0}^{T}\mathbb{E}\Big[\|\nabla\Phi(\theta^k)\|^2\Big] \tag{29}$$

and

$$\sum_{k=0}^{T}\mathbb{E}\Big[C_k\big\|\nabla_\theta\mathcal{L}(\theta^k,\pi^k) - \nabla\Phi(\theta^k)\big\|^2\Big] \;\leq\; \frac{1}{c_m b_0}\sum_{k=0}^{T}\mathbb{E}\Big[\big\|\nabla_\theta\mathcal{L}(\theta^k,\pi^k) - \nabla\Phi(\theta^k)\big\|^2\Big]. \tag{30}$$

Combining (26) and (29), (30), we arrive at

$$\frac{1-\beta_1}{2\sqrt{2K^2+2\frac{\sigma^2}{B}}}\sum_{k=0}^{T}\frac{1}{2}\mathbb{E}\Big[\|\nabla\Phi(\theta^k)\|^2\Big]-\frac{1}{c_m b_0}\sum_{k=0}^{T}\frac{1}{2}\mathbb{E}\Big[\big\|\nabla_\theta\mathcal{L}(\theta^k,\pi^k)-\nabla\Phi(\theta^k)\big\|^2\Big]\ \le\ \frac{\Phi(\theta^0)-\mathbb{E}\,\Phi(\theta^{T+1})}{\gamma_\theta}$$
$$+3\gamma_\theta\kappa L\sum_{k=0}^{T-1}\mathbb{E}\Big[(1+A_k)\|d_\theta^k\|^2\Big]. \tag{31}$$

Using 22 we have:

$$\mathbb{E}\|d_\theta^t\|^2=\frac{4}{c_m^2 b_0^2}\Big(\beta_1^2\cdot 8(K^2+\frac{\sigma^2}{B})+2\,\mathbb{E}\|\nabla\Phi(\theta^t)\|^2+4L^2\,\mathbb{E}[D_\psi(\pi^*(\theta^t),\pi^t)]+\frac{\sigma^2}{B}\Big). \tag{32}$$

By definition of A_k:

$$\mathbb{E}A_t\le\tfrac{\beta_1}{c_m b_0(1-\beta_1)}\sqrt{2K^2+2\frac{\sigma^2}{B}},$$
$$\mathbb{E}\big[(1+A_t)\|d_\theta^t\|^2\big]\ \le\ \left(1+\frac{\beta_1}{c_m b_0(1-\beta_1)}\sqrt{2K^2+2\frac{\sigma^2}{B}}\right)$$
$$\cdot\frac{4}{c_m^2 b_0^2}\Big(\beta_1^2\cdot 8(K^2+\frac{\sigma^2}{B})+2\,\mathbb{E}\|\nabla\Phi(\theta^t)\|^2+4L^2\,\mathbb{E}[D_\psi(\pi^*(\theta^t),\pi^t)]+\frac{\sigma^2}{B}\Big).$$

$$C_A:=\frac{\beta_1}{c_m b_0(1-\beta_1)}\sqrt{2K^2+2\frac{\sigma^2}{B}},\qquad C_D:=\frac{4}{c_m^2 b_0^2}.$$

Then the auxiliary bounds read

$$\mathbb{E}A_t\le C_A,$$
$$\mathbb{E}\big[(1+A_t)\|d_\theta^t\|^2\big]\le(1+C_A)\,C_D\Big(\beta_1^2\cdot 8\Big(K^2+\tfrac{\sigma^2}{B}\Big)+2\,\mathbb{E}\|\nabla\Phi(\theta^t)\|^2+4L^2\,\mathbb{E}[D_\psi(\pi^*(\theta^t),\pi^t)]+\tfrac{\sigma^2}{B}\Big).$$

Substituting these inequalities into the main relation yields

$$\frac{1-\beta_1}{2\sqrt{2K^2+2\frac{\sigma^2}{B}}}\sum_{k=0}^{T}\frac{1}{2}\mathbb{E}\Big[\|\nabla\Phi(\theta^k)\|^2\Big]-\frac{1}{c_m b_0}\sum_{k=0}^{T}\frac{1}{2}\mathbb{E}\Big[\big\|\nabla_\theta\mathcal{L}(\theta^k,\pi^k)-\nabla\Phi(\theta^k)\big\|^2\Big]\le\frac{\Phi(\theta^0)-\mathbb{E}\,\Phi(\theta^{T+1})}{\gamma_\theta}$$
$$+3\gamma_\theta\kappa L\sum_{k=0}^{T-1}(1+C_A)\,C_D\Big(\beta_1^2\cdot 8\Big(K^2+\tfrac{\sigma^2}{B}\Big)+2\,\mathbb{E}\|\nabla\Phi(\theta^k)\|^2+4L^2\,\mathbb{E}[D_\psi(\pi^*(\theta^k),\pi^k)]+\tfrac{\sigma^2}{B}\Big).$$

Using smoothness of $\mathcal{L}$ and the definition of $\pi^*(\theta^k)$:

$$\frac{1-\beta_1}{2\sqrt{2K^2+2\frac{\sigma^2}{B}}}\sum_{k=0}^{T}\frac{1}{2}\mathbb{E}\Big[\|\nabla\Phi(\theta^k)\|^2\Big]-\frac{1}{c_m b_0}\sum_{k=0}^{T}L^2\mathbb{E}\Big[D_\psi(\pi^*(\theta^k),\pi^k)\Big]\le\frac{\Phi(\theta^0)-\mathbb{E}\,\Phi(\theta^{T+1})}{\gamma_\theta}$$
$$+3\gamma_\theta\kappa L\sum_{k=0}^{T-1}(1+C_A)\,C_D\Big(\beta_1^2\cdot 8\Big(K^2+\tfrac{\sigma^2}{B}\Big)+2\,\mathbb{E}\|\nabla\Phi(\theta^k)\|^2+4L^2\,\mathbb{E}[D_\psi(\pi^*(\theta^k),\pi^k)]+\tfrac{\sigma^2}{B}\Big).$$

Using

$$\gamma_\theta \;\le\; \frac{1-\beta_1}{72\,\kappa L(1+C_A)C_D\,\sqrt{2K^2+2\sigma^2/B}},$$

we have

$$\frac{1-\beta_1}{2\sqrt{2K^2+2\frac{\sigma^2}{B}}}\sum_{k=0}^{T}\tfrac{1}{3}\,\mathbb{E}\Big[\|\nabla\Phi(\theta^k)\|^2\Big] \le \left[\frac{7(1-\beta_1)}{2\sqrt{2K^2+2\frac{\sigma^2}{B}}}+\frac{1}{c_m b_0}\right]L^2\sum_{k=0}^{T}\mathbb{E}[D_\psi(\pi^*(\theta^k),\pi^k)]$$
$$+\frac{\Phi(\theta^0)-\mathbb{E}\,\Phi(\theta^{T+1})}{\gamma_\theta}+3\gamma_\theta\kappa L\sum_{k=0}^{T-1}(1+C_A)\,C_D\Big(\beta_1^2\cdot 8\Big(K^2+\tfrac{\sigma^2}{B}\Big)+\tfrac{\sigma^2}{B}\Big). \tag{33}$$

Simplifying our inequality we obtain:

$$\frac{1}{T+1}\sum_{k=0}^{T}\mathbb{E}\Big[\|\nabla\Phi(\theta^k)\|^2\Big] \;\le\; M_1\,\frac{1}{T+1}\sum_{k=0}^{T}\mathbb{E}\Big[D_\psi(\pi^*(\theta^k),\pi^k)\Big]+M_2\,\frac{\Phi(\theta^0)-\mathbb{E}\,\Phi(\theta^{T+1})}{(T+1)\gamma_\theta}+M_3\,\gamma_\theta,$$

where

$$M_1=\left[21+\frac{6\sqrt{2K^2+2\sigma^2/B}}{(1-\beta_1)}\frac{1}{c_m b_0}\right]L^2,$$
$$M_2=\frac{6\sqrt{2K^2+2\sigma^2/B}}{(1-\beta_1)},$$
$$M_3=\frac{18\,\kappa L\sqrt{2K^2+2\sigma^2/B}}{(1-\beta_1)}\,(1+C_A)\,C_D\Big(8\beta_1^2(K^2+\tfrac{\sigma^2}{B})+\tfrac{\sigma^2}{B}\Big).$$

Let us denote $\delta = 1-{}^{1}\!/_{128\kappa^2}$. Lemma 15 transforms into

$$\mathbb{E}[D_\psi(\pi^*(\theta^{t+1}),\pi^{t+1})] \le \Big(1-\tfrac{1}{128\kappa^2}\Big)\,\mathbb{E}[D_\psi(\pi^*(\theta^t),\pi^t)]$$
$$+\gamma_\theta^2\,C_\Phi\,\mathbb{E}\|\nabla\Phi(\theta^t)\|^2+\gamma_\theta^2\,C_B\,\tfrac{\sigma^2}{B}+\gamma_\theta^2\,\beta_1^2 C_\beta,$$

where the constants are

$$C_\Phi=\frac{2080\,\kappa^6}{c_m^2 b_0^2},\qquad C_B=\frac{1040\,\kappa^6}{c_m^2 b_0^2}+\frac{\lambda^2}{32L^4},\qquad C_\beta=\frac{8320\,\kappa^6}{c_m^2 b_0^2}\Big(K^2+\tfrac{\sigma^2}{B}\Big).$$

Hence, by unrolling the recursion, we obtain

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\,D_\psi(\pi^*(\theta^t),\pi^t) \;\le\; \frac{1}{T+1}\cdot\frac{1}{1-\delta}\,D_\psi(\pi^*(\theta^0),\pi^0)$$
$$+\;\frac{1}{1-\delta}\left(\gamma_\theta^2\,C_\Phi\,\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\|\nabla\Phi(\theta^t)\|^2+\gamma_\theta^2\,C_B\,\tfrac{\sigma^2}{B}+\gamma_\theta^2\,\beta_1^2 C_\beta\right).$$

Substituting the bound on the divergence into the main inequality, we obtain:

$$\begin{aligned}\frac{1}{T+1}\sum_{k=0}^{T}\mathbb{E}\Big[\|\nabla\Phi(\theta^k)\|^2\Big] \;\le M_1\Bigg[&\frac{1}{T+1}\cdot\frac{1}{1-\delta}\,D_\psi(\pi^*(\theta^0),\pi^0)\\ &+\;\frac{1}{1-\delta}\Big(\gamma_\theta^2\,C_\Phi\,\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\|\nabla\Phi(\theta^t)\|^2+\gamma_\theta^2\,C_B\,\tfrac{\sigma^2}{B}+\gamma_\theta^2\,\beta_1^2 C_\beta\Big)\Bigg]\\ &+M_2\,\frac{\Phi(\theta^0)-\mathbb{E}\,\Phi(\theta^{T+1})}{(T+1)\gamma_\theta}+M_3\,\gamma_\theta.\end{aligned}$$

Using $\gamma_\theta \;\le\; \sqrt{\frac{(1-\delta)}{2M_1C_\Phi}}$ we obtain

$$\begin{aligned}\frac{1}{T+1}\sum_{k=0}^{T}\mathbb{E}\Big[\|\nabla\Phi(\theta^k)\|^2\Big] \;\le 2M_1\Bigg[&\frac{1}{T+1}\cdot\frac{1}{1-\delta}\,D_\psi(\pi^*(\theta^0),\pi^0)\\ &+\;\frac{1}{1-\delta}\Big(\gamma_\theta^2\,C_B\,\tfrac{\sigma^2}{B}+\gamma_\theta^2\,\beta_1^2 C_\beta\Big)\Bigg]\\ &+2M_2\,\frac{\Phi(\theta^0)-\mathbb{E}\,\Phi(\theta^{T+1})}{(T+1)\gamma_\theta}+2M_3\,\gamma_\theta.\end{aligned}$$

Then, for step size

$$\gamma_\theta \;=\; \min\{\gamma_1,\gamma_2,\gamma_3\},$$

the averaged iterate satisfies

$$\mathbb{E}\,\|\nabla\Phi(\hat\theta_T)\|^2 \;\le\; \frac{A_1}{\gamma_\theta(T+1)}\,\Delta_\Phi \;+\; \gamma_\theta A_2\frac{\sigma^2}{B} \;+\; \frac{A_3}{T+1}D_0 \;+\; \beta_1^2 A_4, \tag{34}$$

where the constants are

$$\begin{aligned}A_1 &= \frac{12\sqrt{2K^2+2\sigma^2/B}}{1-\beta_1},\\ A_2 &= \frac{2M_1\gamma_\theta}{1-\delta}\,C_B+\frac{36\kappa L\sqrt{2K^2+2\sigma^2/B}}{1-\beta_1}\,(1+C_A)C_D,\\ A_3 &= \frac{2M_1}{1-\delta},\\ A_4 &= \left[\frac{288\kappa L\sqrt{2K^2+2\sigma^2/B}}{1-\beta_1}\,(1+C_A)C_D+4\right]\,(K^2+\tfrac{\sigma^2}{B}).\end{aligned}$$

Here

$$C_A=\frac{\beta_1}{c_m b_0(1-\beta_1)}\sqrt{2K^2+2\sigma^2/B},\qquad C_D=\frac{4}{c_m^2 b_0^2},$$

and

$$\gamma_1=\frac{1-\beta_1}{72\kappa L(1+C_A)C_D\sqrt{2K^2+2\sigma^2/B}},\quad \gamma_2=\frac{c_m b_0}{1048L\kappa^4},\quad \gamma_3=\sqrt{\frac{1-\delta}{2M_1C_\Phi}}.$$

We require each term in (34) to be at most $\varepsilon^2/4$. This gives

(i) From the Δ_Φ-term and the D_0-term:

$$T+1 \;\geq\; \max\left\{ \frac{4\Delta_\Phi}{\varepsilon^2} \max\left(\tfrac{A_1}{\gamma_1}, \tfrac{A_1}{\gamma_2}, \tfrac{A_1}{\gamma_3}\right),\; \frac{4A_3}{\varepsilon^2} D_0 \right\}.$$

(ii) From the variance term:

$$B \;\geq\; \frac{4\sigma^2}{\varepsilon^2}\; \min\Big(\gamma_1 A_2,\; \gamma_2 A_2,\; \gamma_3 A_2\Big).$$

(iii) From the momentum term:

$$\beta_1 \;\leq\; \sqrt{\frac{\varepsilon^2}{4A_4}}.$$

Then substituting $\delta = 1 - \frac{1}{128\kappa^2}$, $b_0 = L$, $c_m = \frac{1}{2}$ and with step size $\gamma_\theta = \mathcal{O}(1/\kappa^4)$ the averaged iterate satisfies

$$\mathbb{E}\,\|\nabla\Phi(\hat{\theta}_T)\|^2 \;\leq\; \frac{A_1}{\gamma_\theta(T+1)}\,\Delta_\Phi + \gamma_\theta A_2 \frac{\sigma^2}{B} + \frac{A_3}{T+1} D_0 + \beta_1^2 A_4,$$

where

$$A_1 = \mathcal{O}(K+\sigma), \quad A_2 = \mathcal{O}(\kappa^4), \quad A_3 = \mathcal{O}(\kappa^2 L^2), \quad A_4 = \mathcal{O}(\kappa^4).$$

Requiring each term in the bound to be at most $\varepsilon^2/4$ yields:

(i) Number of iterations:

$$T+1 \;\geq\; \max\left\{ \frac{\Delta_\Phi}{\varepsilon^2} \cdot \mathcal{O}\big(\kappa^4(K+\sigma)\big),\; \frac{D_0}{\varepsilon^2} \cdot \mathcal{O}\big(\kappa^2 L^2\big) \right\}.$$

(ii) Batch size:

$$B \;\geq\; \frac{\sigma^2}{\varepsilon^2} \cdot \mathcal{O}(1).$$

(iii) Momentum parameter:

$$\beta_1 \;\leq\; \frac{\varepsilon}{\mathcal{O}(\kappa^2)}.$$

This finishes the proof. ∎