This paper presents an insightful reinterpretation of mini-batch optimization—particularly in neural network training—under the lens of numerical operator splitting schemes. It identifies a key limitation of popular mini-batch optimizers: in non-interpolating settings, methods like Stochastic Gradient Descent can converge to stationary points or favor interpolating local minima introducing imbalances, due to the sequential nature of how mini-batch gradients are applied. To address this, the authors propose a family of well-balanced splitting schemes that dynamically correct for this imbalance, ensuring convergence to true critical points of the full objective, even under constant learning rates and without access to full gradients.

The proposed approach is motivated by insights from operator splitting theory and is empirically validated on a suite of 1D benchmarks designed to highlight the limitations of unbalanced updates in non-interpolating settings.

I recommend acceptance pending minor revisions.

Major Comments:

1. **Empirical Evaluation Limited to Determinisitic Settings**

   While the proposed balanced splitting schemes are empirically compelling, the paper would benefit from a more formal characterization of their convergence properties in the *stochastic* mini-batch setting. Specifically, how does the introduction of the correction term (e.g., c_n in the Speth-style scheme) interact with stochastic noise when using truly randomized mini-batches (as opposed to deterministic cyclic updates)?
   In practice, most training routines involve sampling batches with replacement or random reshuffling. It's unclear whether the balance mechanism remains effective under such stochasticity, especially in high-dimensional, non-convex landscapes. A theoretical analysis—or at least empirical investigation—of this point would strengthen the paper and clarify the robustness of the method under realistic training conditions.

2. **Simplified Empirical Evaluation.**

   While the proposed method is evaluated extensively across several synthetic 1D benchmarks, these settings—though illustrative—are quite limited in complexity and dimensionality. Since mini-batch optimization is primarily used in high

dimensional and highly non-convex problems (e.g., neural networks), it would be important to test whether the proposed balanced schemes scale to such settings. Can the authors provide empirical evidence (even preliminary) that their methods remain effective in higher-dimensional problems or small neural networks?

Alternatively, a discussion of computational bottlenecks, potential scalability issues, or future plans to test on real-world tasks would help clarify the practical applicability of the approach.

Minor Comments

Redundant Figures and Length:

a. Several empirical figures across Sections 2, 3, and 5 are quite repetitive, especially those showing stationary distributions for all learning rates and optimizers across multiple similar benchmark types. While they are useful for exhaustiveness, some of them might be moved to an appendix or summarized more concisely in the main text.
b. For example, Figures 1–4 and 13–17 contain overlapping insights that could be summarized in fewer plots with aggregate comparisons or referenced from supplementary material.

Potential to Improve Summary and Discussion:

a. The paper is quite lengthy, and while it is thorough, condensing some sections—particularly those reviewing existing optimizers and background—could improve readability. Streamlining parts of Sections 1.0.0.3 and 1.0.0.4, for instance, would help sharpen the focus on the core contributions and make the paper more accessible to the reader.

Other Comments:

- Typo in multiple places: “litterature” → “literature”.
- Clarify early on that the proposed methods allow use of a **constant learning rate** without requiring full gradient evaluations, as this is particularly relevant for the machine learning community.
- Please ensure the caption text in all figures is clear and that axis labels and legends are readable, particularly in the 1D plots.

- The github link attached to the paper appears to be not available. Please fix this.

Final thoughts:

The clarity of the operator splitting interpretation and the construction of the benchmarks are appreciated. This paper could generate interest in both the ML optimization and numerical analysis communities.