Fine-tuning with implicit loss

Łukasz Adamowicz

M2 Mathématiques, Modélisation et Apprentissage
Université Paris Cité
Internship at Hamprecht Lab, IWR Heidelberg

September 5, 2025

Outline

General Optimization Problem

- Our model: $E(\theta, M, p)$.
- $p_{\theta,i}$ are fixed points:

$$p_{\theta,i} = \arg \min_{p:\langle w_i, p\rangle = N_i} E(\theta, M_i, p).$$

- Loss: $L(\theta) = \sum_{i=1}^{n} L_i(p_{\theta,i}) = \sum_{i=1}^{n} \frac{1}{2}\|p_{\theta,i} - p_{gs,i}\|^2$.
- Keywords: bilevel optimization, stochastic bilevel optimization, Deep Equilibrium Models.

Single Molecule Optimization Problem

- For a single molecule M:
- Fixed point :

$$p_\theta = \arg \min_{p:\langle w,p\rangle = N} E(\theta, p)$$

.

- Loss: $L(\theta) = L(p_\theta) = \frac{1}{2}\|p_\theta - p_{gs}\|^2$.
- Goal: Minimize loss function $L(\theta)$.
- In bilevel optimization terms minimizing $L(p_\theta)$ is called an outer problem and minimizing $E(\theta, p)$ is an inner problem.

Jacobian Approach

- Formula for gradient of the loss is:

$$\frac{\partial L(p_\theta)}{\partial \theta} = -(p_\theta - p_{gs}) \cdot \left(\frac{\partial}{\partial p} \mathcal{P}\big(\frac{\partial}{\partial p} E(\theta, p)\big) \right)^{-1} \cdot \frac{\partial}{\partial \theta} \big(\mathcal{P} \frac{\partial}{\partial p} E(\theta, p)\big)$$

- $\mathcal{P}$ is the projection operator onto subspace $\langle w, p \rangle = 0$.
- Hessian $\left(\frac{\partial}{\partial p} [\mathcal{P}\big(\frac{\partial}{\partial p} E(\theta, p)\big)] \right)$ is **not** invertible, we utilize pseudoinverse instead.
- In practice, we solve the linear equation $y \cdot \left(\frac{\partial}{\partial p} [\mathcal{P} \frac{\partial}{\partial p} E(\theta, p)] \right) = -(p_\theta - p_{gs})$.
- This is done with matrix-free methods, because materializing the matrix is slow.

Jacobian Results

- Training is very unstable, depends on inner problem hyperparameters.
- Conjugate gradient method failed to solve linear equation
- Even when training loss decreased, it did not improve network metrics.
- Hessian has big both big (≈ 1000) and small (≈ 0.001) eigenvalues, plain gradient descent on $\|y \cdot \left(\frac{\partial}{\partial p}[\mathcal{P}\frac{\partial}{\partial p}E(\theta, p)]\right) + (p_\theta - p_{gs})\|^2$ fails, resorted to ADAM.
- Overall, Jacobian approach did not work.

Equilibrium Propagation

- Alternative gradient estimation method (see [7]).
- Define total energy:
  $T(\theta, p, \beta) = E(\theta, p) + \beta \frac{1}{2} \|p - p_{gs}\|^2 = E(\theta, p) + \beta L(p)$
- Define p_θ^β as the fixed point of T.

$$p_\theta^\beta = \arg \min_{p : \langle w, p \rangle = N} T(\theta, p, \beta)$$

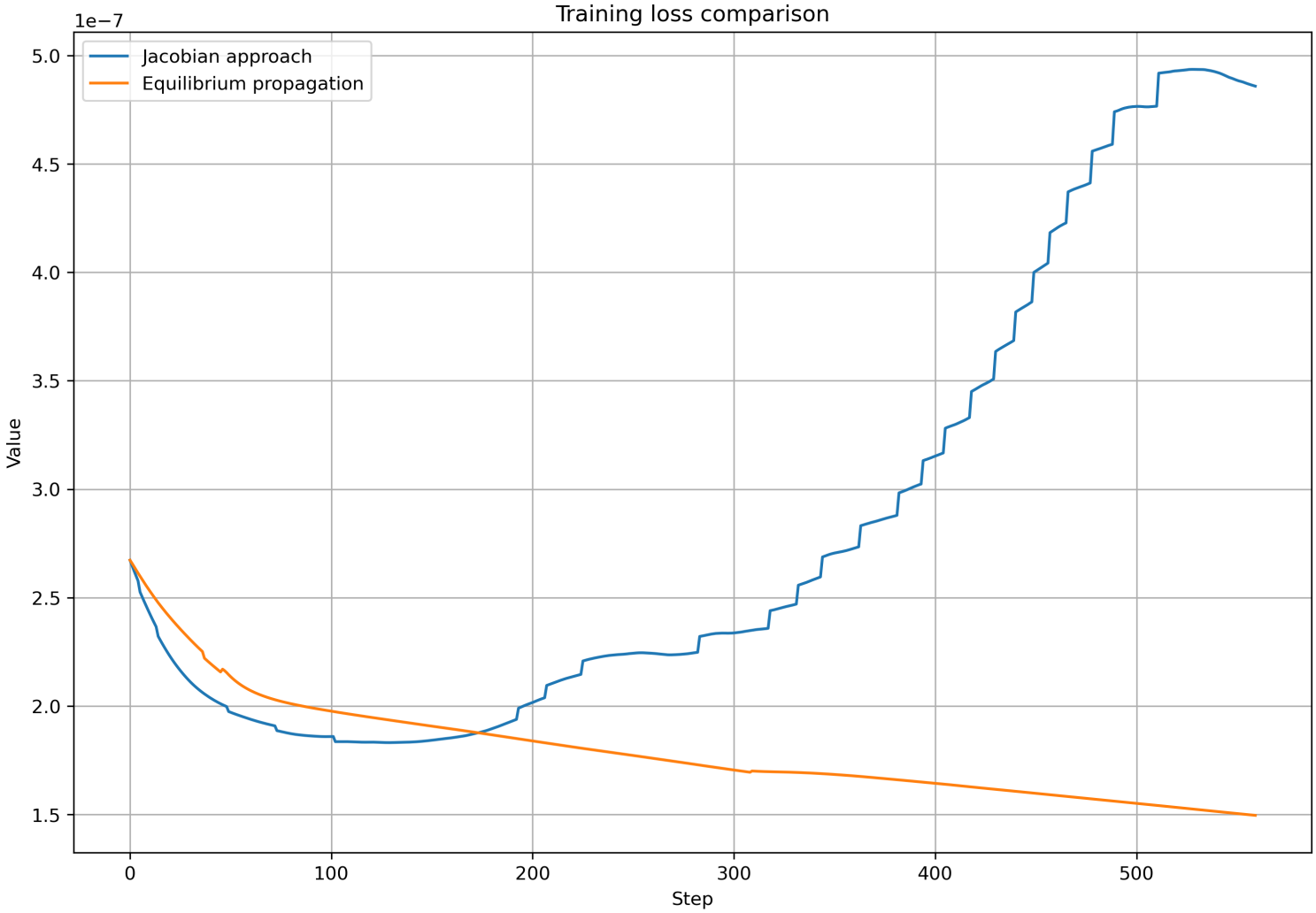
- Gradient formula:

$$\frac{d}{d\theta} L(p_\theta) = \lim_{\beta \to 0} \frac{1}{\beta} \Big[\partial_\theta T(\theta, p_\theta^\beta, \beta) - \partial_\theta T(\theta, p_\theta^0, 0) \Big].$$

- Approximate gradient as finite difference of right-hand side (numerical derivative)

Equilibrium Propagation Results

- Fine-tuned single molecule succesfully.
- Can work, when Jacobian approach fails
- Slower than Jacobian approach.
- Original contribution: implement in DFT setting, comparison with Jacobian (IFT) approach
- More details and derivation in the report.


Training loss comparison
1e−7
Jacobian approach
Equilibrium propagation
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
Value
0
100
200
300
400
500
Step

Stability Issues

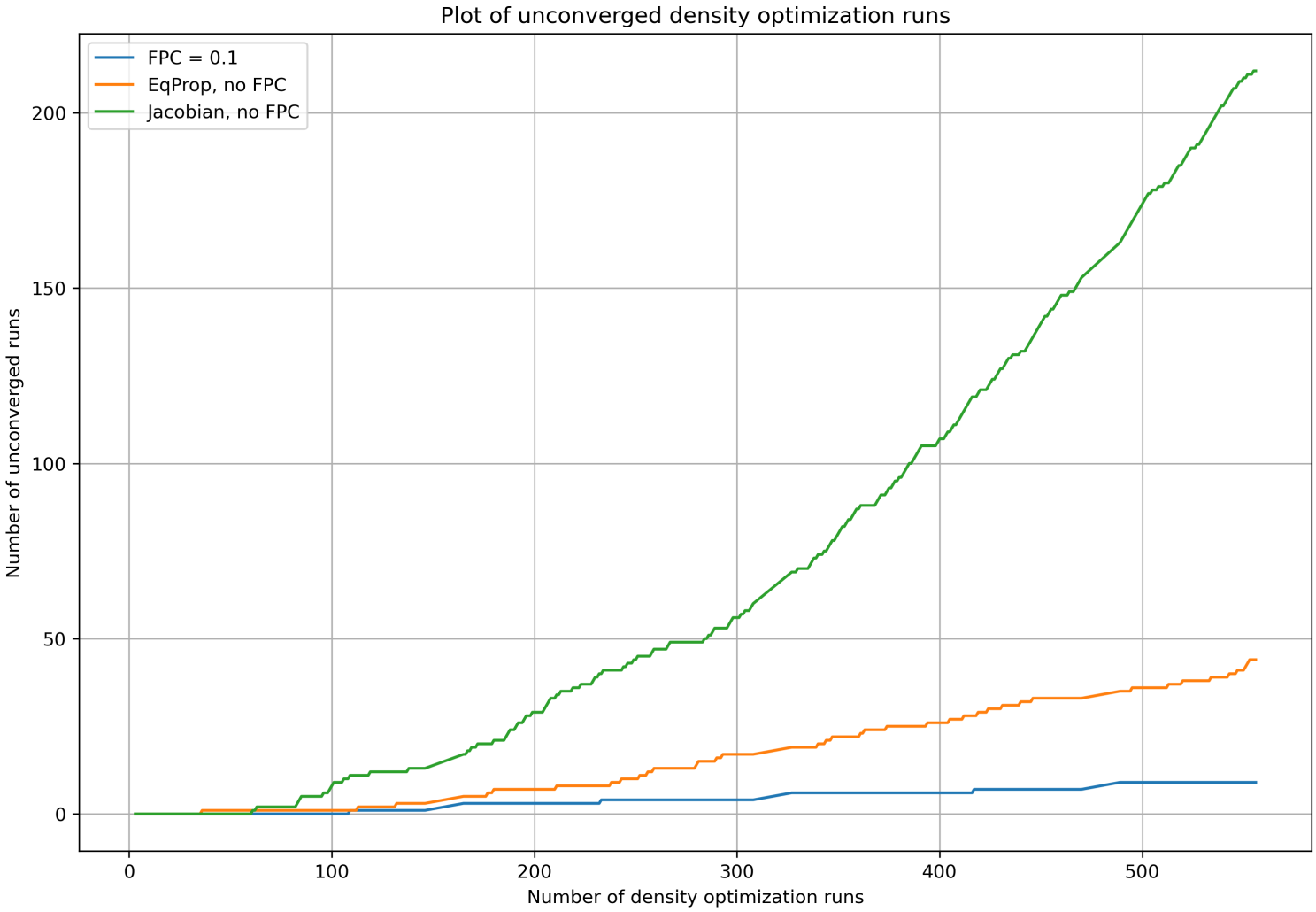
- Training stability deteriorates over time (known issue in DEQ models).
- Fixed point search takes longer and longer during training.
- There are techniques for alleviating the issue
- Appears both in Jacobian and EqProp approaches.
- Appears only, when trying to fine-tune on multiple molecules.

Fixed Point Correction

- Technique: include intermediate trajectory points in loss. Taken from [1].
- Inner problem trajectory is $p_1, \ldots, p_T \approx p_\theta$.
- Uniformly choose n intermediate points and modify the loss function as

$$L_{FPC}(\theta) = \sum_{k=1}^{n} \gamma^{n-k} L(p_{i_k}),$$

- Treat p_{i_k} as fixed points and calculate gradient using Jacobian approach.
- Helps stabilize training, but may reduce performance.
- Original contribution: Adapt to EqProp, add stochasticity.


Plot of unconverged density optimization runs
FPC = 0.1
EqProp, no FPC
Jacobian, no FPC
Number of unconverged runs
0
50
100
150
200
0
100
200
300
400
500
Number of density optimization runs

Conclusions

- Jacobian approach: not working right now, badly conditioned problem.
- Equilibrium propagation: works on single molecules.
- Biggest problem is training stability, additional treatment is needed.
- Training is very slow

End

Thank You for coming

References I

- Shaojie Bai, Zhengyang Geng, Yash Savani, and J Zico Kolter. Deep equilibrium optical flow estimation.
  In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 620–630, 2022.
- Shaojie Bai, Vladlen Koltun, and J Zico Kolter. Stabilizing equilibrium models by jacobian regularization.
  arXiv preprint arXiv:2106.14342, 2021.
- Andreas Burger, Luca Thiede, Alan Aspuru-Guzik, and Nandita Vijaykumar. DEQuify your force field: More efficient simulations using deep equilibrium models.
  In *AI for Accelerated Materials Design - ICLR 2025*, 2025.
- Asen L Dontchev and R Tyrrell Rockafellar. *Implicit functions and solution mappings*, volume 543.
  Springer, 2009.

References II

- Zhengyang Geng and J Zico Kolter. Torchdeq: A library for deep equilibrium models.
  arXiv preprint arXiv:2310.18605, 2023.
- Roman Remme, Tobias Kaczun, Tim Ebert, Christof A Gehrig, Dominik Geng, Gerrit Gerhartz, Marc K Ickler, Manuel V Klockow, Peter Lippmann, Johannes S Schmidt, et al. Stable and accurate orbital-free dft powered by machine learning.
  arXiv preprint arXiv:2503.00443, 2025.
- Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation.
  Frontiers in computational neuroscience, 11:24, 2017.
- Feitong Song and Ji Feng. Neuralscf: Neural network self-consistent fields for density functional theory.
  arXiv preprint arXiv:2406.15873, 2024.

References III

He Zhang, Siyuan Liu, Jiacheng You, Chang Liu, Shuxin Zheng, Ziheng Lu, Tong Wang, Nanning Zheng, and Bin Shao. Overcoming the barrier of orbital-free density functional theory for molecular systems using deep learning.
Nature Computational Science, 4(3):210–223, 2024.

Nicolas Zucchet and Joao Sacramento. Beyond backpropagation: bilevel optimization through implicit differentiation and equilibrium propagation.
Neural Computation, 34(12):2309–2346, 2022.

Jacobian Gradient Derivation - Part 1

- Starting from loss function: $L(\theta) = \frac{1}{2}\|p_\theta - p_{gs}\|^2$.
- Using chain rule: $\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial p_\theta} \cdot \frac{\partial p_\theta}{\partial \theta}$.
- First term is straightforward: $\frac{\partial L}{\partial p_\theta} = (p_\theta - p_{gs})$.
- For second term, we need the implicit function theorem.

Jacobian Gradient Derivation - Part 2

- At the fixed point, we have: $\mathcal{P}\nabla_p E(\theta, p_\theta) = 0$.
- Differentiating this constraint with respect to θ:

$$\frac{\partial}{\partial\theta}\left[\mathcal{P}_{\langle w,p\rangle=N}\nabla_p E(\theta, p_\theta)\right] = 0 \quad (1)$$

$$\frac{\partial}{\partial p}\mathcal{P}_{\langle w,p\rangle=N}\nabla_p E(\theta, p_\theta)\cdot\frac{\partial p_\theta}{\partial\theta} + \frac{\partial}{\partial\theta}\mathcal{P}_{\langle w,p\rangle=N}\nabla_p E(\theta, p_\theta) = 0 \quad (2)$$

- Solving for $\frac{\partial p_\theta}{\partial\theta}$:

$$\frac{\partial p_\theta}{\partial\theta} = -\left(\frac{\partial}{\partial p}\mathcal{P}_{\langle w,p\rangle=N}\nabla_p E(\theta, p_\theta)\right)^{-1}\cdot\frac{\partial}{\partial\theta}\mathcal{P}_{\langle w,p\rangle=N}\nabla_p E(\theta, p_\theta) \quad (3)$$

Jacobian Gradient Derivation - Part 3

- Substituting back into our chain rule formula:

$$\frac{\partial L}{\partial \theta} = (p_\theta - p_{gs}) \cdot \frac{\partial p_\theta}{\partial \theta} \tag{4}$$

$$= -(p_\theta - p_{gs}) \cdot \left(\frac{\partial}{\partial p} \mathcal{P} \nabla_p E(\theta, p_\theta) \right)^{-1} \cdot \frac{\partial}{\partial \theta} \mathcal{P} \nabla_p E(\theta, p_\theta) \tag{5}$$

Equilibrium Propagation Derivation - Part 1

- Define perturbed energy function: $T(\theta, p, \beta) = E(\theta, p) + \beta L(p)$.
- Define p_θ^β as the fixed point of this perturbed energy:

$$p_\theta^\beta = \arg \min_{p:\langle w,p\rangle = N} T(\theta, p, \beta)$$

- Note that $p_\theta^0 = p_\theta$ (the original fixed point).
- Our goal: compute $\frac{d}{d\theta} L(p_\theta)$.

Equilibrium Propagation Derivation - Part 2

- Define function $G(\theta, \beta) = T(\theta, p_\theta^\beta, \beta)$.
- There's symmetry of second derivatives $\frac{d}{d\beta}\frac{d}{d\theta}$ at $\beta = 0, \theta = \theta$

Equilibrium Propagation Derivation - Part 3

- $\frac{dG}{d\beta} = \frac{\partial T}{\partial \beta} + \frac{\partial T}{\partial p}\frac{dp_\theta^\beta}{d\beta}$, but the second term vanishes at $p = p_\theta^\beta$.
- $\frac{\partial T}{\partial \beta}\Big|_{\beta=0} = L(p_\theta)$