# Fine-tuning with implicit loss

Lukasz Adamowicz

August 28, 2025

# Outline

# General Optimization Problem

- Energy model: $E(\theta, M, p)$.
- Ground state density coefficients $p_\theta$ are fixed points:

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

- Loss: $L(\theta) = \sum_{i=1}^{n} L_i(p_{\theta,i}) = \sum_{i=1}^{n} \frac{1}{2}\|p_{\theta,i} - p_{gs,i}\|^2$.
- Bilevel optimization problem across multiple molecules.
- Challenge: compute gradient of $L(\theta)$ with respect to model parameters.

# Single Molecule Optimization Problem

- For a single molecule $M$:
- Fixed point equation: $p_\theta = \arg\min_{p:\langle w,p\rangle = N} E(\theta, p)$.
- Loss: $L(\theta) = L(p_\theta) = \frac{1}{2}\|p_\theta - p_{gs}\|^2$.
- Goal: Find $\theta^* = \arg\min_\theta L(\theta)$.
- Need to compute $\frac{dL}{d\theta}$ without direct differentiation through the fixed point finding process.

# Jacobian Approach

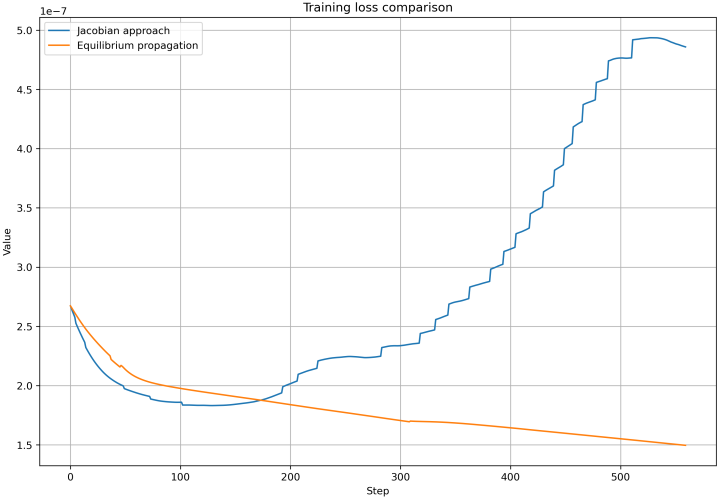
- Based on implicit function theorem.
- Gradient formula:

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

- $\mathcal{P}$ is the projection operator onto subspace $\langle w, p \rangle = N$.
- we solve for $y = -(p_\theta - p_{gs}) \cdot \left( \frac{\partial}{\partial p} \mathcal{P} \nabla_p E(\theta, p) \right)^{-1}$
- Memory and stability issues.

# Jacobian Results

- Training loss sometimes decreases, but diverges later.
- Conjugate gradient method failed.
- Did not improve density difference.
- Jacobian has big spread of eigenvalues values


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

# Equilibrium Propagation

- Alternative gradient estimation method.
- Define total energy: $T(\theta, p, \beta) = E(\theta, p) + \beta L(p)$.
- Define $p_\theta^\beta$ as the fixed point:

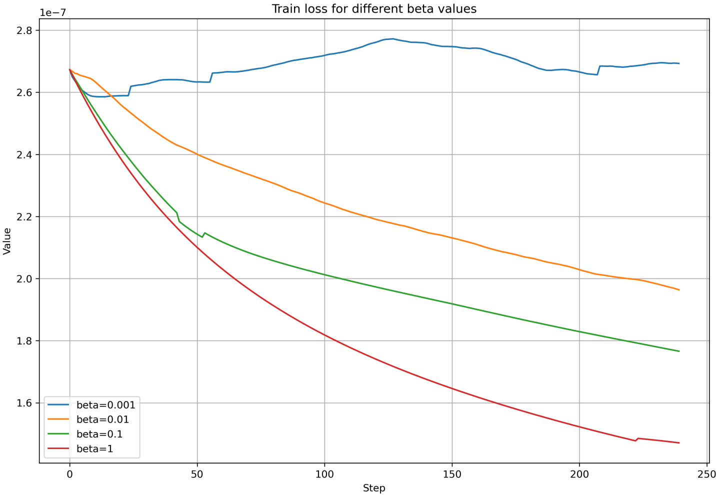
$$p_\theta^\beta = \arg \min_{p:\langle w,p\rangle = N} T(\theta, p, \beta) = \arg \min_{p:\langle w,p\rangle = N} [E(\theta, p) + \beta L(p)]$$

- Gradient formula:

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

# Equilibrium Propagation Results

- Worked for fine-tuning on single molecule.
- Loss decreased consistently.
- Density difference decreased, but energy difference increases

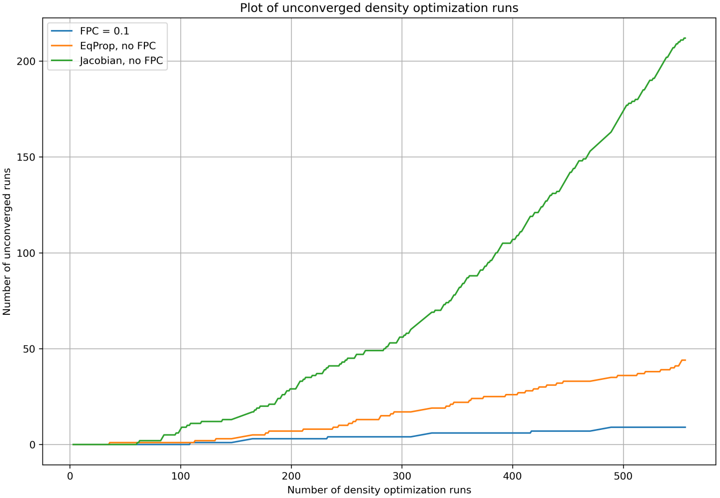

Train loss for different beta values
1e−7
2.8
2.6
2.4
2.2
2.0
1.8
1.6
Value
0
50
100
150
200
250
Step
beta=0.001
beta=0.01
beta=0.1
beta=1

# 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

# Fixed Point Correction (Original Contribution)

- Technique: include intermediate trajectory points in loss.
- Loss function:
$$L_{FPC}(\theta) = \sum_k \gamma^{n-k} L(p_{i_k}),$$
where $p_{i_k}$ are intermediate points.
- Helps stabilize training, but may reduce performance.
- I modified it to be random


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

# Conclusions

- Jacobian approach: not effective for now, would be nice if it works
- Equilibrium propagation: potentially viable
- Training is very unstable and very slow

# End

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

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

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

- 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:

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

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

- The term $\frac{\partial}{\partial p} \mathcal{P}_{\langle w,p \rangle = N} \nabla_p E(\theta, p_\theta)$ is the projected Hessian matrix of the energy function.

# Equilibrium Propagation Derivation - Part 1

- Define perturbed energy function: $T(\theta, p, \beta) = E(\theta, p) + \beta L(p)$.
- Define $p_\theta^\beta$ 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 $G(\theta, \beta) = L(p_\theta^\beta)$.
- 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 $p = p_\theta^\beta$.
- $\frac{\partial T}{\partial \beta}\Big|_{\beta=0} = L(p_\theta)$