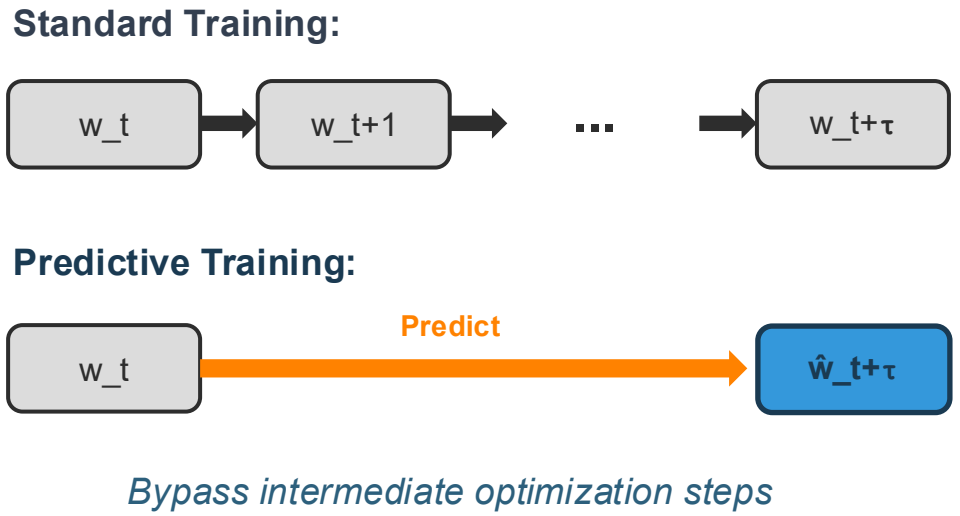
Predictive Differential Training Guided by Training Dynamics

Accelerating training by selectively applying high-fidelity predicted weights

Fanqi Wang[1]*, Weisheng Tang[1]*, Landon Harris[1], Hairong Qi[1], Dan Wilson[1], Igor Mezić[2]
[1] University of Tennessee, Knoxville [2] University of California, Santa Barbara

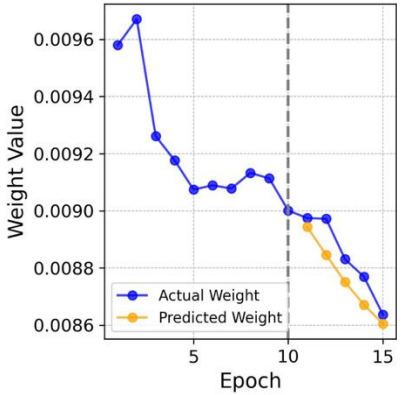
Why Predictive Acceleration for Training?

- Deep neural network training remains fundamentally iterative.
- At scale, repeated gradient computations and weight updates become increasingly expensive.
- This raises a natural question: can some intermediate optimization steps be bypassed by predicting future weights?

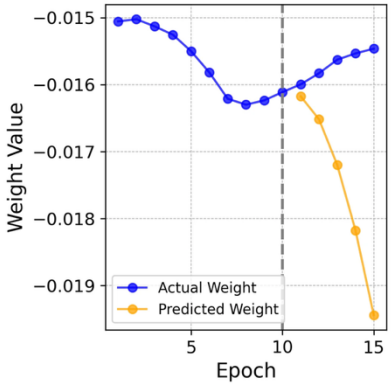


Why Naive Future-Weight Prediction Is Hard

- Training dynamics are highly nonlinear, non-stationary, and stochastic.
- In multi-step prediction, even small errors can accumulate and destabilize optimization.
- Prediction quality is highly heterogeneous across parameters, so not all predicted weights are equally reliable.



Easy to predict



Hard to predict

Despite these challenges, short-horizon prediction remains plausible when training trajectories exhibit exploitable temporal structure or dominant low-dimensional dynamics.

Prior Attempts and the Missing Piece

This has motivated two main directions for future-weight prediction.

Learned Predictors

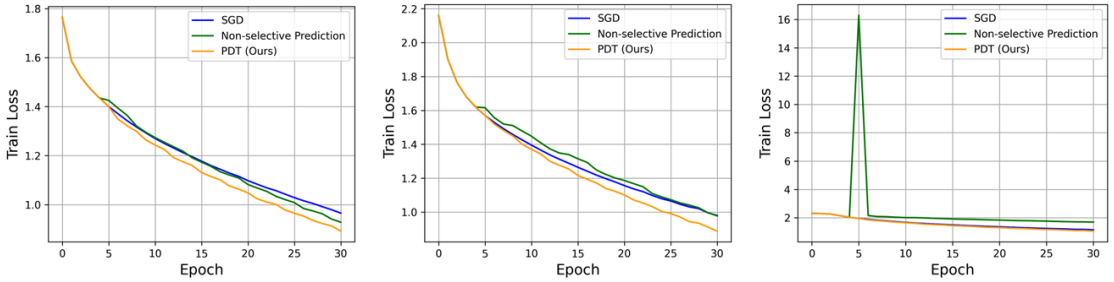
- auxiliary predictor
- nonlinear / expressive
- extra training & inference
- transfer-oriented

DMD-based Prediction

- recent training snapshots
- linear spectral model
- no external predictor
- adapts to current trajectory

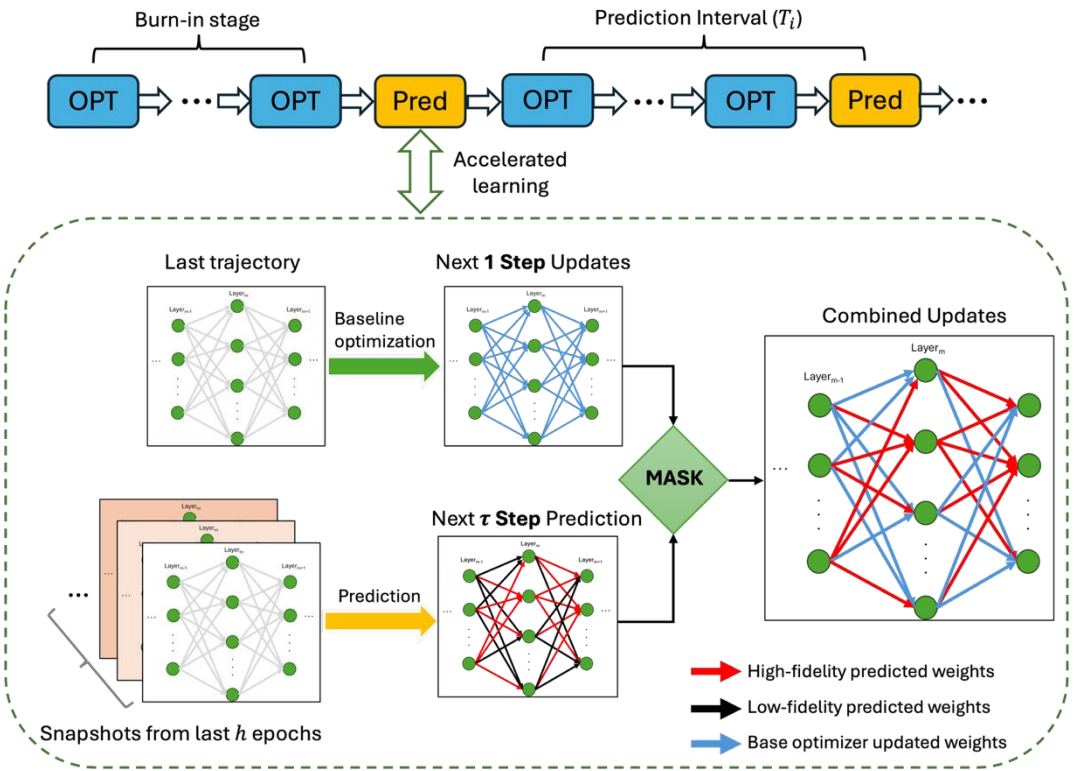
What remains unresolved?

The missing piece is a **reliability-aware** way to apply prediction at scale.



Prior predictive methods do not explicitly account for heterogeneous prediction reliability across parameters.

Our Key Idea: Predict Globally, Trust Selectively



1. Burn in
Collect recent weight snapshots under the baseline optimizer

2. Predict
Forecast future weights from the recent training trajectory

3. Mask
Keep only the high-fidelity predicted components

4. Combine
Let the remaining components follow the baseline optimizer

Prediction Module: Global Forecast from Training Dynamics

Training as a dynamical system

We view training as a discrete-time dynamical system in weight space

$$\mathbf{w}_{i+1} = T(\mathbf{w}_i)$$

Data-driven approximation from recent snapshots

Using recent weight snapshots $\boldsymbol{W}_i = [\mathbf{w}_{i-h+1}, \ldots, \mathbf{w}_i]$, DMD approximates the dominant global training dynamics and yields DMD modes Φ and eigenvalues Λ.

Direct multi-step prediction

PDT then predicts a future weight state by spectral extrapolation:

$$\mathbf{w}_{i+\tau}^{\mathbf{pred}} = \Phi \Lambda^{\tau} \Phi^{\dagger} \mathbf{w}_i$$

This gives a direct multi-step forecast from the recent training trajectory.

Masking Strategy: Selective High-Fidelity Prediction

Global prediction is useful, but not all predicted components are equally reliable. PDT therefore applies prediction selectively.

Criterion I — Acceleration Effectiveness

- **Advance further than a single optimizer step**
- Remain **scale-bounded for stability**

$$\|\mathbf{w}_{i+1}^{opt} - \mathbf{w}_{i}^{opt}\| < \|\mathbf{w}_{i+\tau}^{pred} - \mathbf{w}_{i}^{opt}\| \leq \tau\|\mathbf{w}_{i+1}^{opt} - \mathbf{w}_{i}^{opt}\|$$

Criterion II — Dynamic Consistency

- Predicted evolution must **align with local gradient direction**:

$$\mathrm{sign}(\mathbf{w}_{i+k,j}^{pred} - \mathbf{w}_{i+k-1,j}^{pred}) = \mathrm{sign}(\mathbf{w}_{i+1,j}^{opt} - \mathbf{w}_{i,j}^{opt})$$

Only components satisfying both criteria are accepted; the rest continue to follow the baseline optimizer, helping preserve a conservative descent behavior.

Main Results: Faster Time to Baseline Best Loss

Across architectures and optimizers, PDT consistently reaches the baseline's best loss faster while maintaining competitive accuracy.

Runtime Reduction to Reach Baseline's Best Loss

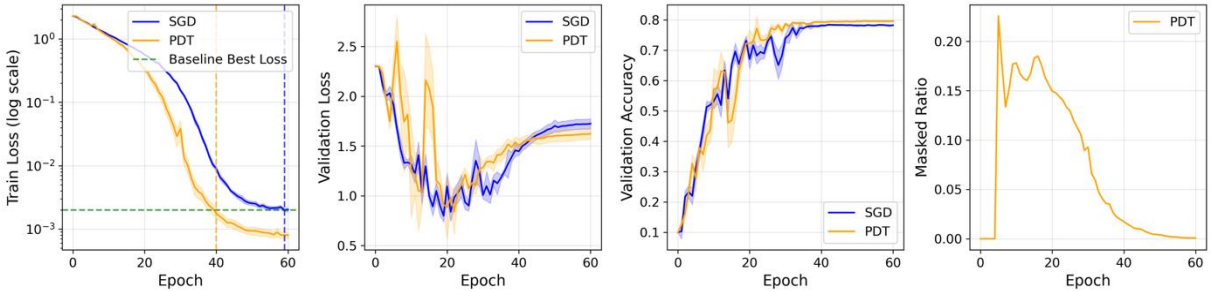
Model	Baseline Optimizer	TTB-Loss (s)		TTB-Acc (s)		Runtime Reduction (%)	
		Baseline	PDT	Baseline	PDT	Train Loss	Val. Acc.
FCN	SGD	2174.32	**1313.52**	2088.58	**1424.14**	**39.59**	**31.81**
AlexNet	SGD	683.93	**430.91**	531.30	**347.11**	**37.00**	**34.67**
ResNet-50	SGD-M	110063.72	**88752.33**	121449.60	**92133.34**	**19.36**	**24.14**
ViT-Base	AdamW	259241.21	**232810.62**	296028.36	**243097.58**	**10.20**	**17.88**
ViT-Huge	AdamW	725564.86	**653854.05**	741220.54	**660711.80**	**9.88**	**10.86**

All reported runtimes include prediction and masking overhead.

- Runtime reduction is typically in the 10%– 40% range
- The benefit appears from FCN/AlexNet to ResNet-50 / ViT

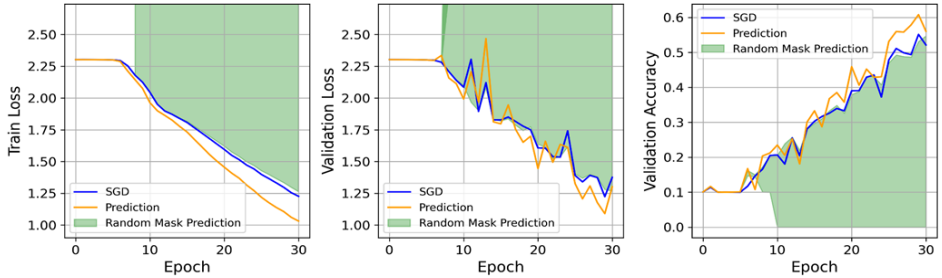
Representative Training Dynamics and Why the Mask Matters

Representative training dynamics on AlexNet show that PDT reaches lower train loss earlier while maintaining competitive test performance.



Why the Mask Matters

Random masking does not reproduce PDT and can destabilize training.



Takeaways

- PDT accelerates training by predicting future weights selectively rather than uniformly.
- The mask is the key to stability: only high-fidelity predicted components are accepted.
- PDT is a lightweight plug-in framework that consistently reduces time to the baseline’s best loss.