Generative Modeling Reinvents Supervised Learning:
Label Repurposing with Predictive Consistency Learning

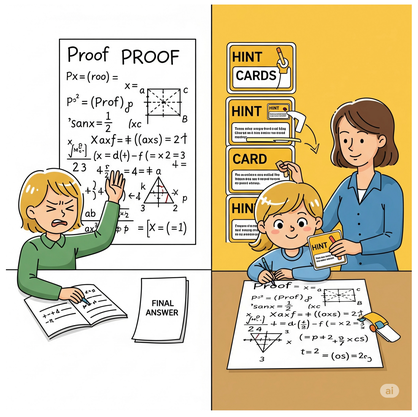
Yang Li[1,2], Jiale Ma[1,2], Yebing Yang[1], Qitian Wu[3], Hongyuan Zha[4], Junchi Yan[1,2]

[1]Shanghai Jiao Tong University [2]Shanghai Innovation Institute [3]Broad Institute of MIT and Harvard [4]The Chinese University of Hong Kong, Shenzhen

TL;NR: Labels shouldn't just be for checking against correct answers; they should more likely act as a helpful reference during the learning process

Background and Motivation

- Traditionally, supervised learning models directly predict labels as "standard answers." **The assumption was that labels were simpler than input data, focusing model design on input feature extraction.**
- However, with increasingly complex labels in deep learning, we reconsider: **Can labels, when information-rich, become learning aids rather than just targets?**



The figure is generated by Gemini.

Analogy: Learning Math Proofs

- **Traditional supervised learning** is like giving the student a theorem, letting them attempt the entire proof independently, and then just comparing their final answer to the complete solution to adjust their strategy.
- However, when the solution contains rich information, this isn't the best way to learn.
- **Humans usually refer to the full solution to grasp the thought process for complex problems**.
- **When stuck, we need step-by-step hints and guidance, gradually reducing these prompts until we can solve it independently.**

Can't make it to the conference in person. Please feel free to reach out to me via LinkedIn, WeChat, or email.

Linkedin WeChat Email:yanglily@sjtu.edu.cn

Our Solution: Predictive Consistency Learning

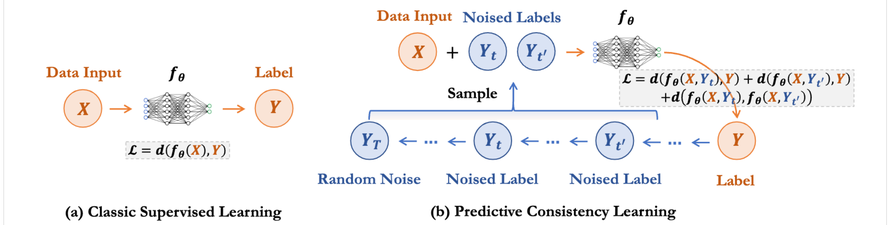


Figure 1. Illustration of predictive consistency learning (PCL). Unlike traditional approaches that predict labels directly from inputs, PCL predicts labels using inputs and noise-perturbed label hints and pursues predictive consistency across different noise steps.

- In traditional supervised learning, $\mathscr{L}_{SL} = d(f_\theta(\mathbf{x}), \mathbf{y})$.
- Progressive Consistency Learning (PCL), inspired by consistency models, is a new supervised learning paradigm. It tackles complex labels by progressively decomposing label information. Instead of just predicting a final answer, PCL uses noisy labels as hints $\mathbf{Y}_t$ **(obtained by same processes like discrete and continuous diffusion models)** at different time steps t to guide the model. PCL trains by:
  - **Mapping noisy labels back to the true label $\mathbf{y}$, conditioned on $\mathbf{x}$.**
  - **Ensuring predictions from different noisy hints consistently approximate $\mathbf{y}$.**
- The model samples two time steps t, t' and aims for both accurate prediction to $\mathbf{y}$ and consistency between the two predictions. This **cross-noise-level consistency** helps the model learn robust representations and reduces reliance on perfect label hints. The PCL loss function is:

$$\mathscr{L}_{\text{PCL}}(\theta) = \mathbb{E}\left[\lambda_1 d\big(f_\theta(\mathbf{x}, \mathbf{y}_t, t), \mathbf{y}\big) + \lambda_1 d\big(f_\theta(\mathbf{x}, \mathbf{y}_{t'}, t'), \mathbf{y}\big)\right) + \lambda_2 d\big(f_\theta(\mathbf{x}, \mathbf{y}_t, t), f_\theta(\mathbf{x}, \mathbf{y}_{t'}, t')\big)\right]$$

Multistep Inference with Consistency Mappings

Algorithm 1 Predictive Consistency Training

Input: Dataset $\mathcal{D}$, model f_θ, noise function $q(\cdot)$, learning rate η, loss weights λ_1, λ_2, noise gap α
repeat
  Sample $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$, and $t_1 \sim \mathrm{U}[1, T]$, $t_2 = \alpha t_1$
  Sample $\mathbf{y}_{t_1} \sim q(\mathbf{y}_{t_1}|\mathbf{y})$, $\mathbf{y}_{t_2} \sim q(\mathbf{y}_{t_2}|\mathbf{y})$
  $\hat{\mathbf{y}}_0^{t_1} \leftarrow f_\theta(\mathbf{x}, \mathbf{y}_{t_1}, t_1)$
  $\hat{\mathbf{y}}_0^{t_2} \leftarrow f_\theta(\mathbf{x}, \mathbf{y}_{t_2}, t_2)$
  $\mathcal{L} \leftarrow \lambda_1 d(\hat{\mathbf{y}}_0^{t_1}, \mathbf{y}) + \lambda_1 d(\hat{\mathbf{y}}_0^{t_2}, \mathbf{y}) + \lambda_2 d(\hat{\mathbf{y}}_0^{t_1}, \hat{\mathbf{y}}_0^{t_2})$
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
until convergence

Algorithm 2 Multistep Prediction

Input: model f_θ, data input $\mathbf{x}$, noise function $q(\cdot)$, time steps $\tau_1 > \tau_2 > \cdots > \tau_{N_i - 1}$, preset inference step N_i

Sample random noise $\mathbf{y}_T$
$\hat{\mathbf{y}}_0 \leftarrow f_\theta(\mathbf{x}, \mathbf{y}_T, T)$
for $n = 1$ to $N_i - 1$ **do**
  Sample $\mathbf{y}_{\tau_n} \sim q(\mathbf{y}_{\tau_n}|\hat{\mathbf{y}}_0)$
  $\hat{\mathbf{y}}_0 \leftarrow f_\theta(\mathbf{x}, \mathbf{y}_{\tau_n}, \tau_n)$
end for
Output: Prediction $\hat{\mathbf{y}}_0$

- **Single-Step Inference:** PCL can achieve efficient predictions with a single forward pass using a randomly sampled noisy label $\mathbf{y}_T$ as a hint. Although this hint contains no information (making it similar to traditional direct prediction), PCL's improved training leads to superior prediction accuracy compared to traditional supervised learning.
- **Multi-Step Inference:** While initial predictions are more accurate with less noise (smaller t), PCL aims to transfer this high accuracy to higher noise levels. Ideally, a single step could suffice, but practically, gradually decreasing t from T to 0 enhances accuracy by refining label information at different granularities.
- The multi-step inference process uses the prediction from $\mathbf{y}_T$ as a hint for the next step's label. In each subsequent step, the noise is gradually reduced, leading to increasingly precise predictions.

Experiments

Table 3. Results on Semantic Segmentation.

Backbone Model	Training	Pixel Acc.↑	mIoU↑	Score↑
MobileNetV2dilated	SL	75.53	33.13	54.33
	PCL	**76.42**	**35.27**	**55.85**
ResNet50dilated+PPM	SL	78.98	41.49	60.24
	PCL	**81.32**	**46.93**	**64.13**

• Vision: Semantic Segmentation

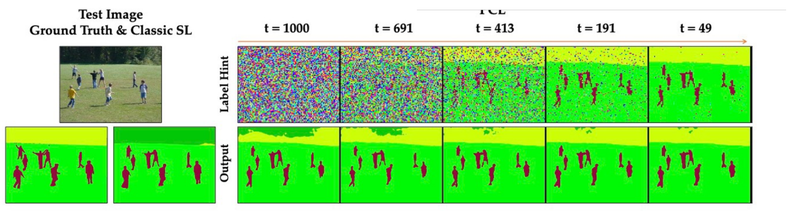


Figure 4. Predictions across varying timesteps based on the last step's predictions in the multistep inference procedure. In each step, the model receives the input and label hint and predicts the output.

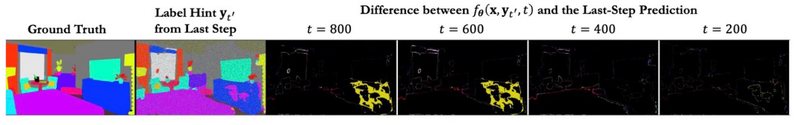


Figure 5. The visualization of the prediction improvements with different t controlling the prediction granularity.

• Graph: N-Body Simulation

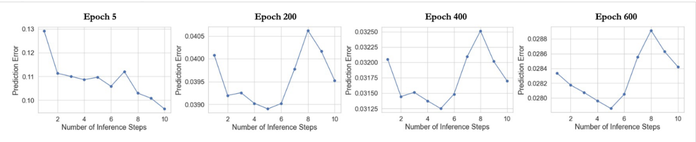


Figure 3. The influence variation of inference steps across the training process.

Table 1. Prediction error ($\times 10^{-2}$) of SL and PCL on top of graph models on various types of N-body simulation systems. The header of each column "p, s, h" denotes the scenario with p isolated particles, s sticks and h hinges.

Backbone Model	Training	1, 2, 0	2, 0, 1	3, 2, 1	0, 10, 0	5, 3, 3
GCN	SL	2.865±0.021	2.534±0.061	3.479±0.110	4.705±0.046	4.303±0.002
	PCL	**2.436±0.040**	**2.268±0.012**	**2.795±0.061**	**3.162±0.251**	**3.228±0.221**
GAT	SL	2.921±0.198	2.707±0.024	3.351±0.111	3.478±0.342	3.407±0.180
	PCL	**2.771±0.208**	**2.581±0.026**	**2.802±0.216**	**2.481±0.059**	**2.534±0.011**
GGNN	SL	3.013±0.022	2.716±0.068	3.293±0.023	4.426±0.044	4.148±0.035
	PCL	**2.614±0.031**	**2.297±0.033**	**2.974±0.011**	**3.191±0.290**	**3.457±0.213**

• Text: Supervised Fine-Tuning of LLMs

Table 4. Evaluation on LLM fine-tuning. Relative performance improvements compared to the raw model are marked in brackets.

Backbone	Training	MMLU	CRASS	BBH
Raw Model	–	41.90	37.59	32.93
Full FT	SL	46.22	58.29	33.38
	PCL	**47.10**	**59.48**	**34.75**