# Progressively Knowledge Distillation via Re-parameterizing Diffusion Reverse Process

**Xufeng Yao** Fanbin Lu Yuechen Zhang Xinyun Zhang Wenqian Zhao Bei Yu

The Chinese University of Hong Kong

## Background and Motivation

Knowledge distillation aims at transferring knowledge from the teacher model to the student one by aligning their distributions. Feature-level distillation often uses $\mathcal{L}_2$ distance or its variants as the loss function, based on the assumption that outputs follow normal distributions.

### Insights behind Loss Function

$$\mathcal{L}_{trans} = -\log p(\boldsymbol{x}^T|\boldsymbol{x}^S) \propto \log \hat{\boldsymbol{\sigma}} + \frac{(\boldsymbol{x}^T - \hat{\boldsymbol{\mu}})^2}{2\hat{\boldsymbol{\sigma}}^2}. \tag{1}$$

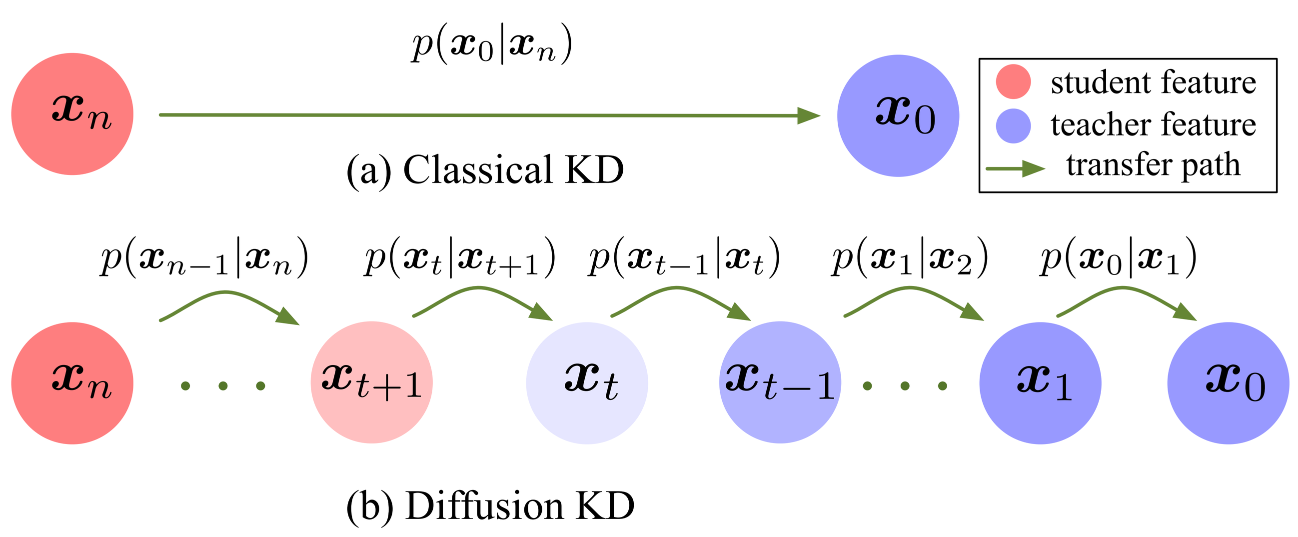
In the standard $\mathcal{L}_2$ loss paradigm, variance is treated as a constant value. This assumption may pose a significant challenge when confronting large distribution gaps.

### Experimental Observations

| | | | |
|---|---|---|---|
| Teacher | Swin | Swin | Swin |
| | 94.48% | 94.48% | 94.48% |
| Student | MobileNetV2 | ResNet18 | ShuffleNetV2 |
| | 84.04% | 84.42% | 76.86% |
| CRD | 83.72% | 84.26% | 77.88% |
| | -0.32 | -0.16 | +1.02 |

## Key Idea

Progessively transfer knowledge!




(a) Classical KD

(b) Diffusion KD

- Decompose the transfer objective into small parts
- Map student features to teachers features step by step.
- Leverage diffusion theories.

## Problem and Solutions

However, directly using diffusion models is impractical.

- Problem: How to map student to teacher features in diffusion manner.
- Problem: How to generate multiple features without extra inference cost.
- Solution: Mapping student to teacher features via diffusion reverse process manner.
- Solution: Utilize structral-reparameterization techinicals.

## Problem Formulation

### General Formulation of Transfer Learning

We define $P$ and $Q$ are corresponding distributions, then the conventional KL divergence between teacher and student distributions can be defined as :

$$\mathrm{KL}(P||Q) = \sum_{\boldsymbol{x}} p(\boldsymbol{x}^T)\log(\frac{p(\boldsymbol{x}^T)}{q(\boldsymbol{x}^S)}), \tag{2}$$

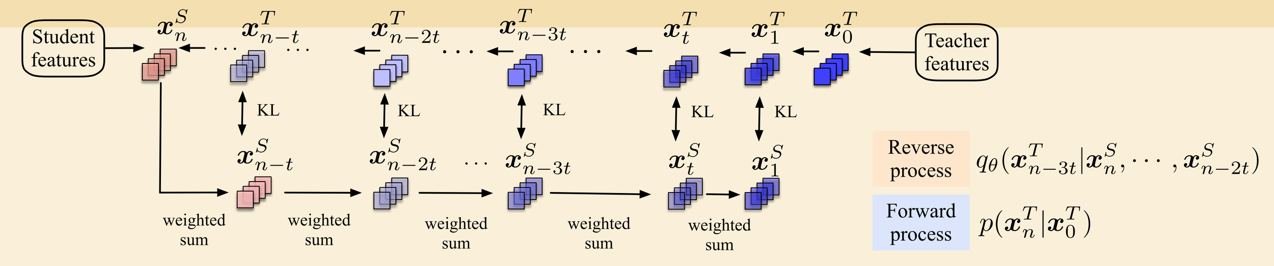
With regard to the maximum likelihood estimation approach, the transfer objective can be defined as $-\log(q_\theta(\boldsymbol{x}^T|\boldsymbol{x}^S))$, the transfer objective can be reformulated as:

$$-\log\left(q_\theta(\boldsymbol{x}_0^T|\boldsymbol{x}_1^T)\cdots q_\theta(\boldsymbol{x}_{t-1}^T|\boldsymbol{x}_t^T)\cdots q_\theta(\boldsymbol{x}_{n-1}^T|\boldsymbol{x}_n^S)\right), \tag{3}$$

Instead of directly predicting $\boldsymbol{x}_0^T$ by $\boldsymbol{x}_n^S$, which may lead to negative transfer, we can optimize the intermediate steps (e.g., $-\log q_\theta(\boldsymbol{x}_{t-1}^T|\boldsymbol{x}_t^T)$) and safely transfer the knowledge.

## KDiffusion

### Whole framework




Proposed knowledge transfer via re-parameterizing diffusion reverse progress.

## Structural Re-parameterization

- Problem: How to generate multiple features without extra inference cost.
- Solution: Utilize structral-reparameterization techinicals.

Structural re-parameterization leverages the linear properties of a set of linear modules $f_0, f_1, \cdots, f_n$ which can produce diverse outputs with a common input. The combination of these modules can be expressed as follows:

$$\alpha_1 f_0(x) + \cdots + \alpha_n f_n(x) = (\alpha_1 f_0 + \cdots + \alpha_n f_n)(x). \tag{4}$$

## Contructing the Diffusion Forward Process

We follow the classicial setting [1]. We can obtain the probability distributions of each intermediate features $\boldsymbol{x}_t^T$ by:

$$q(\boldsymbol{x}_t^T|\boldsymbol{x}_0^T) := \mathcal{N}(\boldsymbol{x}_t^T; \hat{\alpha}_t\boldsymbol{x}_0^T, \hat{\beta}_t^2\sigma_S^2). \tag{5}$$

## Formulating the Diffusion Reverse Process

Assuming that the duration for each reverse step is $t$ ($t \approx \frac{n}{m}$), the objective in timestep $\{n-t\}$ is to recover $\boldsymbol{x}_{n-t}^T$ using $\boldsymbol{x}_n^T$. We introduce $q(\boldsymbol{x}_0^T)$ to achieve the density function:

$$q(\boldsymbol{x}_{n-t}^T|\boldsymbol{x}_n^T, \boldsymbol{x}_0^T) = \frac{q(\boldsymbol{x}_n^T|\boldsymbol{x}_{n-t}^T)q(\boldsymbol{x}_{n-t}^T|\boldsymbol{x}_0^T)}{q(\boldsymbol{x}_n^T|\boldsymbol{x}_0^T)}. \tag{6}$$

The density function can be given as:

$$\begin{aligned} q(\boldsymbol{x}_{n-t}^T|\boldsymbol{x}_n^T, \boldsymbol{x}_0^T) &:= \mathcal{N}(\boldsymbol{x}_{n-t}^T; u(\boldsymbol{x}_n^T) + v(\boldsymbol{x}_0^T), w(\sigma_S^2)), \\ \text{where} \quad u(\boldsymbol{x}_n^T) &= \frac{\hat{\beta}_{n-t}^2\hat{\alpha}_{n2t}}{\hat{\beta}_n^2}\boldsymbol{x}_n^T, v(\boldsymbol{x}_0^T) = \frac{\hat{\beta}_{n2t}^2\hat{\alpha}_{n-t}}{\hat{\beta}_n^2}\boldsymbol{x}_0^T \\ w(\sigma_S^2) &= \frac{\hat{\beta}_{n2t}^2\hat{\beta}_{n-t}^2}{\hat{\beta}_n^2}\sigma_S^2, \hat{\alpha}_{n2t} = \frac{\hat{\alpha}_n}{\hat{\alpha}_{n-t}}, \hat{\beta}_{n2t}^2 = 1 - \hat{\alpha}_{n2t}^2. \end{aligned} \tag{7}$$

## Other Training Strategies

### Target Guided Diffusion Training

Inspired by class guided diffusion, we can introduce $y$ into our formulation:

$$\begin{aligned} &\log p(\boldsymbol{x}_0^T|\boldsymbol{x}_n^S, \cdots, \boldsymbol{x}_1^S, y) = \log p(\boldsymbol{x}_0^T|\boldsymbol{x}_n^S, \cdots, \boldsymbol{x}_1^S) \\ &+ (\log p(y|\boldsymbol{x}_0^T) - \log p(y|\boldsymbol{x}_n^S, \cdots, \boldsymbol{x}_1^S)), \end{aligned} \tag{8}$$

Assume the weights of next teacher layer is $\boldsymbol{w}_t$, for $\boldsymbol{x}_0^T$ and predicted $\hat{\boldsymbol{x}_0^T}$, we simply use $\mathcal{L}_2$ loss, that is:

$$\mathcal{L}_{guided} = \left\|\boldsymbol{x}_0^T\boldsymbol{w}_t - \hat{\boldsymbol{x}_0^T}\boldsymbol{w}_t\right\|^2. \tag{9}$$

### Shuffle Sampling Strategy

One issue is that if we strictly follow diffusion weights rule, the last step of student features will dominate large weights such that other features are not fully stimulated to learn target features. We resolve this problem by introducing the shuffle sampling strategy:

$$p(\frac{1}{m}(\boldsymbol{x}_n^S + \cdots + \boldsymbol{x}_1^S)) = \mathcal{N}(0, \frac{1}{m}\sigma_S^2). \tag{10}$$

## Experiments

Due to the limited poster space, we only showcase the main results. For experiment setup and detailed results, please refer to our paper.

### CIFAR100 and ImageNet100 Results

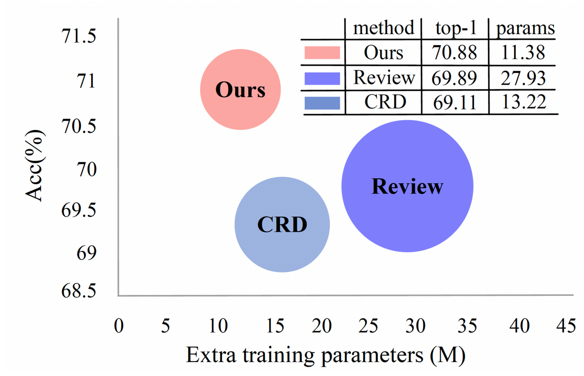
| Distillation Manner | Teacher / Acc | ResNet32x4 79.42 | WRN40-2 75.61 | VGG13 74.64 | ResNet50 79.34 | ResNet32x4 79.42 |
|---|---|---|---|---|---|---|
| | Student / Acc | ShuffleNetV1 70.50 | ShuffleNetV1 70.50 | MobileNetV2 64.6 | MobileNetV2 64.6 | ShuffleNetV2 71.82 |
| Logits | KD | 74.07 | 74.83 | 67.37 | 67.35 | 74.45 |
| Logits | DKD | 76.45 | 76.70 | 69.71 | 70.35 | 77.07 |
| Single Layer | FitNet | 73.59 | 73.73 | 64.14 | 63.16 | 73.54 |
| Single Layer | PKT | 74.10 | 73.89 | 67.13 | 66.52 | 74.69 |
| Single Layer | RKD | 72.28 | 72.21 | 64.52 | 64.43 | 73.21 |
| Single Layer | CRD | 75.11 | 76.05 | 69.73 | 69.11 | 75.65 |
| Multiple Layers | AT | 71.73 | 73.32 | 59.40 | 58.58 | 72.73 |
| Multiple Layers | VID | 73.38 | 73.61 | 65.56 | 67.57 | 73.40 |
| Multiple Layers | OFD | 75.98 | 75.85 | 69.48 | 69.04 | 76.82 |
| Multiple Layers | Review | 77.45 | 77.14 | 70.37 | 69.89 | 77.78 |
| Single Layer | Avgerage | 75.01 | 75.32 | 66.45 | 67.56 | 75.46 |
| Single Layer | Kdiffusion | 76.62 | 75.83 | 69.14 | 69.20 | 76.87 |
| Multiple Layer | Kdiffusion | 77.90 | 76.83 | 69.91 | 69.95 | 77.34 |
| + Target Guide | Kdiffusion | **78.14** | **77.26** | **70.49** | **71.14** | **77.84** |

Table 1. Results on CIFAR-100 with the teacher and student having different architectures.

| Distillation Manner | Teacher / Acc | Swin 94.48 | Swin 94.48 | Swin 94.48 | Swin 94.48 | Swin 94.48 |
|---|---|---|---|---|---|---|
| | Student / Acc | MobileNetV2 84.04 | MobileNetV3 84.98 | ResNet18 84.42 | ShuffleNetV1 74.74 | ShuffleNetV2 76.86 |
| Logits | KD | 85.00 | 86.76 | 85.12 | 77.30 | 79.18 |
| Logits | DKD | 85.38 | 86.86 | 85.50 | 77.28 | 80.02 |
| Single Layer | FitNet | 84.86 | 86.44 | 85.46 | 76.58 | 78.58 |
| Single Layer | PKT | 84.32 | 86.84 | 85.36 | 76.72 | 78.86 |
| Single Layer | SP | 85.02 | 85.90 | 85.20 | 76.96 | 78.86 |
| Single Layer | RKD | 78.68 | 85.06 | 84.82 | 76.90 | 77.48 |
| Single Layer | CRD | 83.72 | 84.94 | 84.26 | 73.20 | 77.88 |
| Multiple Layers | AT | 84.70 | 85.86 | 85.23 | 77.26 | 76.74 |
| Multiple Layers | VID | 85.42 | 86.46 | 85.12 | 77.56 | 79.46 |
| Multiple Layers | Review | 84.94 | 86.94 | 85.22 | 76.88 | 79.92 |
| Single Layer | Kdiffusion | 85.88 | 87.48 | 86.18 | 77.90 | 80.54 |
| Multiple Layer | Kdiffusion | **86.20** | **87.88** | **86.30** | **78.04** | **80.68** |

Table 2. Results on ImageNet-100 with the teacher and student having different architectures.

## Ablation Studies




| Teacher | Student | Baseline | Feature Numbers 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|---|
| Res32x4 | Sf1 | 70.50 | 74.85 | 75.96 | 76.28 | 76.80 |
| Res50 | Mv2 | 64.60 | 67.87 | 68.46 | 68.91 | 70.16 |

| Teacher | Student | Baseline | Student Stage 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| Res32x4 | Sf1 | 70.50 | 72.78 | 74.44 | 77.36 | 76.62 |
| Res32x4 | Sf2 | 71.82 | 73.15 | 75.28 | 77.32 | 76.87 |
| Res50 | Mv2 | 64.60 | 64.69 | 68.71 | 69.82 | 69.20 |

| | Student Stage 1 | 2 | 3 | 4 | Acc |
|---|---|---|---|---|---|
| Feature Num | | | | | 64.60 |
| | | | | ✓ | 64.22 |
| | | | ✓ | ✓ | 63.32 |
| | | ✓ | ✓ | ✓ | 62.31 |
| | ✓ | ✓ | ✓ | ✓ | 61.96 |

## Conclusions

- Large distribution gap distillation problem was studied.
- A novel diffusion-based distillation approach was introduced.
- Extensive experimental results demonstrate the effectiveness of the appproach.

## References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9163–9171, 2019.

The 38th Annual AAAI Conference on Artificial Intelligence, Vancouver, Canada