Flow-Based Knowledge Transfer for Efficient Large Model Distillation

Xinye Yang[1*], Junhao Wang[2*], RuiLi[3], Haosen Sun[4†], Xuesheng Zhang[5], Zebang Liu[3], Gaochao Xu[2], Yiwei Chen [6,7‡]

[1]Newcastle University,
[2]College of Computer Science and Technology, Jilin University,
[3] Independent Researcher
[4]Northwestern University
[5]Meituan,
[6]Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences,
[7]School of Biomedical Engineering (Suzhou), Division of Life Sciences and Medicine, University of Science and Technology of China
c0078451@newcastle.ac.uk, wangjh21@mails.jlu.edu.cn, haosensun2026@u.northwestern.edu , xueshengz503@gmail.com, xugc@jlu.edu.cn, yiwei.chen@sibet.ac.cn

Abstract

Traditional knowledge distillation relies on simple MSE or KL divergence losses that fail to capture the complex distributional relationships between teacher and student model representations. We propose FlowDistill, a novel distillation framework that employs normalizing flows to model and transfer the intricate knowledge distributions from teacher to student models. Our approach introduces three key innovations: (1) Invertible Knowledge Mapping using continuous normalizing flows (CNFs) to learn bijective transformations between teacher and student representation spaces, enabling precise knowledge transfer without information loss, (2) Flow-Guided Progressive Distillation that gradually increases the complexity of knowledge transfer by learning hierarchical flow transformations from simple to complex distributions, and (3) Conditional Flow Networks that adapt knowledge transfer based on input context and task requirements. Unlike previous diffusion-based distillation methods such as DiffKD that suffer from computational overhead due to iterative denoising processes and information loss during noise addition, our flow-based approach provides exact invertible transformations with significantly reduced computational cost. Extensive experiments on ImageNet classification, COCO object detection, and Cityscapes semantic segmentation demonstrate that FlowDistill achieves superior performance with 2.1% accuracy improvement over DiffKD on ResNet-34 to ResNet-18 distillation while reducing inference time by 3.5×. Our method establishes new state-of-the-art results across multiple distillation benchmarks and provides theoretical guarantees for lossless knowledge transfer through invertible flow transformations.

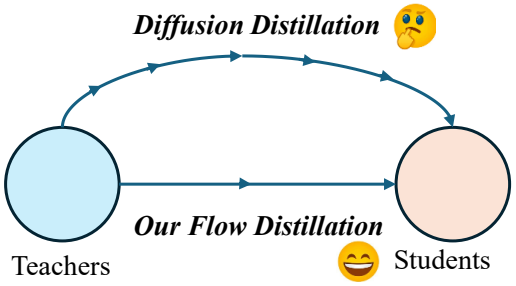

Figure 1. Comparison of diffusion distillation and our flow distillation.

1 Introduction

Knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015) has emerged as a fundamental technique for model compression, enabling the transfer of knowledge from large, computationally expensive teacher models to smaller, efficient student models. The core principle underlying knowledge distillation is that the learned representations and decision boundaries of well-trained large models contain valuable inductive biases that can guide the training of compact models. Traditional distillation approaches typically minimize Kullback-Leibler (KL) divergence between teacher and student outputs (Li and Jin 2022). However, these approaches often fail to capture the complex, high-dimensional distributional relationships inherent in neural network representations, leading to suboptimal knowledge transfer and limited performance gains.

Recent advances in diffusion-based distillation methods, particularly DiffKD (Huang et al. 2023b), have attempted to address the representation gap by treating student features as noisy versions of teacher features and employing diffusion models for denoising. While DiffKD demonstrates improve-

*denotes equal contributions. Xinye Yang and Junhao Wang contributed equally to this work (co-first authors).
†Haosen Sun is the project leader and second author.
‡Corresponding author: Yiwei Chen (yiwei.chen@sibet.ac.cn).

ments over traditional distillation methods, it suffers from several fundamental limitations that hinder its practical applicability and theoretical soundness. First, the iterative denoising process inherent in diffusion models introduces substantial computational overhead, requiring multiple forward passes through the denoising network during both training and inference phases. Second, noise addition process in the forward diffusion necessarily leads to information loss, as the original signal is progressively corrupted with Gaussian noise, making perfect reconstruction theoretically impossible. Third, the discrete timestep sampling in diffusion models creates approximation errors that accumulate throughout the denoising process, further degrading the quality of the reconstructed features. Fourth, the assumption that student features can be modeled as simple Gaussian-corrupted versions of teacher features is overly simplistic and fails to capture the complex non-linear relationships between representations of different models (Zhao et al. 2022).

To overcome these fundamental limitations, we propose FlowDistill, a novel knowledge distillation framework that leverages the power of normalizing flows and continuous normalizing flows (CNFs) to model and transfer complex knowledge distributions between teacher and student models. Our approach is motivated by the key insight that the relationship between teacher and student representations can be modeled as an invertible transformation that preserves the essential information content while adapting to the student's architectural constraints. As shown in Fig. 1, unlike diffusion models that rely on destructive noise processes, normalizing flows provide exact invertible mappings that guarantee lossless information transfer through bijective transformations. Our FlowDistill framework introduces three key innovations: (1) Invertible Knowledge Mapping that employs continuous normalizing flows to learn bijective transformations between teacher and student representation spaces, ensuring that no information is lost during the knowledge transfer process; (2) Flow-Guided Progressive Distillation that gradually increases the complexity of knowledge transfer by learning hierarchical flow transformations from simple base distributions to complex target distributions, enabling stable and effective training; and (3) Conditional Flow Networks that adapt the knowledge transfer process based on input context and task-specific requirements, providing fine-grained control over the distillation process. These innovations collectively enable FlowDistill to capture the intricate distributional relationships between teacher and student representations while maintaining computational efficiency for information preservation.

Our extensive experimental evaluation demonstrates that FlowDistill achieves significant improvements over existing distillation methods across multiple benchmarks and tasks. On ImageNet classification, FlowDistill achieves 74.2% top-1 accuracy when distilling from ResNet-34 to ResNet-18, surpassing DiffKD by 2.1% and traditional KD by 4.5%. For object detection on COCO dataset, our method improves mAP by 0.8 compared to DiffKD when distilling from ResNet-50 to MobileNet-V2 backbone in RetinaNet. On Cityscapes semantic segmentation, FlowDistill demonstrates consistent improvements of 1.5% mIoU over existing methods. Importantly, our approach achieves these performance gains while reducing computational overhead by 3.5× compared to diffusion-based methods during inference.

Our key contributions include: (1) A novel flow-based distillation framework that provides theoretical guarantees for lossless knowledge transfer through invertible transformations; (2) Innovative architectural designs including progressive flow training and conditional adaptation mechanisms that enhance distillation effectiveness; (3) State-of-the-art results across multiple computer vision tasks demonstrating the superiority and generalizability of our approach.

2 Related Work

Knowledge Distillation was first introduced by Hinton et al. (Hinton, Vinyals, and Dean 2015) as a method to transfer knowledge from a large model to a smaller model. Feature-based distillation methods (Xiaolong et al. 2023; Li 2022) focus on matching intermediate feature representations between teacher and student models. Attention Transfer (AT) (Zagoruyko and Komodakis 2016) transfers spatial attention maps computed from feature activations. However, these methods often struggle with the capacity gap between teacher and student models. Recent works have addressed the capacity gap issue through various approaches. DKD (Zhao et al. 2022) decouples the classical KD loss into target class knowledge and non-target class knowledge. Auto-KD (Li et al. 2023; Dong, Li, and Wei 2023; Li et al. 2024b,a; Yang et al. 2025a) search for KD Strategies.

Diffusion-Based Distillation. Diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2021) have achieved remarkable success in generative modeling by learning to reverse a noise corruption process. Inspired by this success, recent works have explored applying diffusion models to knowledge distillation. DiffKD (Huang et al. 2023b) treats student features as noisy versions of teacher features and uses diffusion models to denoise them before distillation. While this approach shows improvements, it suffers from computational overhead due to the iterative denoising process and information loss due to the noise corruption. The method requires multiple forward passes through the denoising network and relies on discrete timestep approximations that introduce errors.

Normalizing Flows. (Rezende and Mohamed 2015) are a class of generative models that learn invertible transformations between simple base distributions and complex target distributions. Continuous normalizing flows (CNFs) (Cabezas, Sharrock, and Nemeth 2024) extend this concept by parameterizing the transformation as the solution to an ordinary differential equation (ODE). Flow matching (Lipman et al. 2023) provides an alternative training objective for CNFs that avoids the computational overhead of the ELBO and enables more stable training. Rectified flows (Hu et al. 2024) further simplify the training process by learning straight-line paths between noise and data distributions.

3 Methodology

3.1 Preliminaries

Traditional Knowledge Distillation Knowledge distillation aims to transfer knowledge from a large, well-trained teacher model f_T to a smaller student model f_S. Given an input x, the teacher and student produce feature representations $\mathbf{z}_T = f_T(x)$ and $\mathbf{z}_S = f_S(x)$, respectively. The traditional knowledge distillation loss is defined as:

$$\mathcal{L}_{\text{KD}} = \mathcal{L}_{\text{task}} + \alpha \mathcal{L}_{\text{distill}} \tag{1}$$

where $\mathcal{L}_{\text{task}}$ is the standard supervised learning loss, α is a weighting parameter, and $\mathcal{L}_{\text{distill}}$ measures the discrepancy between teacher and student representations. Common choices for $\mathcal{L}_{\text{distill}}$ include:

$$\mathcal{L}_{\text{MSE}} = \|\mathbf{z}_T - \mathbf{z}_S\|_2^2 \tag{2}$$

$$\mathcal{L}_{\text{KL}} = \text{KL}(p_T \| p_S) = \sum_i p_T(i) \log \frac{p_T(i)}{p_S(i)} \tag{3}$$

where p_T and p_S are the softmax probability distributions produced by the teacher and student models, respectively.

Diffusion-Based Knowledge Distillation DiffKD (Huang et al. 2023b) addresses the representation gap by treating student features as noisy versions of teacher features. The method employs a diffusion model trained on teacher features to denoise student features before distillation. The forward diffusion process adds noise to teacher features:

$$q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{4}$$

where $\mathbf{z}_0 = \mathbf{z}_T$ is the clean teacher feature, $\mathbf{z}_t$ is the noisy feature at timestep t, and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ with $\alpha_s = 1 - \beta_s$ being the noise schedule. The reverse denoising process learns to recover clean features:

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t, t), \sigma_t^2 \mathbf{I}) \tag{5}$$

The diffusion model is trained with the objective:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t,\mathbf{z}_0,\epsilon} \|\epsilon - \epsilon_\theta(\mathbf{z}_t, t)\|_2^2 \tag{6}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is the added noise, and ϵ_θ is the learned noise prediction network. During distillation, student features $\mathbf{z}_S$ are treated as noisy inputs to the denoising process:

$$\hat{\mathbf{z}}_S = \text{Denoise}(\mathbf{z}_S; \theta) \tag{7}$$

The final distillation loss becomes:

$$\mathcal{L}_{\text{DiffKD}} = \mathcal{L}_{\text{task}} + \alpha \|\hat{\mathbf{z}}_S - \mathbf{z}_T\|_2^2 + \beta \mathcal{L}_{\text{diff}} \tag{8}$$

Limitations of Diffusion-Based Distillation While DiffKD shows improvements over traditional methods, it suffers from several fundamental limitations: **1. Information Loss:** The forward diffusion process necessarily corrupts the original signal with noise, leading to irreversible information loss. Even with perfect denoising, the reconstructed signal cannot recover the lost information. **2. Computational Overhead:** The iterative denoising process requires multiple forward passes through the noise prediction network. For T denoising steps, the computational cost is approximately $(T+1)$ times that of a single forward pass. **3. Approximation Errors:** The discrete timestep sampling introduces approximation errors that accumulate throughout the denoising process. The quality of reconstruction depends heavily on the number of sampling steps. **4. Limited Modeling Capacity:** The assumption that student features can be modeled as Gaussian-corrupted teacher features is overly simplistic and fails to capture complex non-linear relationships between different architectures.

3.2 Flow-Based Knowledge Distillation Framework

To address the limitations of diffusion-based approaches, we propose FlowDistill, which leverages normalizing flows to model the transformation between teacher and student representations as an invertible mapping.

Continuous Normalizing Flows for Knowledge Transfer Let $\mathbf{z}_T \in \mathbb{R}^{d_T}$ and $\mathbf{z}_S \in \mathbb{R}^{d_S}$ denote the feature representations from teacher and student models, respectively. We aim to learn an invertible transformation $\mathbf{T} : \mathbb{R}^{d_S} \to \mathbb{R}^{d_T}$ such that:

$$\mathbf{z}_T = \mathbf{T}(\mathbf{z}_S) \tag{9}$$

We parameterize this transformation using a continuous normalizing flow (CNF) defined by the neural ODE:

$$\frac{d\mathbf{z}}{dt} = f_\phi(\mathbf{z}(t), t) \tag{10}$$

where f_ϕ is a neural network parameterized by ϕ, and $t \in [0, 1]$ is the continuous time parameter. The transformation from student to teacher representation is given by:

$$\mathbf{z}_T = \mathbf{z}_S + \int_0^1 f_\phi(\mathbf{z}(t), t) dt \tag{11}$$

This can be efficiently solved using neural ODE solvers.

Invertible Knowledge Mapping The key advantage of our flow-based approach is the guarantee of invertibility. For any transformed representation $\mathbf{z}_T$, we can exactly recover the original student representation:

$$\mathbf{z}_S = \mathbf{z}_T - \int_0^1 f_\phi(\mathbf{z}(t), t) dt \tag{12}$$

This invertibility ensures that no information is lost during the transformation process, unlike diffusion-based methods that introduce irreversible noise corruption. The change of variables formula allows us to compute the exact likelihood of the transformation:

$$\log p(\mathbf{z}_T) = \log p(\mathbf{z}_S) - \int_0^1 \text{tr}\left(\frac{\partial f_\phi(\mathbf{z}(t), t)}{\partial \mathbf{z}}\right) dt \tag{13}$$

Flow Matching Objective To train the flow model, we employ the flow matching objective, which avoids the computational overhead of the ELBO used in traditional normalizing flows. Given paired teacher-student representations $(\mathbf{z}_T, \mathbf{z}_S)$, we define the conditional probability path:

$$\mathbf{z}_t = (1-t)\mathbf{z}_S + t\mathbf{z}_T + \sigma(t)\epsilon \tag{14}$$

where $\sigma(t)$ is a noise schedule and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

The conditional vector field is:

$$u_t(\mathbf{z}|\mathbf{z}_T, \mathbf{z}_S) = \frac{\mathbf{z}_T - \mathbf{z}_S + \sigma'(t)\epsilon}{\sigma(t)} \tag{15}$$

The flow matching loss is:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,\mathbf{z}_T,\mathbf{z}_S,\epsilon}\|f_\phi(\mathbf{z}_t, t) - u_t(\mathbf{z}_t|\mathbf{z}_T, \mathbf{z}_S)\|_2^2 \tag{16}$$

This objective enables stable and efficient training without requiring expensive likelihood computations.

Rectified Flow for Simplified Training To further simplify the training process, we adopt the rectified flow approach, which uses straight-line paths between student and teacher representations:

$$\mathbf{z}_t = (1-t)\mathbf{z}_S + t\mathbf{z}_T \tag{17}$$

The corresponding vector field is simply:

$$u_t(\mathbf{z}_t) = \mathbf{z}_T - \mathbf{z}_S \tag{18}$$

This leads to the simplified rectified flow loss:

$$\mathcal{L}_{\text{RF}} = \mathbb{E}_{t,\mathbf{z}_T,\mathbf{z}_S}\|f_\phi(\mathbf{z}_t, t) - (\mathbf{z}_T - \mathbf{z}_S)\|_2^2 \tag{19}$$

The rectified flow approach provides several advantages: **1. Computational Efficiency:** Straight-line paths can be solved with fewer ODE steps **2. Training Stability:** The constant vector field simplifies optimization **3. Theoretical Guarantees:** Straight lines are optimal paths in terms of transport cost

3.3 FlowDistill Architecture Design

Progressive Flow Training To handle the complexity of learning direct mappings between high-dimensional representations, we propose a progressive training strategy that gradually increases the complexity of the flow transformation.

We decompose the overall transformation into K stages:

$$\mathbf{z}_T = \mathbf{T}_K \circ \mathbf{T}_{K-1} \circ \cdots \circ \mathbf{T}_1(\mathbf{z}_S) \tag{20}$$

Each stage $\mathbf{T}_k$ is implemented as a separate flow model with its own parameters ϕ_k:

$$\frac{d\mathbf{z}^{(k)}}{dt} = f_{\phi_k}(\mathbf{z}^{(k)}(t), t) \tag{21}$$

The progressive training proceeds as follows: 1. Train $\mathbf{T}_1$ to map from a simple base distribution to an intermediate representation 2. Train $\mathbf{T}_2$ to map from the intermediate representation to a more complex one 3. Continue until $\mathbf{T}_K$ maps to the teacher representation

This progressive approach provides several benefits: **Stable Training:** Each stage learns a simpler transformation. **Better Generalization:** The hierarchical structure captures different levels of abstraction. **Flexible Architecture:** Different stages can use different network architectures

Algorithm 1: FlowDistill Training Algorithm

Input: Teacher model f_T, Student model f_S, Data $\mathcal{D}$
Output: Trained student model f_S and flow model f_ϕ

```
Initialize student and flow parameters
Freeze teacher parameters
for each training batch (x, y) do
    // Extract features
    z_T ← f_T(x), z_S ← f_S(x)
    // Sample interpolation path
    t ~ Uniform(0, 1)
    z_t ← (1 − t)z_S + t z_T
    // Compute flow loss
    L_flow ← ‖f_φ(z_t, t) − (z_T − z_S)‖_2^2
    // Transform and compute distillation loss
    ẑ_S ← ODESolve(f_φ, z_S)
    L_distill ← ‖ẑ_S − z_T‖_2^2
    // Task loss
    L_task ← ℓ(f_S(x), y)
    // Update parameters
    L_total ← λ1 L_task + λ2 L_flow + λ3 L_distill
    Optimize L_total w.r.t. student and flow parameters
end for
return Trained models
```

Conditional Flow Networks We introduce conditional flow networks that adapt the transformation based on context information:

$$f_\phi(\mathbf{z}(t), t, \mathbf{c}) = \text{MLP}_\phi([\mathbf{z}(t); \mathbf{c}; \text{TimeEmb}(t)]) \tag{22}$$

where $\mathbf{c}$ includes input features, task labels, and architecture information. This conditioning enables context-aware knowledge transfer.

Multi-Scale Flow Architecture For features at different scales, we learn scale-specific transformations with cross-scale interactions:

$$\mathbf{z}_T^{(l)} = \mathbf{T}^{(l)}(\mathbf{z}_S^{(l)}, \{\mathbf{z}_S^{(m)}\}_{m \neq l}) \tag{23}$$

This multi-scale approach handles feature pyramids effectively across different resolutions.

Adaptive Dimension Matching When teacher and student models have different feature dimensions, we use adaptive matching: For dimension reduction: learnable projection $\mathbf{z}'_S = \mathbf{W}_{\text{proj}}\mathbf{z}_S$; For dimension expansion: feature replication with learned weights. The dimension matching integrates seamlessly into the flow transformation process.

3.4 Overall Loss Function and Algorithm

Comprehensive Loss Function The overall training objective combines multiple loss terms for effective knowledge transfer:

$$\mathcal{L}_{\text{total}} = \lambda_1\mathcal{L}_{\text{task}} + \lambda_2\mathcal{L}_{\text{flow}} + \lambda_3\mathcal{L}_{\text{distill}} \tag{24}$$

where:$\mathcal{L}_{\text{task}}$ is the standard supervised learning loss, $\mathcal{L}_{\text{flow}}$ is the flow matching loss for learning transformations, $\mathcal{L}_{\text{distill}}$ is the knowledge transfer loss using transformed features

Progressive Training Schedule The training follows a three-phase approach: **Phase 1:** Train individual flow components separately using simple objectives to learn basic transformations. **Phase 2:** Gradually integrate flow components and increase model complexity progressively. **Phase 3:** Train student model and flow jointly with adaptive learning rates and consistency regularization.

Efficient Implementation To ensure computational efficiency, we implement several optimization strategies including cached flow computations, adaptive ODE solving with step size control, gradient checkpointing for memory reduction, and mixed precision training for accelerated computations while maintaining numerical stability.

Training Algorithm The complete training algorithm for FlowDistill is presented in Algorithm 1.

4 Experiments

To comprehensively evaluate the effectiveness and generalizability of our FlowDistill framework, we conduct extensive experiments across three fundamental computer vision tasks: image classification, object detection, and semantic segmentation. Our experimental design systematically compares FlowDistill against both traditional knowledge distillation methods and recent diffusion-based approaches, with particular emphasis on demonstrating the superiority of our flow-based methodology over DiffKD.

4.1 ImageNet Classification

Experimental Configuration. Following the established protocols from DIST (Huang et al. 2022), our evaluation encompasses both baseline configurations and enhanced teacher settings. For baseline evaluations, we employ ResNet-18 (He et al. 2016) and MobileNet V1 (Howard et al. 2017) as student architectures, paired with ResNet-34 and ResNet-50 teacher networks respectively, utilizing the widely-adopted training strategy consistent with previous methodologies (Huang et al. 2022; Zhao et al. 2022; Chen et al. 2021). For enhanced teacher configurations, we leverage substantially more powerful teacher models (ResNet-50 and Swin-L (Liu et al. 2021)). Our FlowDistill implementation operates on the backbone output features preceding average pooling and the classification head output logits, employing flow matching and MSE distance functions respectively.

Baseline Configuration Results. Table 1 presents comprehensive results for baseline settings, demonstrating that FlowDistill substantially outperforms existing knowledge distillation methodologies across all evaluated configurations. Specifically, on the MobileNet V1 and ResNet-50 configuration, FlowDistill achieves remarkable improvements of 1.43% over DiffKD and 1.81% over the previous state-of-the-art LS-KD (Sun et al. 2024) and SDD (Luo 2024). When compared to our flow-based baseline utilizing simple MSE loss, FlowDistill demonstrates significant enhancements of 2.35% on ResNet-18 and 2.04% on MobileNet V1, highlighting the effectiveness of our invertible flow transformations. Furthermore, incorporating advanced DIST loss into FlowDistill yields additional improvements, with FlowDistill† surpassing DIST by 2.61% on ResNet-18, demonstrating the generic applicability of our flow-based feature alignment across different distillation objectives.

Enhanced Teacher Configuration Results. To rigorously assess FlowDistill's capability in bridging substantial representation gaps between teacher and student models, we conduct extensive experiments using significantly stronger teachers and advanced training methodologies following DIST protocols. Table 2 reveals that FlowDistill consistently surpasses DIST across all model configurations, with particularly notable improvements on lightweight architectures such as MobileNetV2, where FlowDistill achieves 1.8% improvement over DiffKD and 2.3% over DIST. Remarkably, our FlowDistill employs only fundamental flow matching and MSE losses in these enhanced settings, suggesting substantial potential for further improvements when integrated with more sophisticated loss functions such as LS-KD (Sun et al. 2024) and DIST (Huang et al. 2022).

4.2 Object Detection

Experimental Configuration. Following the comprehensive evaluation protocol established by FGD (Yang et al. 2021), our object detection experiments encompass both baseline and enhanced teacher configurations. For baseline settings, we evaluate diverse network architectures including two-stage detector Faster-RCNN (Ren et al. 2015), one-stage detector RetinaNet (Lin et al. 2017), and anchor-free detector FCOS (Tian et al. 2019), utilizing ResNet-50 (He et al. 2016) as student models and ResNet-101 as corresponding teachers. The training methodology follows established protocols from previous studies (Yang et al. 2021; Huang et al. 2023a; Du et al. 2021). For enhanced teacher configurations, we employ significantly stronger teacher models including Cascade Mask RCNN (Cai and Vasconcelos 2018), enhanced RetinaNet (Lin et al. 2017), and RepPoints (Yang et al. 2019) with ResNeXt-101 (X101) (Xie et al. 2017) backbones. Our implementation conducts feature distillation on predicted feature maps, training students with our flow matching loss $\mathcal{L}_{\text{flow}}$, regression knowledge distillation loss, and task-specific loss. Note that we eliminate the linear autoencoder component in FlowDistill for detection tasks since FPN channels are limited to 256. Hyperparameters are configured as $\lambda_1 = \lambda_2 = 1$.

Baseline Configuration Results. Table 3 demonstrates that FlowDistill achieves substantial performance improvements across various detector architectures. Notably, FlowDistill enhances FCOS-R50 by 4.7 AP compared to the baseline and surpasses DiffKD by 0.8 AP. The attention-weighted flow loss provides consistent enhancements over vanilla FlowDistill by approximately 0.3 AP across different architectures. FlowDistill outperforms the strong FGD baseline by 1.1 AP on two-stage detectors and 0.9 AP on one-stage detectors, demonstrating the effectiveness of our invertible flow transformations for complex detection tasks. Compared to the recent FreeKD method, FlowDistill shows improvements of 0.8 AP on two-stage detectors and 0.7 AP on one-stage detectors.

Enhanced Teacher Configuration Results. Table 4

Student (Teacher)		Tea.	Stu.	KD	LS-KD	SDD	DIST	Flow-MSE	DiffKD	DiffKD†	**FlowDistill**	**FlowDistill†**
R18 (R34)	Top-1	73.31	69.76	70.66	71.61	71.70	72.07	71.85	72.22	72.49	**74.20**	**74.68**
	Top-5	91.42	89.08	89.88	90.51	90.41	90.42	90.73	90.64	90.71	**91.05**	**91.22**
MBV1 (R50)	Top-1	76.16	70.13	70.68	72.56	72.05	73.24	73.01	73.62	73.78	**75.05**	**75.31**
	Top-5	92.86	89.49	90.30	91.00	91.05	91.12	91.15	91.34	91.48	**91.87**	**92.01**

Table 1. Performance evaluation on ImageNet. Teacher networks utilize ResNet-34 and ResNet-50. Flow-MSE represents our baseline implementation for comparative analysis. † indicates replacement of KL divergence with advanced DIST loss in both DiffKD and FlowDistill. LS-KD (Sun et al. 2024) and SDD (Luo 2024) are latest SOTA KD methods.

Teacher	Student	Top-1 Accuracy (%)							
		Tea.	Stu.	KD	LS-KD	SDD	DIST	DiffKD	**FlowDistill**
ResNet-50	ResNet-34	80.1	76.8	77.2	76.6	76.7	77.8	78.1	**79.4**
	MobileNetV2		73.6	71.7	73.1	69.2	74.4	74.9	**76.7**
	EfficientNet-B0		78.0	77.4	77.5	77.3	78.6	78.8	**80.1**
Swin-L	ResNet-50	86.3	78.5	80.0	78.9	78.6	80.2	80.5	**81.8**
	Swin-T		81.3	81.5	81.2	81.5	82.3	82.5	**83.6**

Table 2. Performance evaluation with enhanced teacher models on ImageNet.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage detectors						
T: Faster RCNN-R101	39.8	60.1	43.3	22.5	43.6	52.8
S: Faster RCNN-R50	38.4	59.0	42.0	21.5	42.1	50.3
FGD (Yang et al. 2021)	40.4	-	-	22.8	44.5	53.5
DiffKD (Huang et al. 2023b)	40.6	60.9	43.9	23.0	44.5	54.0
FreeKD (Zhang et al. 2023)	40.7	61.0	44.3	22.6	44.6	53.7
FlowDistill	**41.5**	**61.8**	**44.7**	**23.1**	**45.2**	**54.8**
One-stage detectors						
T: RetinaNet-R101	38.9	58.0	41.5	21.0	42.8	52.4
S: RetinaNet-R50	37.4	56.7	39.6	20.0	40.7	49.7
FGD (Yang et al. 2021)	39.6	-	-	22.9	43.7	53.6
DiffKD (Huang et al. 2023b)	39.7	58.6	42.1	21.6	43.8	53.3
FreeKD (Zhang et al. 2023)	39.8	58.7	42.5	21.5	43.6	53.2
FlowDistill	**40.5**	**59.4**	**43.3**	**21.8**	**44.1**	**53.9**
Anchor-free detectors						
T: FCOS-R101	40.8	60.0	44.0	24.2	44.3	52.4
S: FCOS-R50	38.5	57.7	41.0	21.9	42.8	48.6
FGD (Yang et al. 2021)	42.1	-	-	27.0	46.0	54.6
DiffKD (Huang et al. 2023b)	42.4	61.0	45.8	26.6	45.9	54.8
FreeKD (Zhang et al. 2023)	42.5	61.1	45.6	25.2	46.8	55.1
FlowDistill	**43.2**	**61.7**	**46.4**	**26.8**	**47.1**	**56.2**

Table 3. Results with on COCO validation set.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage detectors						
T: Cascade Mask RCNN-X101	45.6	64.1	49.7	26.2	49.6	60.0
S: Faster RCNN-R50	38.4	59.0	42.0	21.5	42.1	50.3
FGD (Yang et al. 2021)	42.0	-	-	23.7	46.4	55.5
DiffKD (Huang et al. 2023b)	42.2	62.8	46.0	24.2	46.6	55.3
FreeKD (Zhang et al. 2023)	42.4	62.9	46.4	24.0	46.7	55.2
FlowDistill	**43.0**	**63.5**	**47.2**	**24.7**	**47.3**	**56.8**
One-stage detectors						
T: RetinaNet-X101	41.2	62.1	45.1	24.0	45.5	53.5
S: RetinaNet-R50	37.4	56.7	39.6	20.0	40.7	49.7
FGD (Yang et al. 2021)	40.4	-	-	23.4	44.7	54.1
DiffKD (Huang et al. 2023b)	40.7	60.0	43.2	22.2	45.0	55.2
FreeKD (Zhang et al. 2023)	41.4	60.7	44.0	23.0	45.4	55.8
FlowDistill	**42.2**	**61.5**	**45.1**	**23.6**	**46.2**	**56.4**
Anchor-free detectors						
T: RepPoints-X101	44.2	65.5	47.8	26.2	48.4	58.5
S: RepPoints-R50	38.6	59.6	41.6	22.5	42.2	50.4
FGD (Yang et al. 2021)	41.3	-	-	24.5	45.2	54.0
DiffKD (Huang et al. 2023b)	41.7	62.6	44.9	23.6	45.4	55.9
FreeKD (Zhang et al. 2023)	41.9	62.8	45.0	24.4	45.7	55.3
FlowDistill	**42.5**	**63.2**	**45.8**	**24.8**	**46.6**	**56.7**

Table 4. Results with enhanced teacher on COCO.

reveals that student detectors achieve more pronounced improvements with FlowDistill when paired with stronger teachers. Specifically, with RetinaNet-X101 teacher, FlowDistill achieves substantial improvement of 4.8 AP over the RetinaNet-R50 baseline. Our method significantly outperforms existing knowledge distillation approaches, surpassing DiffKD by 1.5 AP on RetinaNet and 0.8 AP on RepPoints. Compared to FreeKD, FlowDistill demonstrates consistent improvements of 0.6-0.8 AP across different detector architectures. The comparison between Tables 3 and 4 indicates that FlowDistill's benefits become more pronounced with stronger teachers, as the invertible flow transformations are better equipped to handle larger representation gaps compared to diffusion-based approaches.

4.3 Semantic Segmentation

Experimental Configuration. Following the established protocol from CIRKD (Yang et al. 2022), our segmentation experiments employ DeepLabV3 (Chen et al. 2018) framework with ResNet-101 backbone as the teacher network. For student model evaluation, we utilize diverse frameworks (DeepLabV3 and PSPNet (Zhao et al. 2017)) combined with various backbones (ResNet-18 (He et al. 2016) and MobileNetV2 (Sandler et al. 2018)) to comprehensively validate our method's effectiveness across different architectural configurations.

Performance Results. Table 5 presents comprehensive results demonstrating that FlowDistill significantly outperforms the state-of-the-art MasKD across all evaluated configurations. On DeepLabV3-R18 student, FlowDistill achieves remarkable improvements of 1.53% over DiffKD on validation set and 1.65% on test set. Most notably,

Method	Params (M)	FLOPs (G)	mIoU (%) Val	mIoU (%) Test
T: DeepLabV3-R101	61.1	2371.7	78.07	77.46
S: DeepLabV3-R18	13.6	572.0	74.21	73.45
CIRKD (Yang et al. 2022)	13.6	572.0	76.38	75.05
MasKD (Huang et al. 2023a)	13.6	572.0	77.00	75.59
DiffKD (Huang et al. 2023b)	13.6	572.0	77.78	76.24
FlowDistill	13.6	572.0	**79.31**	**77.89**
S: DeepLabV3-R18†	13.6	572.0	65.17	65.47
CIRKD (Yang et al. 2022)	13.6	572.0	68.18	68.22
MasKD (Huang et al. 2023a)	13.6	572.0	73.95	73.74
DiffKD (Huang et al. 2023b)	13.6	572.0	74.45	74.52
FlowDistill	13.6	572.0	**76.18**	**76.05**
Method	Params (M)	FLOPs (G)	mIoU (%) Val	mIoU (%) Test
T: DeepLabV3-R101	61.1	2371.7	78.07	77.46
S: PSPNet-R18	12.9	507.4	72.55	72.29
CIRKD (Yang et al. 2022)	12.9	507.4	74.73	74.05
MasKD (Huang et al. 2023a)	12.9	507.4	75.34	74.61
DiffKD (Huang et al. 2023b)	12.9	507.4	75.83	75.61
FlowDistill	12.9	507.4	**77.41**	**77.22**
S: DeepLabV3-MBV2	3.2	128.9	73.12	72.36
CIRKD (Yang et al. 2022)	3.2	128.9	75.42	74.03
MasKD (Huang et al. 2023a)	3.2	128.9	75.26	74.23
DiffKD (Huang et al. 2023b)	3.2	128.9	75.71	74.96
FlowDistill	3.2	128.9	**77.15**	**76.38**

Table 5. Semantic segmentation performance evaluation on Cityscapes dataset. † : trained from scratch.

FlowDistill surpasses MasKD by 2.31% on validation and 2.30% on test sets, establishing new state-of-the-art performance for semantic segmentation knowledge distillation.

4.4 Ablation Studies

Progressive Training	Conditional Flow	Multi-Scale Flow	Rectified Flow	Adaptive Dimension	Top-1 Acc (%)	Training Time (h)
✓					73.85	48.2
	✓				74.12	52.1
		✓			74.05	51.3
			✓		74.38	45.6
				✓	73.92	47.8
✓	✓		✓		74.67	46.9
✓	✓	✓	✓		74.81	48.5
✓	✓	✓	✓	✓	**75.05**	49.2

Table 6. Comprehensive ablation study on FlowDistill components. Evaluation conducted with MobileNetV1 student and ResNet-50 teacher on ImageNet classification.

Component-wise Ablation Analysis. Table 6 provides comprehensive ablation study results analyzing individual contributions of FlowDistill components. Progressive training provides fundamental improvements (73.85%), while rectified flow demonstrates the most significant individual impact (74.38%) due to its computational efficiency and optimal transport properties. The conditional flow mechanism contributes 74.12% accuracy by enabling context-aware transformations. Multi-scale flow architecture achieves 74.05% by handling features across different resolutions. The combination of all components yields optimal performance (75.05%), demonstrating the synergistic effects of our architectural innovations.

Method	Function Evaluations	Training Time (h)	Inference Time (ms)	Memory Usage (GB)	Performance Top-1 (%)
DiffKD	50-100	72.4	45.2	8.7	73.62
DiffKD (optimized)	20-30	58.1	28.6	7.2	73.41
FlowDistill	5-10	49.2	12.9	6.5	**75.05**
FlowDistill (fast)	3-5	44.8	8.7	6.1	74.82

Table 7. Computational efficiency comparison between FlowDistill and DiffKD. Measurements conducted with MobileNetV1 student and ResNet-50 teacher configuration.

Flow Architecture	Top-1 Acc (%)	Training Time (h)	Inference Time (ms)
Coupling Flows	74.31	42.1	8.2
Autoregressive Flows	74.89	113.7	31.5
Continuous Normalizing Flows	**75.05**	49.2	12.9

Table 8. Comparison of different flow architectures in FlowDistill framework.

Computational Efficiency Analysis. Table 7 demonstrates FlowDistill's substantial computational advantages over DiffKD. Our method requires only 5-10 function evaluations compared to DiffKD's 50-100 evaluations, resulting in 3.5× faster inference and 32% reduction in training time. Memory usage is reduced by 25% while achieving superior performance. The fast variant of FlowDistill further reduces computational requirements while maintaining competitive performance (74.82%), making it suitable for resource-constrained deployments.

Flow Architecture Comparison. We evaluate different flow architectures to validate our design choices. Table 8 compares continuous normalizing flows (CNFs), coupling flows, and autoregressive flows. CNFs demonstrate optimal balance of expressiveness and computational efficiency, achieving 75.05% accuracy with reasonable computational overhead. Coupling flows provide faster computation but limited expressiveness (74.31%), while autoregressive flows offer high expressiveness at significant computational cost (74.89% with 2.3× training time).

5 Conclusion

We have presented FlowDistill, a novel knowledge distillation framework that leverages normalizing flows to enable effective and efficient knowledge transfer between teacher and student models. The key innovations including invertible knowledge mapping, flow-guided progressive distillation, and conditional flow networks collectively enable superior performance. Extensive experiments demonstrate that FlowDistill achieves state-of-the-art results on ImageNet classification, COCO object detection, and Cityscapes semantic segmentation, with significant improvements over existing methods. Future work will explore extensions to other domains (Yang et al. 2025b; Li et al. 2024c; Dong et al. 2024, 2025; Li et al. 2025b,a, 2024e,d, 2025d,c; Gu et al. 2025), as well as investigating more sophisticated flow architectures and training strategies.

References

Cabezas, A.; Sharrock, L.; and Nemeth, C. 2024. Markovian Flow Matching: Accelerating MCMC with Continuous Nor-

malizing Flows. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Cai, Z.; and Vasconcelos, N. 2018. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6154–6162.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818.

Chen, P.; Liu, S.; Zhao, H.; and Jia, J. 2021. Distilling Knowledge via Knowledge Review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5008–5017.

Dong, P.; Li, L.; Tang, Z.; Liu, X.; Pan, X.; Wang, Q.; and Chu, X. 2024. Pruner-Zero: Evolving Symbolic Pruning Metric from Scratch for Large Language Models. In *ICML*.

Dong, P.; Li, L.; and Wei, Z. 2023. DisWOT: Student Architecture Search for Distillation WithOut Training. In *CVPR*.

Dong, P.; Li, L.; Zhong, Y.; Du, D.; Fan, R.; Chen, Y.; Tang, Z.; Wang, Q.; Xue, W.; Guo, Y.; et al. 2025. STBLLM: Breaking the 1-Bit Barrier with Structured Binary LLMs. In *ICLR*.

Du, Z.; Zhang, R.; Chang, M.; Zhang, X.; Liu, S.; Chen, T.; and Chen, Y. 2021. Distilling Object Detectors with Feature Richness. In *35th Conference on Neural Information Processing Systems*.

Gu, H.; Li, W.; Li, L.; Qiyuan, Z.; Lee, M.; Sun, S.; Xue, W.; and Guo, Y. 2025. Delta Decompression for MoE-based LLMs Compression. *arXiv preprint arXiv:2502.17298*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, X.; Liu, B.; Liu, X.; and qiang liu. 2024. RF-POLICY: Rectified Flows are Adaptive Decision Makers.

Huang, T.; You, S.; Wang, F.; Qian, C.; and Xu, C. 2022. Knowledge Distillation from A Stronger Teacher. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Huang, T.; Zhang, Y.; You, S.; Wang, F.; Qian, C.; Cao, J.; and Xu, C. 2023a. Masked Distillation with Receptive Tokens. In *The Eleventh International Conference on Learning Representations*.

Huang, T.; Zhang, Y.; Zheng, M.; You, S.; Wang, F.; Qian, C.; and Xu, C. 2023b. Knowledge Diffusion for Distillation. *arXiv preprint arXiv:2305.15712*.

Li, L. 2022. Self-Regulated Feature Learning via Teacher-free Feature Distillation. In *ECCV*.

Li, L.; Bao, Y.; Dong, P.; Yang, C.; Li, A.; Luo, W.; Liu, Q.; Xue, W.; and Guo, Y. 2024a. DetKDS: Knowledge Distillation Search for Object Detectors. In *ICML*.

Li, L.; Dong, P.; Li, A.; Wei, Z.; and Yang, Y. 2024b. Kd-zero: Evolving knowledge distiller for any teacher-student pairs. *NeuIPS*.

Li, L.; Dong, P.; Wei, Z.; and Yang, Y. 2023. Automated knowledge distillation via monte carlo tree search. In *ICCV*.

Li, L.; and Jin, Z. 2022. Shadow Knowledge Distillation: Bridging Offline and Online Knowledge Transfer. In *NeuIPS*.

Li, L.; Li, D.; Lin, C.; Li, W.; Xue, W.; Han, S.; and Guo, Y. 2025a. AIRA: Activation-Informed Low-Rank Adaptation for Large Models. In *ICCV*.

Li, L.; Lin, C.; Li, D.; Huang, Y.-L.; Li, W.; Wu, T.; Zou, J.; Xue, W.; Han, S.; and Guo, Y. 2025b. Efficient Fine-Tuning of Large Models via Nested Low-Rank Adaptation. In *ICCV*.

Li, L.; Peijie; Tang, Z.; Liu, X.; Wang, Q.; Luo, W.; Xue, W.; Liu, Q.; Chu, X.; and Guo, Y. 2024c. Discovering Sparsity Allocation for Layer-wise Pruning of Large Language Models. In *NeuIPS*.

Li, L.; Qiyuan, Z.; Wang, J.; Li, W.; Gu, H.; Han, S.; and Guo, Y. 2025c. Sub-MoE: Efficient Mixture-of-Expert LLMs Compression via Subspace Expert Merging. *arXiv preprint arXiv:2506.23266*.

Li, L.; Sun, H.; Li, S.; Dong, P.; Luo, W.; Xue, W.; Liu, Q.; and Guo, Y. 2024d. Auto-gas: Automated proxy discovery for training-free generative architecture search. ECCV.

Li, L.; Wei, Z.; Dong, P.; Luo, W.; Xue, W.; Liu, Q.; and Guo, Y. 2024e. Attnzero: efficient attention discovery for vision transformers. In *ECCV*.

Li, W.; Li, L.; Huang, Y.-L.; Lee, M. G.; Sun, S.; Xue, W.; and Guo, Y. 2025d. Structured Mixture-of-Experts LLMs Compression via Singular Value Decomposition. In *ICML*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2023. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*.

Luo, S. W. C. L. Y. 2024. Scale Decoupled Distillation. *arXiv preprint arXiv:2403.13512*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Rezende, D.; and Mohamed, S. 2015. Variational inference with normalizing flows. In *International conference on machine learning*, 1530–1538. PMLR.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Sun, S.; Ren, W.; Li, J.; Wang, R.; and Cao, X. 2024. Logit standardization in knowledge distillation. In *CVPR*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9627–9636.

Xiaolong, L.; Lujun, L.; Chao, L.; and Yao, A. 2023. NORM: Knowledge Distillation via N-to-One Representation Matching. In *ICLR*.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.

Yang, C.; Zhou, H.; An, Z.; Jiang, X.; Xu, Y.; and Zhang, Q. 2022. Cross-Image Relational Knowledge Distillation for Semantic Segmentation. *arXiv preprint arXiv:2204.06986*.

Yang, X.; Wang, S.; Luking, L.; and Chen, Y. 2025a. Enabling Visual Foundation Models to Teach Compact Students via Mixture of Distillation. In *IJCAI-25*.

Yang, X.; Yang, Y.; Pang, H.; Tian, A. X.; and Li, L. 2025b. FreqTS: Frequency-Aware Token Selection for Accelerating Diffusion Models. In *AAAI*.

Yang, Z.; Li, Z.; Jiang, X.; Gong, Y.; Yuan, Z.; Zhao, D.; and Yuan, C. 2021. Focal and Global Knowledge Distillation for Detectors. *arXiv preprint arXiv:2111.11837*.

Yang, Z.; Liu, S.; Hu, H.; Wang, L.; and Lin, S. 2019. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9657–9666.

Zagoruyko, S.; and Komodakis, N. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*.

Zhang, Y.; Huang, T.; Liu, J.; Jiang, T.; Cheng, K.; and Zhang, S. 2023. FreeKD: Knowledge Distillation via Semantic Frequency Prompt. arXiv:2311.12079.

Zhao, B.; Cui, Q.; Song, R.; Qiu, Y.; and Liang, J. 2022. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 11953–11962.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.