Enhancing Multi-codebook Vector Quantization for Knowledge Distillation via Multi-layer Supervision and Label Smoothing

Tongtong Zhao, Liangxun Shuo

***Abstract*—This paper focuses on the limitations of single-layer supervision and overconfident one-hot targets in Multi-codebook Vector Quantization (MVQ) for knowledge distillation. To this end, we enhance MVQ by integrating multi-layer supervision and label smoothing. The implementation involves two key steps: during the knowledge extraction phase, knowledge is drawn from multiple teacher layers instead of a single one; during the knowledge transfer phase, label smoothing is applied to the one-hot codebook index targets. Cross-modal experiments on image (CIFAR-100) and speech (AISHELL-1) tasks show that multi-layer supervision and label smoothing can improve student performance in a complementary manner: multi-layer supervision provides a direct and robust gain, whereas the benefit of label smoothing is obtained through careful tuning of its noise parameter. Our work provides a straightforward enhancement for MVQ-based knowledge distillation and suggests that future work could explore dynamic noise scheduling for further performance improvement.**

***Index Terms*—Knowledge distillation, vector quantization, multi-layer distillation, label smoothing**

I. Introduction

KNOWLEDGE distillation (KD) transfers knowledge from large neural networks to compact models, reducing computational and storage costs while maintaining competitive accuracy [1], [2], [3]. It is widely adopted in resource-constrained deployments for speech recognition and image classification [4], [5], [6], [7].

A major practical challenge in offline distillation is the storage overhead of teacher outputs [8], [9]. The Multi-codebook Vector Quantization (MVQ) framework addresses this by compressing teacher representations into compact codebook indexes, achieving high compression ratios while maintaining competitive performance [10]. MVQ-based distillation comprises knowledge extraction and transfer phases, yet current approaches are limited in both: extraction typically employs single-layer supervision, which may limit knowledge completeness, while transfer uses overconfident, noise-sensitive one-hot targets, compromising robustness.

To address these limitations, we explore two enhancements: (1) leveraging multi-layer supervision for more comprehensive knowledge extraction, inspired by FitNets [11], and (2) applying label smoothing [12] to the quantized targets to mitigate overconfidence and enhance transfer robustness.

School of Information Engineering, Hebei GEO University, Graduate Student (2023), Shijiazhuang, China. Email: 230360854001036@hgu.edu.cn

Corresponding author. School of Information Engineering, Hebei GEO University, Shijiazhuang, China. Email: Sshuolx@hgu.edu.cn

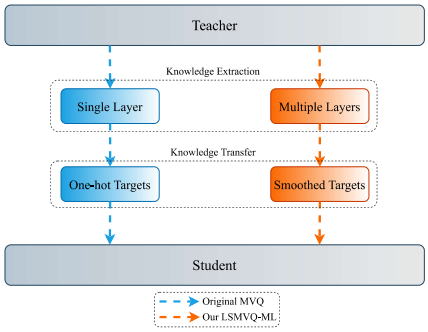


Fig. 1: Comparison between the original MVQ and our proposed LSMVQ-ML framework.

We first extend MVQ to multiple layers and observe consistent performance gains. Surprisingly, during experiments, we discovered that minimal label smoothing ($\epsilon = 0.05$) impairs single-layer distillation but benefits the two-layer setting—a counterintuitive finding that motivated further investigation.

Analysis of the optimization gradients (Fig. 4) reveals that supervisory signals often conflict, pulling parameters in divergent directions. Label smoothing appears to alter these conflict patterns and is empirically correlated with a redistribution of gradient conflicts, rather than functioning solely as a conventional regularizer. We further verify that appropriately tuned smoothing can also improve single-layer distillation.

Our key contributions are:

- Revealing gradient conflicts in multi-source supervision within the MVQ-based distillation and demonstrating that label smoothing correlates with a redistribution of these conflicts.
- Providing evidence that multi-layer supervision can enhance the performance of MVQ-based distillation.
- Suggesting that combining multi-layer supervision and label smoothing can yield stronger improvements and hinting at the characteristics of beneficial noise.

II. Methodology

A. Overall Framework

The proposed Label-Smoothed MVQ for Multi-Layer distillation (LSMVQ-ML) enhances standard MVQ with two key extensions: multi-layer knowledge distillation and label smoothing of the compressed targets. As illustrated in Fig. 2, intermediate teacher features are quantized into 8-bit codebook indexes via a multi-codebook vector quantizer. These indexes

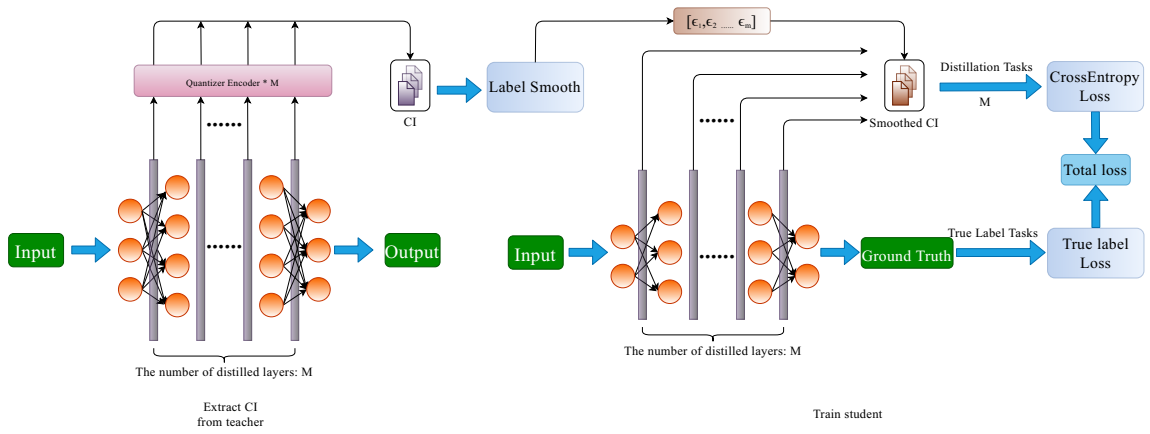


Fig. 2: Pipeline of the LSMVQ-ML framework. Step 1: Extract uint8 codebook indexes (CI) from teacher layers. Step 2: Train the student using a combined loss from smoothed CI targets (with layer-wise smoothing factors) and ground-truth labels.

serve as distillation targets for the student to predict from its corresponding layers, while the student is also supervised by the ground-truth labels.

B. Multi-Codebook Vector Quantization Foundation

LSMVQ-ML builds on the storage-efficient MVQ framework[10]. A multi-codebook quantizer with an encoder-decoder structure compresses teacher representations. Given input x, the encoder produces index sequences $i = \text{Encoder}(x)$, while the decoder reconstructs $\tilde{x} = \text{Decoder}(i)$.

The quantizer, comprising an encoder and a decoder, is trained end-to-end using the following joint loss:

$$\mathcal{L}_{\text{quantizer}} = \underbrace{\|x - \tilde{x}\|_2^2}_{\text{Reconstruction Loss}} + \sum_{n=1}^{N} \underbrace{-\log C_n(x)_{i_n}}_{\text{Prediction Loss}}, \quad (1)$$

where N is the number of codebooks and $C_n(x)_{i_n}$ denotes the predicted probability of index i_n from the n-th codebook.

For distillation, only the encoder is used. Teacher features are quantized into indexes $\text{CI} = \text{Encoder}(x_{\text{teacher}})$, and the student aligns with these targets through a linear prediction head, which is optimized via a cross-entropy (CE) loss:

$$\mathcal{L}_{\text{cb}} = \text{CE}\big(\text{Linear}(\text{Student}_{\text{features}}), \text{CI}_{\text{teacher}}\big). \quad (2)$$

C. Multi-layer Distillation with Smoothed Quantized Targets

The original MVQ uses hard one-hot vectors as targets for the codebook index prediction. We apply label smoothing to soften these targets, producing a smoothed distribution defined as:

$$\text{CI}_j^{\text{smooth}} = \begin{cases} 1-\epsilon, & j = \text{CI}_{\text{true}}, \\ \frac{\epsilon}{K-1}, & \text{otherwise}, \end{cases} \quad (3)$$

where ϵ is the smoothing factor and K is the codebook size.

For single-layer distillation, the loss function is:

$$\mathcal{L}_{\text{cb}}^{\text{LS}} = \text{CE}\big(\text{Linear}(\text{Student}_{\text{features}}), \text{CI}_{\text{teacher}}^{\text{smooth}}\big). \quad (4)$$

In the multi-layer setting, the overall training objective for the student is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \sum_{m=1}^{M} \lambda_m \mathcal{L}_{\text{cb},m}^{\text{LS}} \quad (5)$$

where $\mathcal{L}_{\text{task}}$ denotes the primary task loss, M is the number of distilled layers, and λ_m balances the objectives.

III. Experiments

A. Datasets, Models, and Implementation Details

We evaluate on CIFAR-100 image classification [13] and AISHELL-1 speech recognition [14]. CIFAR-100 consists of 60k 32 x 32 images across 100 classes, split into 50k training and 10k test samples; no augmentation is used to isolate the distillation effect.[1] AISHELL-1 contains 150 hours of Mandarin speech. Following standard practice, we extract 80-dim filter-bank features and apply speed perturbation [15] and MUSAN noise [16], expanding training to 450h.[2] A 30-hour subset is also prepared for low-data evaluation. Teacher-student pairs are chosen to cover varied data-capacity relationships:

- CIFAR-100: ResNet [17] and RegNet [18]:
  - ResNet56 → ResNet8 (sufficient data; 0.08M params)
  - ResNet32x4 → ResNet56 (mildly insufficient data; 0.86M params)
  - RegNetX_400MF → RegNetX_200MF (highly insufficient data; 2.36M params)
- AISHELL-1: Zipformer models [19]:
  - Zipformer-M → Zipformer-XS (30h insufficient, 450h sufficient; 15.06M params)

The following is the detailed information of the teacher model, summarized in Table I:

TABLE I: Teacher Model Information

Teacher Model	Parameters	Performance
ResNet56	0.86M	Acc@1: 73.62%
ResNet32x4	7.43M	Acc@1: 79.54%
RegNetX_400MF	4.81M	Acc@1: 78.88%
Zipformer-M	73.41M	Test CER: 4.67%

For image tasks, models are trained for 240 epochs, with learning rate reduced by a factor of 10 at epochs 150, 180,

[1]Image code: https://github.com/dddmmys/MTKD-RL/tree/LS-MVQ
[2]Speech code: https://github.com/dddmmys/icefall/tree/LS-MVQ

and 210. For speech tasks, low-data models train for 60 epochs, while full-data models are first jointly supervised by both teacher and ground-truth signals for 50 epochs, then by ground-truth only for 20 epochs. MVQ-L1, MVQ-L2, and MVQ-L3 denote the original MVQ with single-, two-, and multi-layer distillation, respectively; LSMVQ-L1, LSMVQ-L2, and LSMVQ-L3 denote their label-smoothed counterparts. Character Error Rate (CER) for speech recognition is obtained via greedy search decoding. Table II details the number of codebooks(N) and the corresponding loss weighting coefficients (λ) configured for each distillation layer setting in both image and speech tasks.

TABLE II: Number of codebooks and loss weighting coefficients per layer setting.

Layer Setting	Image	Speech
L1	N=8, λ=1.0	N=16, λ=0.1
L2	N=[16,8], λ=[1.0,1.0]	N=[16,16], λ=[0.05,0.05]
L3	N=[8,16,8], λ=[1.0,1.0,1.0]*	–

*The L3 configuration was only tested on ResNet8.

All implementations in this study are fully open-sourced.

B. A Counterintuitive Effect of Label Smoothing

Tables III and IV compare the proposed LS-MVQ-ML with several baselines on CIFAR-100 and AISHELL-1 across different data regimes and model capacities. All label smoothing results reported here use a fixed strength of $\epsilon = 0.05$. In the tables, upward arrows (↑) denote improvements attributable to label smoothing, whereas downward arrows (↓) indicate performance drops.

TABLE III: Top-1 accuracy on CIFAR-100 (mean ± std over three runs).

Method	ResNet8	ResNet56	RegNetX_200MF
Baseline	52.51 ± 0.29	62.25 ± 0.54	69.30 ± 0.11
MVQ-L1	55.01 ± 0.15	64.77 ± 0.10	72.11 ± 0.43
MVQ-L2	55.63 ± 0.42	66.54 ± 0.40	74.61 ± 0.25
LSMVQ-L1	54.64 ± 0.25↓	64.75 ± 0.55↓	72.67 ± 0.23↑
LSMVQ-L2	55.85 ± 0.21↑	66.51 ± 0.24↓	74.88 ± 0.06↑

TABLE IV: Character error rate (%) on AISHELL-1 under low-data and full-data regimes. Low-data results are mean ± std over three runs (Test/Dev), full-data results are from a single run.

Method	Low-Data (30h) Test / Dev (%)	Full-Data (450h) Test / Dev (%)
Baseline	19.15 ± 0.26 / 17.80 ± 0.08	6.49 / 5.95
MVQ-L1	14.55 ± 0.10 / 13.54 ± 0.05	6.08 / 5.57
MVQ-L2	12.97 ± 0.01 / 12.19 ± 0.04	6.01 / 5.57
LSMVQ-L1	14.43 ± 0.09 / 13.44 ± 0.07↑	6.11 / 5.55↓
LSMVQ-L2	13.00 ± 0.03 / 12.14 ± 0.07↓	5.96 / 5.54↑

To quantitatively analyze the impact of label smoothing, Table V lists the performance gains of both MVQ and LS-MVQ relative to the no-distillation baseline.

TABLE V: Performance gains across configurations ordered by increasing data-to-capacity ratio.

Setting Data/Cap.	RegNetX_200MF (High Insuf.)	Low-Data (Insuf.)	ResNet56 (Mild Insuf.)	ResNet8 (Suf.)	Full-Data (Suf.)
MVQ-L1	2.81	4.60	2.52	2.50	0.41
LSMVQ-L1	3.37	4.72	2.50	2.13	0.38
MVQ-L2	5.31	6.18	4.29	3.12	0.48
LSMVQ-L2	5.58	6.15	4.26	3.34	0.53

Figure 3 plots the performance ratio LSMVQ/MVQ across configurations ordered by increasing data-to-capacity ratio. A ratio above 1 indicates that label smoothing improves performance over the MVQ baseline.

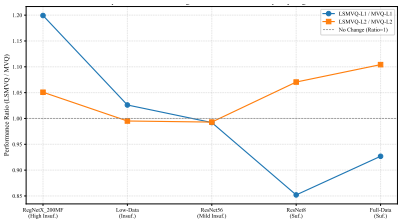


Fig. 3: Performance ratio (LSMVQ/MVQ) across data-to-capacity regimes.

The effect of label smoothing depends on both distillation depth and data-to-capacity regime: for single-layer distillation (L1), it transitions from beneficial to detrimental as the data-to-capacity ratio increases; for two-layer distillation (L2), the trend is more complex, showing benefits in low- and high-data regimes but slight harm in medium regimes.

C. A Case Study: What Label Smoothing Does

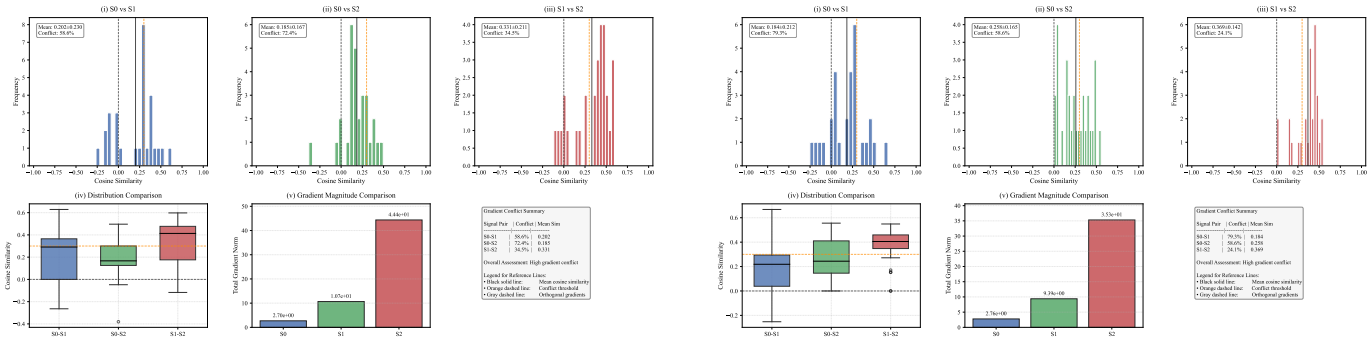
To understand what this beneficial cross-modal label smoothing affects, we analyze gradient directions using the ResNet8 configuration under the two-layer distillation setting. A pre-trained ResNet8 model is loaded, selecting the checkpoint from the run closest to the mean performance across three independent runs for both settings (with and without label smoothing). A forward pass is then performed on a batch of data to measure the degree of gradient direction conflict between the supervisory signals.

Gradient analysis (Fig. 4) reveals persistent conflicts among supervisory signals. Label smoothing affects these conflicts: it slightly alleviates inter-teacher conflicts (S0–S1) while mildly amplifies the conflicts between teacher and ground-truth signals (S0–S2 and S1–S2). This adjustment is accompanied by a more concentrated distribution of S0–S2 similarities, more dispersed distributions of S0–S1 and S1–S2 similarities, and a shift of the most concentrated distribution from S1–S2 to S0–S2, which collectively help unify the supervisory directions. Additionally, label smoothing increases the relative dominance of the ground-truth signal.

This observed relationship prompts a further question: can we systematically reproduce the beneficial state by tuning the noise strength?

D. In-depth Exploration: Layer-wise Smoothing Allocation

To verify the broader prevalence of the performance gains from introducing noise, rather than its effect under specific



(a) Pretrain w/ LS (b) Pretrain w/o LS

Fig. 4: Gradient direction conflict analysis in the pre-trained model: (a) with label smoothing, (b) without label smoothing. Each panel contains six sub-figures: (i-iii) histograms of cosine similarities between gradients of three supervisory signals—S0 (shallow teacher), S1 (deep teacher), and S2 (ground truth); (iv) box plots of these similarities; (v) gradient magnitude comparisons; and a summary text.

configurations, we conduct a series of three-layer distillation experiments on ResNet8 (CIFAR-100). Table VI presents the results for various smoothing allocations, where the ϵ values are listed from the shallowest to the deepest distillation target (e.g., $\epsilon = [0.05, 0, 0.05]$ applies $\epsilon = 0.05$ to the shallowest and deepest targets, and no smoothing to the middle target).

TABLE VI: Performance of LSMVQ-L3 with different layer-wise label smoothing allocations on ResNet8 (CIFAR-100).

Setting	Mean $\pm$ std
MVQ-L3	55.85 $\pm$ 0.17
LSMVQ-L3 ϵ=0.05 (all)	55.25 $\pm$ 0.34
LSMVQ-L3 ϵ=[0.05, 0, 0.05]	55.50 $\pm$ 0.25
LSMVQ-L3 ϵ=[0, 0.05, 0.05]	55.39 $\pm$ 0.38
LSMVQ-L3 ϵ=[0.05, 0.05, 0]	55.80 $\pm$ 0.26
LSMVQ-L3 ϵ=[0.05, 0.03, 0.01]	55.88 $\pm$ 0.21

Uniform smoothing ($\epsilon = 0.05$ on all three layers) reduces accuracy compared to no smoothing. The decreasing sequence $\epsilon = [0.05, 0.03, 0.01]$ achieves the highest accuracy, surpassing both the no-smoothing and uniform-smoothing baselines.

TABLE VII: Revisiting single-layer distillation with tuned label smoothing on ResNet8 (CIFAR-100) and Full-Data (AISHELL-1).

Task	Setting	Performance
ResNet8	MVQ-L1	55.01 $\pm$ 0.15
	LSMVQ-L1 $\epsilon = 0.05$	54.64 $\pm$ 0.25
	LSMVQ-L1 $\epsilon = 0.025$	55.54 $\pm$ 0.26
Full-Data	MVQ-L1	6.08 / 5.57
	LSMVQ-L1 $\epsilon = 0.05$	6.11 / 5.55
	LSMVQ-L1 $\epsilon = 0.025$	6.13 / 5.64
	LSMVQ-L1 (dynamic strategy)	6.02 / 5.60

We also revisit single-layer distillation to examine whether a tuned smoothing strength can improve performance. On CIFAR-100 with ResNet8, we observe that a small smoothing strength of $\epsilon = 0.025$ leads to a measurable improvement over the no-smoothing baseline. However, in the full-data AISHELL-1 speech task, static smoothing strengths (e.g., $\epsilon = 0.025$) in the single-layer setting do not surpass the no-smoothing baseline. Instead, a dynamic smoothing strategy—training with $\epsilon = 0.05$ for 30 epochs, then $\epsilon = 0.025$ for 20 epochs, followed by 20 epochs using only ground-truth labels—achieves a test CER of 6.02, outperforming both fixed-smoothing and no-smoothing baselines. These results are summarized in Table VII. Across these configurations, beneficial noise persists; when combined with the gain analysis presented in Table V, this demonstrates a generally observable phenomenon.

IV. Discussion and Conclusion

Our investigation demonstrates that the Multi-codebook Vector Quantization (MVQ) distillation framework is effectively enhanced by two primary modifications. First, incorporating multi-layer teacher supervision yields direct and consistent performance gains across image and speech tasks (Tables III and IV). Second, label smoothing interacts with student generalization in a more complex manner. Our gradient analysis (Fig. 4) reveals that this introduced noise measurably redistributes conflicts between the supervisory signals, providing a clue that its conditional benefit may hinge on the noise strength. Building on this observation, our exploration of smoothing schedules (Tables VI and VII) indicates that beneficial noise is a general occurrence. Furthermore, the results suggest an optimal noise pattern characterized by a layer-wise decreasing sequence $\epsilon = [\epsilon_1, \epsilon_2, ..., \epsilon_m]$ (where $\epsilon_1 \geq \epsilon_2 \geq ... \geq \epsilon_m$) whose overall strength decays during training. This layer-wise pattern stems from the observation that shallower signals, being influenced by a greater number of deeper supervisory paths, require stronger smoothing to harmonize the multi-source guidance. The temporal decay aligns with the principle of maintaining a beneficial teacher-student capacity gap [20]: stronger noise early helps bridge a wider gap, while reduced noise later facilitates finer alignment. Future work could focus on designing adaptive ϵ scheduling strategies to further optimize this gap management.

In conclusion, this work empirically validates two effective pathways to enhance MVQ-based knowledge distillation: utilizing multi-layer supervision and introducing calibrated noise into the quantized targets. These findings provide practical guidance for enhancing learning from compressed knowledge, contributing to the future deployment of knowledge distillation systems on resource-constrained devices.

REFERENCES

[1] G. E. Hinton, O. Vinyals, J. Dean, "Distilling the Knowledge in a Neural Network," 2015, *arXiv:1503.02531* .

[2] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, Q. Liu, "TinyBERT: Distilling BERT for Natural Language Understanding," in *Proc. EMNLP Findings*, 2019.

[3] S. Panchapagesan, D. S. Park, C.-C. Chiu, Y. Shangguan, Q. Liang, and A. Gruenstein, "Efficient Knowledge Distillation for RNN-Transducer Models," in *Proc. IEEE ICASSP*, 2021, pp. 5639–5643.

[4] K. Zhao, H. Nguyen, A. Jain, N. Susanj, A. Mouchtaris, L. Gupta, and M. Zhao, "Knowledge Distillation via Module Replacing for Automatic Speech Recognition with Recurrent Neural Network Transducer," in *Proc. Interspeech*, 2022, pp. 4436-4440.

[5] S. Tian, K. Deng, Z. Li, L. Ye, G. Cheng, T. Li, and Y. Yan, "Knowledge Distillation for CTC-Based Speech Recognition via Consistent Acoustic Representation Learning," in *Proc. Interspeech*, 2022, pp. 2633-2637.

[6] Z. Yang, Z. Li, A. Zeng, Z. Li, C. Yuan, and Y. Li, "ViTKD: Feature-Based Knowledge Distillation for Vision Transformers," in *Proc. IEEE CVPRW*, 2024, pp. 1379-1388.

[7] D. Chen, J.-P. Mei, H. Zhang, C. Wang, Y. Feng, and C. Chen, "Knowledge Distillation with the Reused Teacher Classifier," in *Proc. IEEE CVPR*, 2022, pp. 11923-11932.

[8] R. Yu, S. Liu, Z. Chen, J. Ye, and X. Wang, "Heavy Labels Out! Dataset Distillation with Label Space Lightening," in *Proc. IEEE ICCV*, 2025, pp. 5017–5026.

[9] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A survey," *Int. J. Comput. Vis.*, 2021, vol. 129, pp. 1789–1819.

[10] L. Guo, X. Yang, Q. Wang, Y. Kong, Z. Yao, F. Cui, F. Kuang, W. Kang, L. Lin, M. Luo, P. Zelasko, D. Povey, "Predicting Multi-Codebook Vector Quantization Indexes for Knowledge Distillation," in *Proc. ICASSP*, 2023, pp. 1–5.

[11] A. Romero, N. Ballas, S. Ebrahimi Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for Thin Deep Nets," 2015, *arXiv:1412.6550*.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proc. IEEE CVPR*, 2016, pp. 2818–2826.

[13] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," Tech. Rep. TR-2009, University of Toronto, 2009.

[14] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An Open-Source Mandarin Speech Corpus and A Speech Recognition Baseline," in *Proc. O-COCOSDA*, 2017, pp. 1–5.

[15] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586-3589.

[16] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, *arXiv:1510.08484*.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.

[18] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing Network Design Spaces," in *Proc. IEEE CVPR*, 2020, pp. 10425-10433.

[19] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang, Y. Yang, Z. Jin, L. Lin, and D. Povey, "Zipformer: A faster and better encoder for automatic speech recognition," in *Proc. ICLR*, 2024.

[20] S. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved Knowledge Distillation via Teacher Assistant," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, pp. 5191–5198, 2020.