Learning What to Trust: Confidence-Aware Curriculum Distillation for TinyML Vision

Satarupa Das*, Soumen Garai*, Rajrup Saha*, Soumya Chatterjee†, and Suman Samui*
*Dept. of Electronics and Communication Engineering,
†Dept. of Electrical Engineering
National Institute of Technology Durgapur, West Bengal 713209, India

***Abstract*—Deploying accurate vision models on microcontroller-class TinyML devices requires operating under sub-megabyte Flash budgets, hard SRAM constraints, and fine-grained visual ambiguity. We propose Confidence-Aware Curriculum Knowledge Distillation (CL-KD), a two-stage framework that jointly optimizes student design and training. Stage I formulates architecture selection as a constrained bi-objective problem—maximising accuracy while minimising Flash under an SRAM bound—solved via Multi-Objective Bayesian Optimization (MOBO) with expected hypervolume improvement, yielding Pareto-optimal students across sub-100,KB to MB-scale regimes. Stage II introduces confidence-aware curriculum distillation, where Monte Carlo dropout estimates per-sample uncertainty to induce difficulty-aware training, and a novel sample-wise dynamic wisdom ratio adaptively balances soft and hard supervision based on teacher confidence and student capacity. CL-KD subsumes standard KD and supervised learning as limiting cases. On a seven-class hydrophobicity dataset, we observe accuracy improvements of up to +20.14 percentage points for highly constrained models and average improvements of +9.65 percentage points with CL-KD, while satisfying stringent TinyML deployment constraints.**

***Index Terms*—Knowledge Distillation, TinyML, Curriculum Learning, Monte Carlo Dropout, Pareto Optimization, Hydrophobicity Classification, Composite Insulators.**

I. Introduction

Deploying deep computer-vision models at the edge is a central challenge in modern machine learning. While server-class networks routinely exceed human-level accuracy on tasks such as image classification, object detection, and semantic segmentation, their computational and memory footprints render them incompatible with microcontroller-class hardware. TinyML platforms impose hard resource ceilings—Flash memory $< 4\,\text{MB}$, SRAM $256\,\text{KB}$–$1\,\text{MB}$, milliwatt-class power [1], and often fewer than 100K parameters—precluding direct deployment of off-the-shelf architectures and necessitating joint optimization of model design and training [2], [3].

Knowledge Distillation (KD) [4] is a natural compression pathway: a compact *student* $f_{\boldsymbol{\theta}}$ learns to mimic the softened outputs of a larger *teacher* $g_{\boldsymbol{\phi}}$. However, the standard formulation suffers from two failure modes exacerbated under extreme capacity constraints:

1) **Uniform supervision:** A fixed mixing coefficient α weights hard-label and soft-label losses identically for every sample, ignoring that teacher confidence varies dramatically. When the teacher is uncertain, its softened logits inject noise that can dominate the limited gradient capacity of a tiny student, sometimes *degrading* accuracy below the no-KD baseline.
2) **Decoupled architecture search:** Student topology is typically selected by manual trial-and-error, without formal consideration of the accuracy–memory trade-off surface.

These failure modes are especially severe in fine-grained visual recognition tasks where inter-class boundaries are inherently ambiguous. We ground our study in one such domain: *hydrophobicity classification of silicone rubber insulators:* Silicone rubber insulators are widely used in high-voltage networks due to their low weight, mechanical robustness, and intrinsic hydrophobicity. However, environmental stressors—UV radiation, ageing, humidity, and partial discharge—progressively degrade surface hydrophobicity [5], increasing leakage current and flashover risk [6]. The IEC 62073 standard [6] defines seven ordinal classes (HC1–HC7), where adjacent grades differ by subtle contact-angle variations that are visually near-indistinguishable. Image-based approaches—graph-based methods [7], Swin-Transformer models [8], and morphology-driven CNNs [9]—have shown promising accuracy but none address deployment on resource-constrained embedded platforms.

We propose **CL-KD**, a unified two-stage framework:

- **S-I. Hardware-aware Pareto student selection:** via constrained multi-objective Bayesian optimization, yielding deployable architectures with formal optimality guarantees (Section IV-A).
- **S-II. Confidence-aware curriculum distillation:** using MC-dropout to quantify per-sample teacher uncertainty, partition data into difficulty tiers, and introduce a *dynamic wisdom ratio* $\alpha_s(\mathbf{x})$ (Section IV-B).

Major Contributions of this work are given as follows:

- A bi-objective MOBO formulation for TinyML NAS with explicit SRAM constraints and augmented Tchebycheff selection.
- An MC-dropout confidence estimator assigning difficulty scores *without modifying the trained teacher*.
- A dynamic sample-wise wisdom ratio $\alpha_s(\mathbf{x})$ that provably reduces to standard KD and pure supervised learning as limiting cases (Proposition 21).

II. Related Work

Knowledge Distillation: Hinton *et al.* [4] introduced logit-based knowledge distillation via temperature-scaled softmax. Subsequent works explore feature-level alignment [10] and dynamic teacher–student coupling [11]. However, these approaches assume a fixed student architecture and uniform sample importance.

Hardware-Aware Neural Architecture Search: DeepMaker [12] formulates NAS as multi-objective optimization over embedded hardware costs. We extend this paradigm to TinyML by incorporating explicit SRAM constraints and coupling Pareto-optimal architectures with a confidence-aware distillation pipeline.

III. Problem Formulation and Background

Definition 1 (Learning setting)**.** Let $\mathcal{X} \subseteq \mathbb{R}^{H\times W\times C}$ denote the input space of $H\times W$ RGB images and $\mathcal{Y} = \{1, \ldots, K\}$ the label space with $K{=}7$ hydrophobicity classes. The full dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ is partitioned into $\mathcal{D} = \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{val}} \cup \mathcal{D}_{\text{te}}$.

Definition 2 (Teacher and student)**.** The *teacher* $g_{\boldsymbol{\phi}} : \mathcal{X} \to \mathbb{R}^K$ with $\boldsymbol{\phi} \in \mathbb{R}^{d_T}$ is trained to convergence. A *student* $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^K$ has $\boldsymbol{\theta} \in \mathbb{R}^{d_S}$, $d_S \ll d_T$. Normalised predictions are

$$\mathbf{p}(\mathbf{x};\boldsymbol{\theta}) = \operatorname{softmax}\big(f_{\boldsymbol{\theta}}(\mathbf{x})\big) \in \Delta^{K-1}. \tag{1}$$

Definition 3 (Temperature-scaled softmax)**.** For temperature $T > 0$:

$$p_k^{(T)}(\mathbf{x};\boldsymbol{\theta}) = \frac{\exp\big(z_k/T\big)}{\sum_{j=1}^{K}\exp\big(z_j/T\big)}, \quad k = 1,\dots,K, \tag{2}$$

where z_k is the k-th logit.

Definition 4 (Standard KD loss [4])**.**

$$\ell_{\mathrm{KD}}(\mathbf{x}, y; \alpha, T) = \alpha\,\ell_{\mathrm{CE}}\big(y, \mathbf{p}_s\big) + (1-\alpha)\,T^2\ \mathrm{KL}\big(\mathbf{p}_t^{(T)} \big\| \mathbf{p}_s^{(T)}\big), \tag{3}$$

where $\alpha \in [0,1]$ is the *wisdom ratio*, ℓ_{CE} is cross-entropy with one-hot labels, and T^2 compensates for gradient magnitude reduction.

IV. Proposed Framework

The overall CL-KD framework is illustrated in Fig. 1. Stage I searches for Pareto-optimal TinyML student architectures under hardware constraints via constrained MOBO [13]. Stage II applies the teacher's MC-dropout confidence estimates to construct a three-tier curriculum ($\mathcal{D}_{\mathrm{easy}}, \mathcal{D}_{\mathrm{med}}, \mathcal{D}_{\mathrm{hard}}$) and distils knowledge into each selected student S_i using the dynamic wisdom ratio, producing improved students S_i'.

A. Stage I: Hardware-Aware Pareto Student Selection

1) Constrained Bi-Objective Formulation: A student architecture is specified by $\mathbf{h} \in \mathcal{H}$. Optimal weights solve:

$$\boldsymbol{\theta}^*(\mathbf{h}) \in \arg\min_{\boldsymbol{\theta}} \frac{1}{|\mathcal{D}_{\mathrm{tr}}|} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{tr}}} \ell_{\mathrm{CE}}\big(y, \mathbf{p}(\mathbf{x};\boldsymbol{\theta},\mathbf{h})\big). \tag{4}$$

Two upper-level objectives on $\mathcal{D}_{\mathrm{val}}$:

Definition 5 (Accuracy objective)**.**

$$f_1(\mathbf{h}) = -\frac{1}{|\mathcal{D}_{\mathrm{val}}|} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{val}}} \mathbb{1}\big[\arg\max_k p_k = y\big]. \tag{5}$$

Definition 6 (Flash memory objective)**.**

$$f_2(\mathbf{h}) = \mathrm{Flash}(\mathbf{h}) = \sum_{\ell=1}^{L(\mathbf{h})} d_\ell(\mathbf{h})\cdot b_\ell + C_{\mathrm{code}}(\mathbf{h}), \tag{6}$$

where d_ℓ is the parameter count of layer ℓ, b_ℓ the per-parameter byte cost, and C_{code} the compiled code size.

Definition 7 (SRAM constraint)**.** Peak SRAM during inference:

$$\mathrm{RAM}(\mathbf{h}) = \max_t \big[\mathrm{act}(\mathbf{h},t) + \mathrm{ws}(\mathbf{h},t) + \mathrm{buf}(\mathbf{h},t)\big], \tag{7}$$

with deployability requiring $g(\mathbf{h}) \coloneqq \mathrm{RAM}(\mathbf{h}) - B_{\mathrm{RAM}} \le 0$.

The constrained bi-objective programme is:

$$\boxed{\min_{\mathbf{h}\in\mathcal{H}}\ \mathbf{f}(\mathbf{h}) = \big(f_1(\mathbf{h}), f_2(\mathbf{h})\big)\ \text{ s.t. }\ g(\mathbf{h}) \le 0.} \tag{8}$$

Definition 8 (Pareto optimality)**.** $\mathbf{h}$ *dominates* $\mathbf{h}'$, $\mathbf{f}(\mathbf{h}) \prec \mathbf{f}(\mathbf{h}')$, if $f_k(\mathbf{h}) \le f_k(\mathbf{h}')$ for all k with strict inequality for some k. The Pareto set is $\mathcal{P} = \{\mathbf{h} \in \mathcal{H}_{\mathrm{feas}} \mid \nexists\, \mathbf{h}' : \mathbf{f}(\mathbf{h}') \prec \mathbf{f}(\mathbf{h})\}$.

2) Multi-Objective Bayesian Optimization: Each objective is modelled by an independent GP:

$$f_k(\mathbf{h}) \sim \mathcal{GP}\big(\mu_k(\mathbf{h}), \kappa_k(\mathbf{h},\mathbf{h}')\big), \tag{9}$$

with a Matérn-$5/2$ kernel and ARD. The acquisition function is expected hypervolume improvement (EHVI) [14]:

$$\alpha_{\mathrm{EHVI}}(\mathbf{h}) = \mathbb{E}\big[\mathrm{HV}(\mathcal{F}_t \cup \{\mathbf{f}(\mathbf{h})\}, \mathbf{r}) - \mathrm{HV}(\mathcal{F}_t, \mathbf{r})\big]. \tag{10}$$

The SRAM constraint is incorporated via feasibility weighting:

$$\tilde{\alpha}(\mathbf{h}) = \alpha_{\mathrm{EHVI}}(\mathbf{h}) \cdot \mathbb{P}\big[g(\mathbf{h}) \le 0 \mid \mathcal{D}_t\big]. \tag{11}$$

After T iterations, the Pareto set is extracted:

$$\mathcal{P} = \big\{\mathbf{h} \in \mathcal{D}_T : \nexists\, \mathbf{h}' \colon \mathbf{f}(\mathbf{h}') \prec \mathbf{f}(\mathbf{h}),\ g(\mathbf{h}) \le 0\big\}. \tag{12}$$

3) Architecture Search Space: Each student is specified by:

$$\mathbf{h} = (L, F, K, S, D, B, U) \in \mathcal{H}, \tag{13}$$

where $L \in \{1,\dots,4\}$ (conv. stages), $F \in \{16, 32, 64, 128\}$ (filters), $K \in \{3, 5\}$ (kernel), S (stride), $D \in [0, 0.5]$ (dropout), $B \in \{0, 1\}$ (batch-norm), $U \in \{1, 2, 3\}$ (dense multiplier).

4) Augmented Tchebycheff Selection: Let $\mathbf{f}^* = (f_1^*, f_2^*)$ be the ideal point over $\mathcal{P}$.

Definition 9 (Tchebycheff score)**.** For $\mathbf{w} = (w_1, w_2)$ with $w_k > 0$, $\sum_k w_k = 1$:

$$J(\mathbf{h};\mathbf{w}) = \max_k\ w_k|f_k(\mathbf{h}) - f_k^*| + \rho\textstyle\sum_k w_k|f_k(\mathbf{h}) - f_k^*|, \tag{14}$$

where $\rho > 0$ ensures unique solutions on non-convex regions.

Proposition 10 (Pareto optimality)**.** *For any* $\mathbf{w}$ *with* $w_k > 0$ *and* $\rho > 0$*, every minimiser of* $J(\mathbf{h};\mathbf{w})$ *is Pareto-optimal for* (8)*.*

B. Stage II: Confidence-Aware Curriculum KD

Stage II takes a trained teacher $g_{\boldsymbol{\phi}}$ and a Pareto student $\mathbf{h} \in \mathcal{P}$, producing a distilled student via three mechanisms.

1) MC-Dropout Teacher Confidence:

Definition 11 (Stochastic forward pass)**.** With dropout active at rate p, M stochastic passes yield:

$$\big\{\mathbf{p}_t^{(m)}(\mathbf{x}) = \mathrm{softmax}\big(g_{\boldsymbol{\phi}}^{(m)}(\mathbf{x})\big)\big\}_{m=1}^{M}. \tag{15}$$

Definition 12 (Predictive uncertainty)**.** Mean class-wise variance:

$$v(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} \mathrm{Var}_m\big[p_{t,k}^{(m)}(\mathbf{x})\big]. \tag{16}$$

Definition 13 (Confidence score)**.** Regularised inverse variance:

$$c(\mathbf{x}) = \frac{1}{v(\mathbf{x}) + \varepsilon}, \tag{17}$$

where $\varepsilon > 0$ ensures numerical stability.

Remark 14. $v(\mathbf{x})$ is monotonically related to predictive entropy $\mathcal{H}[y|\mathbf{x}]$ under the MC-dropout approximate posterior.

Lemma 15 (Confidence bounds)**.** *For* K*-class classification with* $\varepsilon > 0$*:* $\frac{1}{1/(4K)+\varepsilon} \le c(\mathbf{x}) \le 1/\varepsilon$.

Proof. The variance of $p_{t,k}^{(m)} \in [0,1]$ is bounded by $1/4$. Averaging over K classes: $v(\mathbf{x}) \le 1/(4K)$, giving the lower bound on $c(\mathbf{x})$. The upper bound follows from $v(\mathbf{x}) \ge 0$. □

2) Curriculum Partitioning: Confidence scores are computed once before training. Quantile thresholds $\tau_1 = Q_{1/3}$, $\tau_2 = Q_{2/3}$ partition data:

Definition 16 (Difficulty tiers)**.**

$$\mathcal{D}_{\mathrm{easy}} = \big\{(\mathbf{x},y) : c(\mathbf{x}) \ge \tau_2\big\}, \tag{18}$$

$$\mathcal{D}_{\mathrm{med}} = \big\{(\mathbf{x},y) : \tau_1 \le c(\mathbf{x}) < \tau_2\big\}, \tag{19}$$

$$\mathcal{D}_{\mathrm{hard}} = \big\{(\mathbf{x},y) : c(\mathbf{x}) < \tau_1\big\}. \tag{20}$$

Training follows the staged schedule:

$$\mathcal{D}_{\mathrm{easy}} \xrightarrow{\text{converge}} \mathcal{D}_{\mathrm{med}} \xrightarrow{\text{converge}} \mathcal{D}_{\mathrm{hard}}. \tag{21}$$

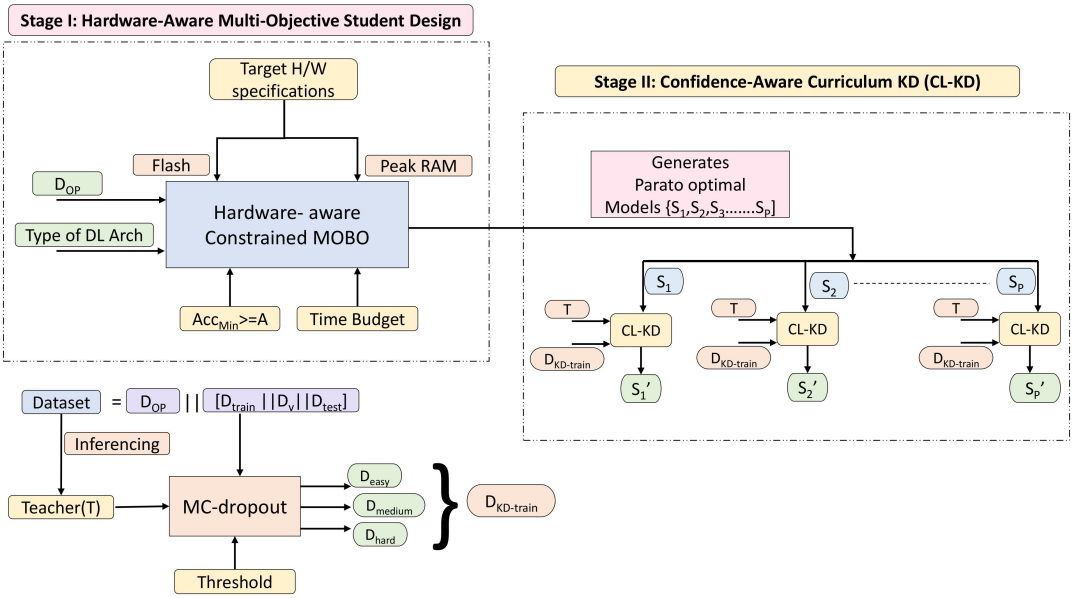

Fig. 1: Overview of the proposed two-stage CL-KD framework: Stage I selects Pareto-optimal TinyML students via constrained MOBO; Stage II distils knowledge through an MC-dropout confidence-driven curriculum into each Pareto student S_i yielding an improved student S_i'.

3) Dynamic Sample-Wise Wisdom Ratio:

Definition 17 (Normalised student capacity)**.**

$$\mu_s = \frac{M_s - M_{\min}}{M_{\max} - M_{\min}} \in [0, 1]. \tag{22}$$

Definition 18 (Size modulation)**.**

$$\beta(\mu_s) = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\,\mu_s. \tag{23}$$

Definition 19 (Confidence modulation)**.**

$$\eta(\mathbf{x}) = \frac{\gamma}{c(\mathbf{x}) + \varepsilon_c}, \tag{24}$$

where $\gamma > 0$ is a scaling constant.

Definition 20 (Dynamic wisdom ratio)**.**

$$\boxed{\alpha_s(\mathbf{x}) = \mathrm{clip}\Big(\beta(\mu_s)\cdot\eta(\mathbf{x}),\ 0,\ 1\Big).} \tag{25}$$

Proposition 21 (Boundary recovery)**.** *The dynamic wisdom ratio satisfies:*

(i) ***Standard KD:*** $\mu_s{=}1$, $c(\mathbf{x}){\to}\infty \Rightarrow \alpha_s{\to}0$ *(pure soft-label KD).*
(ii) ***Supervised learning:*** $\mu_s{=}0$, $c(\mathbf{x}){\to}0^+ \Rightarrow \alpha_s{\to}1$ *(pure cross-entropy).*

4) CL-KD Training Objective: The per-sample CL-KD loss:

$$\ell_{\text{CL-KD}}(\mathbf{x}, y; s) = \alpha_s\,\ell_{\text{CE}}(y, \mathbf{p}_s) + (1{-}\alpha_s)\,T^2\ \mathrm{KL}(\mathbf{p}_t^{(T)} \| \mathbf{p}_s^{(T)}), \tag{26}$$

where the MC-averaged teacher prediction is $\bar{\mathbf{p}}_t^{(T)} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{p}_t^{(T,m)}$. Algorithm 1 summarises Stage II.

Algorithm 1 CL-KD Training Procedure (Stage II)

Require: Teacher $g_{\boldsymbol{\phi}}$; student $\mathbf{h} \in \mathcal{P}$ with μ_s; $\mathcal{D}_{\text{tr}}$; M MC passes; temperature T; $\alpha_{\min}, \alpha_{\max}, \gamma, \varepsilon, \varepsilon_c$
Ensure: Trained student $f_{\boldsymbol{\theta}^*}$

```
// Phase 1: Confidence Scoring
for each (x_i, y_i) ∈ D_tr do
    Run M stochastic passes; compute v(x_i) via (16)
    c(x_i) ← 1/(v(x_i) + ε)
end for
Partition into D_easy, D_med, D_hard
// Phase 2: Curriculum Distillation
for s ∈ {easy, med, hard} do
    for each epoch do
        for each mini-batch B_s ⊂ D_s do
            Compute α_s(x) via (25); ℓ via (26)
            θ ← θ − η∇_θ (1/|B_s|) Σ ℓ
        end for
        if early stopping on D_val then break
        end if
    end for
end for
return f_θ
```

V. Experimental Result

Case Study: Hydrophobicity Classification

Rationale: Composite-insulator hydrophobicity classification provides a stringent benchmark for evaluating CL-KD:

1) **Fine-grained ordinal ambiguity:** The seven IEC 62073 classes [6] differ by subtle contact-angle variations; adjacent grades (HC3–HC5) are visually near-indistinguishable, producing high teacher uncertainty on boundary samples. Moreover, Dataset's *Fisher Discriminant Ratio* = 1.218, indicates moderate global separability with significant intra-class spread.
2) **Safety-critical deployment:** Thousands of insulators across transmission corridors require autonomous on-device inference—mandating TinyML-class models [6].

Dataset: We use the publicly available hydrophobicity dataset [5] (IEC Standard 62073 spray method, Mendeley Data[1]). It contains ≈4,500 images across HC1–HC7. Class-balanced split: training

[1]https://data.mendeley.com/datasets/w2pg3k6c8m/1

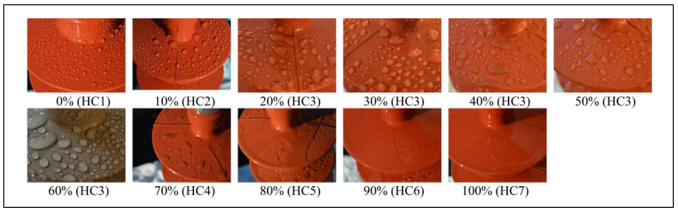

Fig. 2: Representative images for hydrophobicity classes HC1–HC7. Visual differences are coarse for extreme classes but subtle for intermediate grades (HC3–HC5) [5].

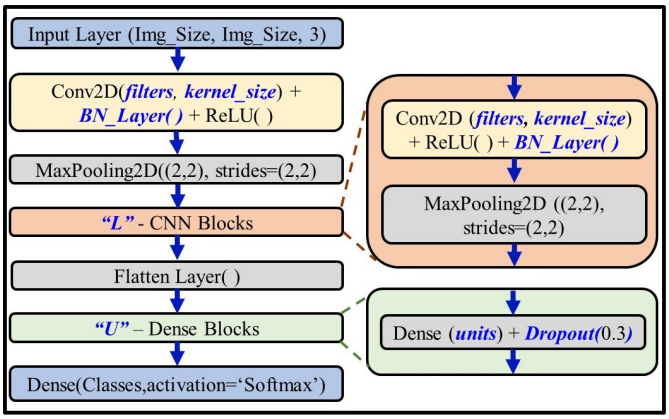

Fig. 3: Parameterised student CNN backbone. Searched hyper-parameters $\{L, F, K, U, B, D\}$ are annotated.

(400×7 = 2,800), validation (100×7 = 700), test (122–165 per class, ≈980 total). Sample images for each class HC1–HC7 are shown in Fig. 2.

The teacher model is a convolutional neural network featuring multiple Conv2D, Batch Normalization, and pooling layers with a total size of about 15.8 MB. It achieves strong performance, with around 95% validation accuracy, 93.67% test accuracy, and a macro F1-score of approximately 0.90 across seven classes. The student backbone (refer to Fig.3) is a customised lightweight CNN built from standard and optionally depth-wise separable convolutions, drastically reducing multiply–accumulate operations (MACs). **Training Details:** *Stage I (MOBO)* employs $T = 240$ iterations with $n_0 = 10$ random initialisations, EHVI acquisition, and an ARD Mat'ern-5/2 kernel, with experiments conducted on an NVIDIA RTX A4000 and Xeon W-2245 system. *Stage II (CL-KD)* uses the Adam optimizer (lr $= 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$), batch size 64, and up to 300 epochs per curriculum stage with early stopping (patience 10); the distillation temperature is set to $T = 4$, MC-dropout confidence scores are computed once using $N = 50$ stochastic forward passes, and hyperparameters γ, $\alpha_{\min}$, and $\alpha_{\max}$ are tuned on $\mathcal{D}_{\text{val}}$.

Figure 4 shows the Pareto front from MOBO. Table I reports each Pareto model with its accuracy, size, TS score, and hyper-parameter configuration. Five representative students are selected using equal weights $w_1 = w_2 = 0.5$: three *small* (sub-1 MB), one *medium* (0.4–2.5 MB), one *large* (∼12 MB). Table I reports the five Pareto students spanning 0.09 MB to 12.27 MB. Table II compares Baseline, Standard KD ($\alpha = 0.5$), and CL-KD.

We observe four key findings: (1) **consistent superiority**, as CL-KD outperforms both baseline and standard KD across all student models; (2) **pronounced gains for ultra-small models**, where for Small 2 (0.09 MB, 11K parameters) standard KD degrades accuracy by 5.85 pp below baseline, while CL-KD recovers a +15.85 pp improvement

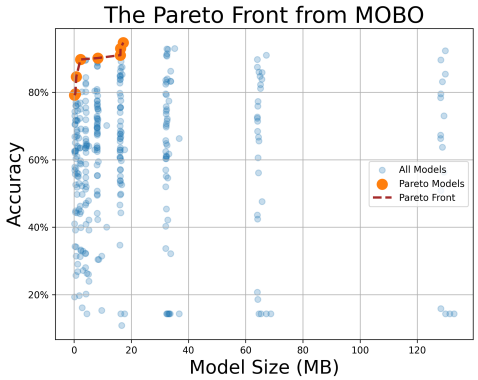

Fig. 4: Pareto front from MOBO architecture search. Orange curve: Pareto frontier; five selected students annotated.

TABLE I: Pareto-optimal models obtained from MOBO

Model	Acc (%)	Size (MB)	TS score	h=$\{L, F, K, U, B, D\}$
MOBO_0 (S1)	84.6	0.77	0.339	$\{4, 32, (3, 3), 3, 1, 0\}$
MOBO_91 (S2)	79.1	0.09	0.079	$\{4, 16, (3, 3), 1, 1, 0\}$
MOBO_112(S3)	79.6	0.43	0.169	$\{4, 32, (5, 5), 1, 1, 0\}$
MOBO_25 (M1)	89.7	2.21	1.059	$\{4, 128, (3, 3), 1, 1, 0\}$
MOBO_236(L1)	94.7	17.19	8.549	$\{4, 64, (5, 5), 1, 0, 1\}$

over standard KD; (3) **diminishing returns at higher capacity**, with the Large model showing a modest +3.10 pp gain, consistent with Proposition 21 as $\mu_s \to 1$ and the dynamic wisdom ratio approaches standard KD; and (4) **super-additive interaction**, where ablation (Table III) on Medium 1 shows curriculum-only (+5.77 pp) and dynamic weighting-only (+4.14 pp) contributions combine to yield +10.14 pp, exceeding their individual sum.

Dynamic vs. fixed wisdom ratio: Table III directly isolates the contribution of $\alpha_s(\mathbf{x})$. Comparing rows 3 and 5: replacing a fixed α=0.5 with the dynamic ratio *within the same curriculum* lifts accuracy from 91.20% to 95.57% (+4.37 pp), confirming that sample-wise confidence modulation provides a clear and independent gain beyond curriculum scheduling alone. Conversely, comparing rows 2 and 4 shows that injecting $\alpha_s(\mathbf{x})$ *without* curriculum still improves over standard KD by +4.14 pp, demonstrating that the dynamic ratio is beneficial irrespective of the training schedule.

Middle-ground Pareto students: Small 1 (0.77 MB Flash, 580 KB RAM) and Small 3 (0.43 MB Flash, ∼300 KB RAM) occupy the middle of the Pareto front. Their HIL latencies (390.3 ms and ∼250 ms respectively on the Cortex-M7) sit between the extremes of Small 2 (155.5 ms) and Medium 1 (2963 ms), offering a practical sweet-spot: both fit within the 1 MB SRAM ceiling of mainstream Cortex-M4/M7 targets while delivering 89.14% and 84.00% CL-KD accuracy, making them attractive candidates for field-deployable periodic monitoring at ∼2.5–4 Hz. The monotonic relationship between Flash size and on-device latency (driven by MAC count) enables practitioners to navigate this trade-off directly from the Pareto front without additional hardware profiling.

MC-dropout computational overhead: The confidence scoring phase (Phase 1 of Algorithm 1) requires M=50 stochastic forward passes through the teacher for each training sample—a one-time cost of ∼12 minutes on a single RTX A4000 GPU for the $|\mathcal{D}_{\text{tr}}|$=2,800 sample training set. This is amortised across all Pareto students and adds <3% to the total Stage II wall-clock time (which is dominated by multi-epoch student training), making the overhead negligible in practice compared to standard KD.

TABLE II: Test accuracy (%). Bold = best per row.

Student	Flash (MB)	Base (%)	Std KD [4] (%)	Proposed CL-KD (%)	Δ_K (pp)
Small 1	0.77	69.00	78.00	**89.14**	+11.14
Small 2	0.09	76.14	70.29	**86.14**	+15.85
Small 3	0.43	78.86	70.29	**84.00**	+13.71
Medium 1	2.21	83.29	85.43	**95.57**	+10.14
Large_1	12.27	88.29	91.57	**94.67**	+3.10
Average		79.12	79.12	**89.90**	+10.79

TABLE III: Ablation study on Medium 1).

Configuration	curriculum	Acc (%)
Baseline (no KD)	×	83.29
+ Standard KD [4] (α=0.5)	×	85.43
+ Curriculum KD (fixed α)	✓	91.20
+ Dynamic α_s only	×	89.57
Proposed CL-KD (full)	✓	**95.57**

Hardware-in-the-Loop Validation: To validate deployability beyond software profiling, we perform hardware-in-the-loop (HIL) benchmarking using the *STMicroelectronics Edge AI Developer Cloud*[2] with the *STM32H7S78-DK* development board (Arm Cortex-M7, 600 MHz, up to 2 MB Flash, 1 MB SRAM). Each CL-KD-trained student is converted to an INT8-quantised TensorFlow Lite model and profiled on-device. Table IV reports the measured metrics for three representative Pareto students. As observed from the results provided by HIL, it is evident that sub-second inference is achieved for ultra-small models with faithful INT8 deployment using CL-KD, while maintaining the Pareto properties derived by MOBO.

VI. Discussion

Teacher–student capacity gap: A natural question is whether a larger teacher always benefits a sub-100 KB student. Our results suggest otherwise: the dynamic wisdom ratio $\alpha_s(\mathbf{x})$ implicitly mitigates the capacity-gap problem [15] by down-weighting the teacher signal on samples where the teacher's predictive variance is high—precisely the regime where a large-capacity teacher's dark knowledge is least reliable for a tiny student. For Small 2 (d_S=11K), standard KD with the full teacher *degrades* accuracy by -5.85 pp (Table II), whereas CL-KD recovers $+15.85$ pp by shifting to ground-truth supervision on uncertain samples. This indicates that *selective trust*—not teacher size—is the critical factor for ultra-small students. Systematically studying the interaction between teacher depth, student capacity, and $\alpha_s(\mathbf{x})$ remains an open direction.

Future work: While CL-KD is formulated as a domain-agnostic framework, the present evaluation is limited to one fine-grained classification task. Validating CL-KD on standard benchmarks such as CIFAR-10 or ImageNet-10 under comparable TinyML constraints would strengthen generalisability claims and is a natural next step. Additional directions include: (i) multi-teacher ensembles to reduce confidence estimation variance; (ii) continuous or self-paced curriculum schedulers as alternatives to the fixed three-tier partition; and (iii) extension to feature-level distillation and other edge-vision domains (e.g., plant disease, defect inspection).

VII. Conclusion

We presented *CL-KD*, a two-stage framework for TinyML vision that jointly optimizes student design and training. Stage I employs constrained bi-objective MOBO to obtain Pareto-optimal architectures across orders of magnitude in model size, while Stage II introduces confidence-aware curriculum distillation with a dynamic wisdom ratio $\alpha_s(\mathbf{x})$ that unifies KD and supervised learning. On a seven-class hydrophobicity benchmark, CL-KD achieves +9.65,pp average improvement over baselines and +8.28,pp over standard KD, with gains up to +20.14,pp on sub-100,KB models where KD degrades. These results highlight the critical role of uncertainty-aware curriculum and adaptive distillation for effective TinyML deployment.

[2]https://stedgeai-dc.st.com

TABLE IV: HLL results on STM32H7S78-DK (INT8 TFLite)

Student	MACs (M)	Latency (ms)	Flash (KB)	RAM (KB)	CL-KD Acc (%)
Small 2	20.6	155.5	99.45	270.85	86.14
Small 1	66.4	390.3	613.34	580.11	89.14
Medium 1	858.2	2963.0	980.07	2273.3	95.57

References

[1] P. Warden and D. Situnayake, *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. O'Reilly Media, 2019.

[2] S. Somvanshi, M. M. Islam, G. Chhetri, R. Chakraborty, M. S. Mimi, S. A. Shuvo, K. S. Islam, S. Javed, S. A. Rafat, A. Dutta *et al.*, "From tiny machine learning to tiny deep learning: A survey," *ACM Computing Surveys*, vol. 58, no. 7, pp. 1–33, 2025.

[3] F. Karl, T. Pielok, J. Moosbauer, F. Pfisterer, S. Coors, M. Binder, L. Schneider, J. Thomas, J. Richter, M. Lang *et al.*, "Multi-objective hyperparameter optimization in machine learning—an overview," *ACM Transactions on Evolutionary Learning and Optimization*, vol. 3, no. 4, pp. 1–50, 2023.

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

[5] C.-C. A. Kokalis, T. Tasakos, V. T. Kontargyri, G. Siolas, and I. F. Gonos, "Hydrophobicity classification of composite insulators based on convolutional neural networks," *Engineering Applications of Artificial Intelligence*, vol. 91, p. 103613, 2020.

[6] International Electrotechnical Commission, "Guidance on the measurement of wettability of insulator surfaces," Standard IEC TS 62073, 2016.

[7] S. S. Roy, A. Paramane, B. Ganguly, A. K. Das, S. Paul, and S. Chatterjee, "Condition monitoring of polymeric insulators: A remote diagnostic framework with improved intermediate hydrophobicity grade detection," *IEEE Trans. Industry Applications*, vol. 61, no. 3, pp. 5251–5260, 2025.

[8] R. Akash, R. Sarathi, and M. Haddad, "Hybrid Swin transformer approach for hydrophobicity monitoring of outdoor polymeric insulators in transmission lines," *IEEE Trans. Dielectrics and Electrical Insulation*, vol. 32, no. 5, pp. 2922–2929, 2025.

[9] S. Chatterjee, S. S. Roy, A. Chatterjee, B. Ganguly, and S. Paul, "Recognition of hydrophobicity class of polymeric insulators employing residual morphological neural network and granulometry-based image analysis," *IEEE Trans. Instrumentation and Measurement*, vol. 71, p. 5008609, 2022.

[10] X. Ye, R. Jiang, X. Tian, R. Zhang, and Y. Chen, "Knowledge distillation via multi-teacher feature ensemble," *IEEE Signal Processing Letters*, vol. 31, pp. 566–570, 2024.

[11] Z. Tu, X. Liu, and X. Xiao, "A general dynamic knowledge distillation method for visual analytics," *IEEE Trans. Image Processing*, vol. 31, pp. 6517–6531, 2022.

[12] M. Loni, S. Sinaei, A. Zoljodi, M. Daneshtalab, and M. Sjodin, "DeepMaker: A multi-objective optimization framework for deep neural networks in embedded systems," *Microprocessors and Microsystems*, vol. 73, p. 102989, 2020.

[13] S. Garai and S. Samui, "Advances in small-footprint keyword spotting for tinyml: A comprehensive review of efficient models and algorithms," *Neurocomputing*, p. 134028, 2026.

[14] S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy, "Multi-objective Bayesian optimization over high-dimensional search spaces," in *Proc. UAI*, 2022, pp. 507–517.

[15] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2019, pp. 4794–4802.