

ORIGINAL ARTICLE

Special Issue

Cross-corruption distillation for robust instruction following under text corruptions

Yongjin Bae  | Kyoungman Bae 

Visual Intelligence Research Section,
Electronics and Telecommunications
Research Institute, Daejeon, Republic of
Korea

Correspondence

Yongjin Bae, Visual Intelligence Research
Section, Electronics and
Telecommunications Research Institute,
Daejeon, Republic of Korea.
Email: yongjin@etri.re.kr

Funding information

This work was supported by Institute for
Information & communications
Technology Planning & Evaluation (IITP)
grant funded by the Korea government
(MSIT) (no. RS-2022-II220369, (Part 4)
Development of AI Technology to support
Expert Decision-making that can Explain
the Reasons/Grounds for Judgment
Results based on Expert Knowledge).

Abstract

Traditional knowledge distillation transfers information from a large teacher model to a compact student by aligning static output distributions or internal representations for fixed inputs. Such formulations overlook how predictions change in response to input perturbations, even though robustness is critical for deployment. We interpret distillation as the transfer of functional response behavior under input variations and propose cross-corruption distillation (CCD), which aligns corruption-induced differences in teacher and student output via a Kullback–Leibler objective. Rather than matching absolute logits, CCD distills how the teacher redistributes its prediction mass under perturbations, transferring teacher-conditioned input sensitivity without architectural correspondence. On diverse multimodal benchmarks, CCD performs competitively, often outperforming conventional logit-based distillation. It also slows performance degradation as perturbation strength increases and remains stable across corruption strategies and training schedules. These results position CCD as a response-aware distillation framework integrating robustness directly into the distillation objective.

KEYWORDS

behavior-level distillation, knowledge distillation, robustness, trustworthy AI
vision-language models

1 | INTRODUCTION

With the rapid advancement of large-scale neural network models, knowledge distillation (KD) has become a fundamental strategy for transferring knowledge from powerful teacher models to compact student models. In large language and multimodal model settings, KD is crucial in maintaining strong performance under limited computational resources. Most existing distillation approaches, however, formulate knowledge as a

static mapping from input to output, aligning either output distributions or internal representations for a fixed input x . In such formulations, the teacher's prediction $T(x)$ serves as the primary supervision signal, and training is designed to enforce $S(x) \approx T(x)$.

While static alignment effectively transfers predictive distributions and representation structures, it does not explicitly model how outputs should change when the input is perturbed. In many real-world scenarios, reliability depends not only on clean accuracy but also on

structured and consistent responses under incomplete or locally corrupted inputs. For example, when certain tokens in a text input are missing or altered, a well-behaved model should preserve core semantic structure while adjusting predictions in a controlled manner. From this perspective, an important component of the teacher's knowledge lies not only in what it predicts but also in how its predictions change in response to input variations.

A number of prior studies have sought to improve robustness under input perturbations through techniques such as Jacobian-based regularization, noise injection, adversarial training, and consistency regularization. These approaches typically encourage invariance or smoothness by directly limiting output variation. However, they impose generic regularization constraints on the student model and do not explicitly transfer the structured input–output response patterns learned by the teacher. In other words, they aim to reduce output variation, whereas the teacher may exhibit meaningful and conditional variation depending on the type of perturbation. As a result, existing robustness-oriented methods do not treat conditional response behavior itself as a distillation target.

Motivated by this limitation, we reinterpret KD as the transfer of functional response behavior of the input–output mapping. Rather than aligning only absolute outputs for fixed inputs, we distill how the teacher model redistributes its predictions under stochastic input corruptions. Specifically, we posit that the teacher's knowledge includes a conditional response function that governs how logits change when the input is partially corrupted, and this response structure should be explicitly transferred to the student. Hence, we introduce cross-corruption distillation (CCD). CCD considers pairs of original and corrupted inputs and aligns the corruption-induced output variations of the teacher and student models. Instead of matching absolute logits, CCD aligns logit-level differences induced by identical perturbations via a Kullback–Leibler (KL) divergence objective. This formulation shifts the distillation target from static predictions to corruption-induced response distributions, enabling the transfer of teacher-conditioned input sensitivity without requiring architectural alignment or layer-wise correspondence. We evaluate CCD on diverse multimodal benchmarks spanning English and Korean. Experimental results demonstrate that CCD maintains competitive clean performance and often improves upon conventional logit-based distillation. More importantly, CCD consistently slows performance degradation as input perturbation strength increases and exhibits stable behavior across different corruption strategies and training corruption schedules. These findings indicate that

CCD functions not merely as data augmentation or generic regularization but also as a response-aware distillation method that explicitly transfers functional response characteristics from teacher to student. The main contributions of this work are as follows:

- We reinterpret KD as the transfer of functional response behavior under input perturbations, redefining the distillation target from static outputs to corruption-induced response distributions.
- We propose CCD, which aligns teacher and student logit variations induced by identical corruptions through a KL-based objective.
- Through extensive multimodal experiments, we demonstrate that CCD preserves clean performance while significantly improving robustness to input perturbations and maintaining stability across corruption strategies and training schedules.

By extending KD beyond static output alignment, we bridge distillation and robustness-aware learning, providing a principled response-aware formulation for training more reliable multimodal models.

2 | RELATED WORKS

KD has become a standard framework for transferring knowledge learned by large models to compact student models. Prior studies have expanded the definition of knowledge to include teacher model outputs, internal representations, or structural relationships and have trained student models to mimic these signals. In this section, we organize existing work according to the level at which knowledge is defined and discuss how model response behavior to input variations has been treated in existing frameworks.

2.1 | Output-level KD

Output-level KD regards the teacher model's output distribution itself as the primary form of knowledge, with logit-based distillation being the most representative approach. Early studies [1] demonstrated that the soft targets produced by a teacher model contain interclass similarity information, often referred to as dark knowledge, and that transferring this information can improve student performance. Subsequent work [2, 3] interpreted logit-based KD using output-space regularization and analyzed its effects on generalization. However, these approaches focus on aligning output distributions for the same input and do not directly address changes in

outputs to input perturbations. Hence, while output values are distilled, dynamic properties such as input sensitivity or output function response structure are typically only implicitly reflected.

2.2 | Representation-level KD

To transfer richer forms of knowledge beyond output-level information, many studies have extended distillation targets to the internal representations of teacher models. Feature- and layer-based distillation methods align intermediate representations [4] or define interlayer feature flows and decision boundary structures as knowledge [5, 6]. Other approaches leverage feature similarity or relational structures [7–10] to alleviate dimensional mismatches between teacher and student models and to convey structural properties of the representation space. With the widespread adoption of Transformer-based architectures, distillation methods that simultaneously utilize multiple internal signals—such as output logits, hidden states, and self-attention patterns—have also been proposed [11–14]. While these methods have proven effective for maintaining performance in modern Transformer models, they focus on static alignment of representations or structures for fixed inputs. Consequently, conditional response behaviors describing how representations or outputs change under input variations are generally not treated as explicit distillation targets.

2.3 | Robustness and input sensitivity

Beyond preserving performance through KD, prior work has explored improving the robustness of student models under input perturbations. Existing approaches can be broadly divided into magnitude-based regularization and consistency-based training.

Magnitude-based methods, such as Jacobian regularization [15], constrain output gradients with respect to input variations, encouraging local smoothness and limiting the magnitude of output changes. Similarly, adversarial and virtual adversarial training methods [16] promote stability by reducing output variation under worst-case perturbations. While effective for enhancing local robustness, these approaches primarily control the magnitude of output changes rather than modeling structured response behavior.

Consistency-based approaches enforce agreement between model outputs under clean and perturbed inputs. In teacher-guided variants, student predictions on corrupted inputs are aligned with teacher predictions. However, these methods generally match absolute output

distributions under perturbed inputs rather than explicitly modeling the relative response differences between clean and corrupted inputs. Hence, they emphasize invariance or stability rather than transferring how output distributions conditionally evolve under specific perturbations.

In contrast, instead of penalizing output magnitudes or enforcing unconditional consistency, CCD aligns the distribution of logit differences induced by identical corruptions in the teacher and student models. By matching corruption-induced response distributions defined over logit-level variations, CCD transfers the teacher's conditional response structure. Unlike generic robustness regularization methods that are teacher-independent, CCD relies on a teacher-defined response distribution, distilling structured input sensitivity rather than imposing uniform smoothness.

2.4 | Motivation for CCD

Existing KD methods have transferred teacher knowledge at various levels, including output values, internal representations, and relational structures, but they are often designed around static alignment for fixed inputs. Robustness-oriented approaches regulate the magnitude of output changes under input perturbations but do not explicitly model or transfer the teacher's response behavior. Motivated by these limitations, we propose CCD, which defines knowledge not as specific output values or representations but as the patterns of output variation induced by input corruptions. By training the student model to match the teacher's responses to observable input perturbations, CCD provides a more direct means of distilling the teacher's input sensitivity structure and functional behavior. Unlike prior approaches that either distill static outputs or enforce magnitude-based smoothness, CCD explicitly shifts the distillation target to corruption-induced response distributions conditioned on the teacher model.

3 | METHOD

3.1 | Preliminaries

Before introducing CCD, we summarize the basic learning setup and conventional KD losses used in this work. We assume a dataset $\mathcal{D} = \{(x, y)\}$ consisting of input–output pairs. The teacher model $T(\cdot)$ and student model $S(\cdot)$ perform the same task but differ in model capacity and representational power. The former is assumed to have sufficient parameters and expressive capability, while the latter is a comparatively lightweight model.

Given an input sequence x of length L , model output is formulated using logits $z \in \mathbb{R}^{L \times V}$, where V denotes vocabulary size. The student model is primarily trained using standard cross-entropy loss with respect to ground-truth labels, defined as follows:

$$\mathcal{L}_{\text{CE}} = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\text{CE}(S(x), y)]. \quad (1)$$

This loss encourages the student's predicted distribution to match the ground-truth labels and serves as the fundamental training objective. KD further incorporates the teacher model's output distribution as an additional supervision signal. The softened output distributions of the teacher and student are aligned using a KL-divergence loss:

$$\mathcal{L}_{\text{KD}} = \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\text{KL} \left(\sigma \left(\frac{z_T}{\tau} \right) \parallel \sigma \left(\frac{z_S}{\tau} \right) \right) \right], \quad (2)$$

where $z_T = T(x)$, $z_S = S(x)$, and $\sigma(\cdot)$ denotes the softmax function. This loss is widely adopted as a distillation approach for transferring interclass similarity information from teacher to student.

In this work, we regard such logit-based KD as a representative static distillation method, in which the output value for a given input is treated as the primary unit of knowledge. The following section introduces CCD, which, in contrast, defines knowledge at a different level by directly modeling output responses to input variations.

3.2 | CCD

Most existing KD methods treat the teacher model's output values or internal representations for a given input as the unit of knowledge. That is, they aim to satisfy $S(x) \approx T(x)$ for a fixed input x , thereby transferring the teacher's predictive distribution or representation structure to the student. Although such static alignment can improve performance on clean inputs, it does not explicitly model the model's response behavior, which characterizes how outputs change when the input is partially corrupted.

In real-world scenarios, model reliability and generalization depend not only on prediction accuracy but also on how consistently the model responds under incomplete or locally corrupted inputs. We observe that this response behavior constitutes an important form of knowledge learned by the teacher model, but it is not treated as an explicit distillation target in conventional frameworks. Motivated by this observation, we redefine the unit of knowledge in distillation as the structure of

output variation induced by input corruption rather than the output values themselves. That is, the teacher's knowledge encompasses not only what it predicts but also how its predictions change when the input is modified.

CCD is based on this perspective and directly distills the teacher's output responses to input corruptions. To observe response behavior, CCD considers pairs of original and corrupted inputs. Given an input x , we define a stochastic corruption function $C(\cdot)$ and generate a corrupted input $\tilde{x}$ as $\tilde{x} = C(x)$. For a text sequence $x = (x_1, x_2, \dots, x_L)$, each token is independently corrupted with probability p , either by replacing it with a designated placeholder token (e.g., UNK when available) or by substituting it with a randomly sampled token from the vocabulary. Special tokens (BOS, EOS, and PAD) are excluded from corruption to preserve the input structure. Importantly, corruption in CCD is not used as data augmentation but as an instrument for probing and transferring the teacher's input sensitivity structure. Figure 1 illustrates the overall procedure of CCD.

The original and corrupted inputs are passed through both the teacher and student models to obtain the following logits:

$$z_T = T(x), \tilde{z}_T = T(\tilde{x}), \quad (3)$$

$$z_S = S(x), \tilde{z}_S = S(\tilde{x}). \quad (4)$$

Instead of aligning absolute output values, CCD focuses on the output variations induced by the same input corruption. We define the output variations for the teacher and student as follows:

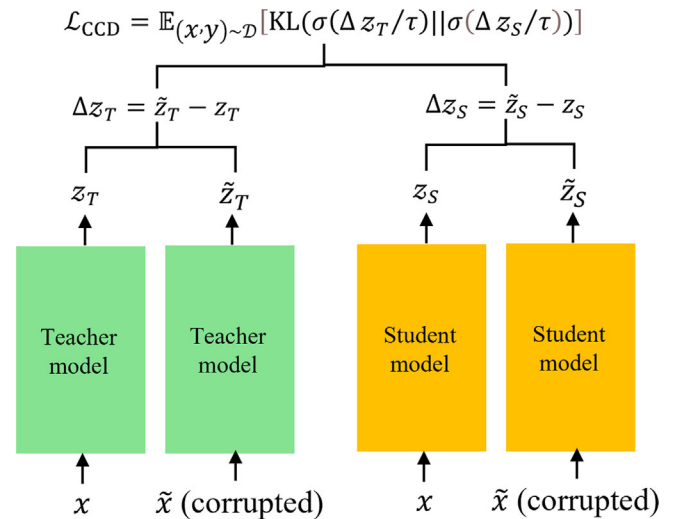


FIGURE 1 Overview of CCD.

$$\Delta z_T = \tilde{z}_T - z_T, \Delta z_S = \tilde{z}_S - z_S. \quad (5)$$

Here, $z \in \mathbb{R}^{L \times V}$ denotes the full sequence-level logit tensor produced by either the teacher or the student model, where L is the sequence length and V is the vocabulary size. The subtraction in (5) is performed element-wise over this tensor, that is, position-wise across tokens and dimension-wise across vocabulary logits. Since our corruption preserves sequence length and excludes special tokens (e.g., BOS/EOS/PAD), z and $\tilde{z}$ are aligned position-wise.

We interpret the logit differences as response signals and align their relative response patterns over the vocabulary using temperature-scaled softmax and KL divergence. This allows us to distill how the teacher redistributes its prediction mass under input corruption. To encourage the student model to exhibit similar response behavior, we define the CCD loss as follows:

$$\mathcal{L}_{\text{CCD}} = \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\text{KL} \left(\sigma(\Delta z_T / \tau) \parallel \sigma(\Delta z_S / \tau) \right) \right]. \quad (6)$$

The KL objective aggregates discrepancies across token positions and vocabulary dimensions via the vocabulary distribution at each position. This loss aligns the functional response behavior of the teacher and student models, distinguishing CCD from conventional distillation methods that mimic absolute output values or internal representations. Notably, CCD does not require assumptions about hidden dimensionality or layer-wise correspondence, and it can be applied without additional projection modules. Following standard instruction-tuning practice, all loss terms (CE, KD, and CCD) are computed only on target tokens, while nontarget tokens are masked out using an ignore index. The final training objective is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \alpha \mathcal{L}_{\text{KD}} + \gamma \mathcal{L}_{\text{CCD}}, \quad (7)$$

allowing supervised learning, output distribution alignment, and response behavior distillation to be jointly optimized. The overall training procedure of CCD is summarized in Algorithm 1.

3.3 | CCD discussion and interpretation

CCD is motivated by a limitation of existing distillation methods that focus on static alignment of outputs or representations for fixed inputs. This section discusses how CCD differs from prior work and provides an interpretation of its conceptual implications.

ALGORITHM 1 Training procedure with CCD1

Input

Training dataset $\mathcal{D} = \{(x, y)\}$
 Student model $S(\cdot)$, Teacher model $T(\cdot)$ (frozen)
 Corruption function $C(\cdot)$ with probability p , Temperature τ (used for KD)
 loss weight α, γ

Output

Trained student model $S(\cdot)$
 1 Initialize student model $S(\cdot)$
 2 Freeze parameters of teacher model $T(\cdot)$
 3 **for** each training step **do**
 4 Sample a mini-batch $(x, y) \sim \mathcal{D}$
 5 Generate corrupted inputs $\tilde{x} = C(x)$
 6 Student logits: $z_S = S(x)$, $\tilde{z}_S = S(\tilde{x})$
 7 Teacher logits: $z_T = T(x)$, $\tilde{z}_T = T(\tilde{x})$
 8 Output variations $\Delta z_T = \tilde{z}_T - z_T$,
 $\Delta z_S = \tilde{z}_S - z_S$
 9 Compute losses:
 10 Supervised loss $\mathcal{L}_{\text{CE}} = \text{CE}(z_S, y)$
 11 **if** KD is enabled **then**
 $\mathcal{L}_{\text{KD}} = \text{KL}(\text{softmax}(z_T / \tau) \parallel \text{softmax}(z_S / \tau))$
 12 **if** CCD is enabled **then**
 $\mathcal{L}_{\text{CCD}} = \text{KL}(\sigma(\Delta z_T / \tau) \parallel \sigma(\Delta z_S / \tau))$
 13 Total loss $\mathcal{L} = \mathcal{L}_{\text{CE}} + \alpha \mathcal{L}_{\text{KD}} + \gamma \mathcal{L}_{\text{CCD}}$
 14 Update student model parameters
 15 **end for**

3.3.1 | CCD versus conventional KD

Conventional KD methods improve student performance by aligning output distributions or internal representations between teacher and student models for the same input. While effective for transferring predictive structure, these approaches do not explicitly encode how outputs should change when the input is partially corrupted. In contrast, CCD aligns the output variation induced by identical input corruptions in both models. It thus transfers behavior-level knowledge from teacher to student without requiring explicit layer-wise correspondence or architectural alignment.

3.3.2 | CCD versus data augmentation

Although CCD utilizes input corruption, it differs fundamentally from data augmentation. Data augmentation typically enforces invariance by assigning identical labels

to augmented inputs, whereas corrupted inputs in CCD serve as reference points for measuring and aligning output responses. CCD does not force the student's outputs to remain unchanged under corruption. Instead, it encourages the student to mimic the teacher's conditional response, depending on whether the teacher preserves or alters its prediction for a perturbation. Thus, CCD aligns local input–output response characteristics rather than enforcing unconditional invariance.

3.3.3 | Interpretation from a robustness perspective

By aligning the student's output variations with those of the teacher under input corruptions, CCD encourages stable functional behavior. However, CCD is not a conventional robustness regularization method.

Traditional robustness-oriented regularization techniques typically aim to directly minimize the magnitude of output variation, thereby encouraging smoother or more invariant responses. In contrast, CCD does not suppress output changes. Instead, it aligns the teacher and student response distributions induced by input corruption via KL divergence, transferring how the teacher redistributes its prediction mass under perturbation. More specifically, CCD matches teacher and student corruption-induced response distributions over the vocabulary rather than enforcing small absolute output changes. If the teacher exhibits substantial distributional changes under a specific corruption, CCD encourages the student to respond similarly. Thus, CCD transfers conditional response patterns rather than enforcing unconditional smoothness.

Another key distinction is that CCD is explicitly teacher-dependent. The target response behavior is defined by the teacher model, whereas standard regularization methods operate independently of a teacher and impose generic smoothness constraints. CCD therefore does not merely constrain the student model but instead transfers the teacher's learned input sensitivity structure. Accordingly, CCD is better characterized not as an adversarial defense or a robustness regularization technique but as a behavior-level KD method that explicitly incorporates response alignment as a learning objective.

4 | EXPERIMENTAL SETUP

This section describes the datasets, model configurations, and training procedures used to evaluate the effectiveness of CCD. All experiments were conducted under identical training settings to ensure a fair comparison of different distillation components.

4.1 | Datasets

To train the student model under a multimodal instruction-following setting, we used large-scale image–text instruction datasets covering both English and Korean. All experiments were based on the same training data configuration. Specifically, we employed Bunny-695K [17] and LLaVA-665K [18] as the primary training datasets. These datasets include diverse forms of multimodal instructions, such as image-based question answering, visual description generation, object and attribute recognition, and simple visual reasoning. In addition, a subset of existing English multimodal instruction data was automatically translated into Korean using a large language model [19]. We further supplemented the training data with additional Korean multimodal instruction samples covering various visual question answering formats. The final training corpus consists of 3 840 977 image–instruction pairs, comprising 2 057 638 unique images. Among them, 1 359 908 pairs originate from the original Bunny and LLaVA datasets. Both English and Korean multimodal instructions were included, and the same data composition and proportions were maintained across all experiments to enable fair comparisons between different distillation strategies. For evaluation, we used multimodal benchmarks spanning both English and Korean. The English benchmarks include MMBench (DEV_V11), MMStar, and SEEDBench2+, while the Korean benchmarks include K-MMBench, K-MMStar, and K-SEED. All models were evaluated under identical benchmark settings.

4.2 | Models

The student model is based on Qwen3-VL-2B [20] and has been further reduced in size by applying a predefined structured layer pruning strategy to the language model's transformer layers. This pruning selectively retains only a subset of layers from the original language model, reducing the total number of parameters from approximately 2.1B to 1.6B. The same pruning configuration was fixed and used consistently across all experiments.

The objective of this study was not to evaluate the pruning strategy itself but rather to compare the learning effects of different distillation strategies (KD and CCD) under an identical lightweight student architecture. Accordingly, pruning was treated as part of the experimental setup that defines the student model rather than as a variable of interest. To further isolate the effects of distillation, the vision encoder and the multimodal projector were frozen during training and only the language model parameters were updated. This setting reduced variability

arising from visual representations and allowed us to focus on how output response behavior under input corruption was transferred through the language model.

4.3 | Training procedure

For the pruned student model, training was conducted in two sequential stages: a continual learning stage for representation stabilization and a subsequent KD stage. Both stages used the same multimodal instruction dataset and were trained for a single epoch, with only the loss function configuration varying across experiments. The objective of this study was not to maximize absolute final performance but to compare the relative effects of different distillation strategies under controlled and identical training conditions.

4.3.1 | Continual learning

After pruning, the reduced depth of the language model may lead to instability in the initial representation space. To mitigate this issue, we first implemented a stabilization stage before applying KD. In this stage, the student model was trained for one epoch using only the cross-entropy loss with respect to the ground-truth labels, without incorporating the teacher model. The purpose of this stage is to stabilize the pruned student model and prevent it from remaining in an excessively unstable initialization state. This provides a common and stable starting point from which the effects of different distillation losses can be more clearly observed. The resulting model serves as the shared initialization for all subsequent distillation experiments.

4.3.2 | Distillation settings

KD was then performed using the student model obtained after the continual learning stage as initialization. During the distillation stage, the model architecture,

training data, and number of training epochs were kept identical and only the loss function configuration was varied to compare different distillation strategies. We define the baseline setting that uses only the cross-entropy loss with ground-truth labels as “Supervised.” Adding the logit-based KD loss that aligns the teacher’s output distribution is referred to as “KD,” while adding the proposed CCD loss is called “CCD.” A setting that applies both KD and CCD is referred to as “KD+CCD.”

In all distillation experiments, Qwen3-VL-4B-Instruct was the teacher model, and its parameters were frozen throughout training. Moreover, identical optimizer settings, learning rate, batch size, and training schedules were used for the Supervised, KD, CCD, and KD+CCD settings. Loss weights (α and γ) were fixed to the same values for all experiments. This ensures that observed performance differences arise from the composition of the distillation losses rather than from differences in training hyperparameters or loss weight tuning. Optimization and distillation-specific hyperparameters are summarized in Appendix A.

5 | RESULTS

5.1 | Clean performance and loss ablation

In this section, we analyze the effect of different distillation strategies, defined in Section 4.3.2, on model performance under a clean evaluation setting without any input perturbation. This analysis aims to quantitatively assess how CCD behaves in comparison to conventional logit-based KD when inputs are uncorrupted.

As shown in Table 1, CCD consistently improves performance over the supervised baseline across all Korean and English benchmarks. This result indicates that, although CCD is designed to distill response behavior under input perturbations, it does not degrade performance in clean evaluation settings. On the contrary, CCD often enhances general multimodal understanding and reasoning capability even when no perturbation is

TABLE 1 Clean performance on Korean and English multimodal benchmarks under different distillation settings.

Language	Korean			English		
Distillation method	K-MMBench	K-MMStar	K-SEED	MMBench	MMStar	SEEDBench2+
Supervised	0.5713	0.5828	0.6977	0.6881	0.5193	0.5195
KD [†]	0.6022	0.5893	0.7122	0.6749	0.5293	0.5283
CCD [‡]	0.6177	0.5991	0.7128	0.7198	0.5613	0.5437
KD+CCD	0.6271	0.6009	0.6937	0.6796	0.5400	0.5314

Note: Bold values indicate the best performance in each setting.

present. On benchmarks such as MMBench and MMStar, which require a tight integration of visual perception and language reasoning, CCD outperforms KD. This suggests that response patterns learned through CCD may capture aspects of decision-relevant knowledge that are less effectively transferred through static alignment of output distributions alone.

In contrast, KD provides consistent performance gains over the Supervised baseline, although its improvements are generally smaller than those of CCD across most benchmarks. While KD focuses on matching the teacher's output distribution for a fixed input, CCD implicitly regularizes the model's response function by aligning output variations induced by input changes. As a result, CCD encourages more consistent output behavior even under clean conditions, which translates into improved benchmark performance.

The combined setting KD+CCD maintains competitive performance overall and achieves the best results on K-MMBench and K-MMStar. However, it does not consistently outperform CCD alone on all English benchmarks. This behavior suggests that CCD is not primarily optimized for maximizing clean accuracy but rather acts as a behavioral regularization constraint. For capacity-limited student models, simultaneously applying KD and CCD may introduce excessive constraints in certain cases, leading to marginal differences in clean performance.

Overall, these results demonstrate that CCD can effectively replace or complement conventional logit-based distillation even in clean evaluation environments. Despite explicitly relying on input perturbations during training, CCD consistently improves clean performance, reinforcing the interpretation that CCD functions not merely as a robustness-oriented technique but as a distillation method that regularizes the model's overall output behavior. However, clean performance alone is insufficient to characterize how models respond to corrupted or incomplete inputs. In the following sections, we therefore

analyze model behavior under progressively increasing input perturbations to better elucidate the robustness properties induced by different distillation strategies.

5.2 | Robustness to input perturbations

In this section, we analyze the performance degradation behavior of different distillation methods as the level of input perturbation increases. During evaluation, each token in the input text is perturbed independently with probability p and model performance is measured as p increases to assess how robustness changes under progressively corrupted inputs.

In perturbation-based evaluations, direct comparison of absolute performance curves can be misleading due to differences in clean performance across models. For example, a model with higher clean accuracy may appear less robust under the same absolute performance drop, while a weaker model may appear artificially robust. To mitigate this issue, we define a normalized performance drop metric that directly compares the relative rate of degradation:

$$\text{Drop}(p) = 1 - \frac{\text{score}(p)}{\text{score}(0)}. \quad (8)$$

This metric aligns all methods at the same starting point ($p=0$) and measures the accumulated performance loss as input perturbation increases. Lower values indicate smaller degradation under the same perturbation level.

Figure 2 shows the normalized performance drop curves as a function of the input perturbation probability p . Across all methods, performance degradation becomes increasingly pronounced as the perturbation level increases, particularly beyond $p=0.1$, which corresponds to a regime where a substantial portion of the input semantic structure is corrupted. Despite this overall collapse trend, clear differences are observed in the rate at

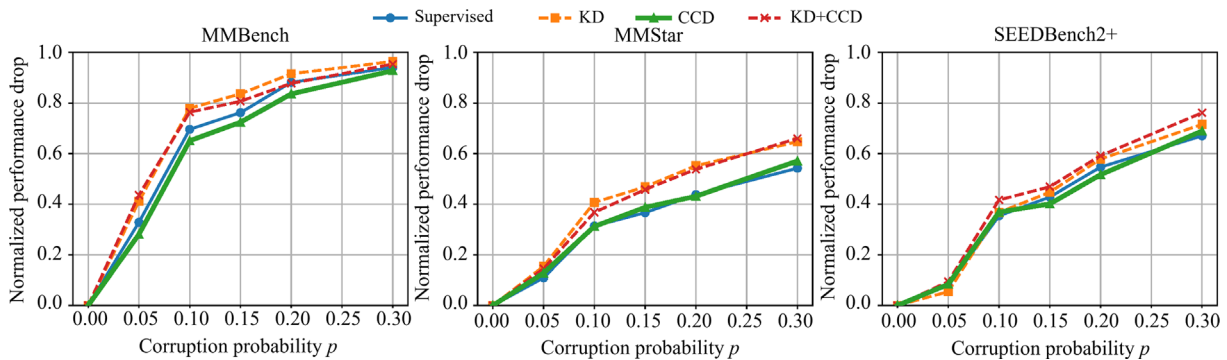


FIGURE 2 Normalized performance drop under increasing input perturbation probability p .

which performance degrades. Models trained with CCD generally exhibit more gradual degradation under moderate perturbation levels. In particular, under moderate perturbation regimes ($p \in [0.05, 0.20]$), CCD accumulates performance loss more slowly than both Supervised and KD. This indicates that CCD does not merely mimic the teacher's static output distribution but instead transfers response characteristics that enable the student model to maintain more stable behavior under the evaluated perturbation setting. In contrast, KD achieves competitive performance under clean inputs, but its performance deteriorates more rapidly as perturbation increases. This suggests that static alignment of output distributions alone is insufficient to preserve the model's sensitivity structure with respect to input changes. Applying KD and CCD jointly (KD+CCD) does not consistently outperform CCD alone. This can be attributed to the fact that the two distillation objectives impose different types of constraints on the student model. While KD enforces a global constraint by aligning output distributions, CCD introduces a local constraint by aligning output variations induced by input perturbations. For lightweight student models with limited capacity, simultaneously applying both constraints may lead to over-regularization, thereby limiting the cumulative benefit. We note that the relative advantage of CCD is most pronounced under moderate corruption levels ($p \in [0.05, 0.20]$). Under extreme corruption rates (e.g., $p = 0.30$), all methods exhibit severe performance degradation, suggesting that the remaining input information becomes insufficient for reliable reasoning regardless of the distillation strategy.

Overall, these results demonstrate that CCD effectively slows down performance degradation as input perturbation increases while preserving clean performance. By explicitly incorporating the model's dynamic response to perturbations as a learning signal, CCD improves the transfer of perturbation-induced response behavior and enables more stable performance under the evaluated perturbation setting. The corresponding absolute performance scores under varying perturbation probabilities are reported in Appendix B.

5.3 | Effect of corruption strategy

In this section, we analyze how different *corruption strategies* affect model performance while keeping the perturbation strength fixed. In Section 5.2, we evaluated robustness by comparing the overall degradation trends as the perturbation probability p increased. In contrast, the goal of this section is to examine whether models maintain stable output behavior when the form of input corruption changes under the same perturbation intensity.

For this analysis, we fix the perturbation probability at $p = 0.1$ during evaluation and consider two corruption strategies. The first is random replacement, where each selected token is replaced with a randomly sampled token from the vocabulary, thereby introducing semantic distortion. The second is PAD replacement, where selected tokens are replaced with a padding token to simulate information deletion rather than semantic substitution. In this experiment, PAD replacement does not correspond to actual sequence padding but instead serves as a PAD-as-UNK strategy to mimic token removal using a semantically empty token. Models were evaluated on the same data and under the same perturbation probability, only varying the corruption strategy, enabling a controlled comparison of strategy sensitivity. To quantify sensitivity to corruption strategy, we define the absolute performance difference between PAD-based and random-based evaluation for a model as follows:

$$\Delta_{\text{strategy}} = |\text{score}_{\text{PAD}} - \text{score}_{\text{Random}}|. \quad (9)$$

This metric measures how much performance changes when only the corruption strategy differs under the same perturbation strength. Smaller values indicate lower sensitivity to the specific implementation of input corruption.

Figure 3 presents the strategy sensitivity results on MMBench, MMStar, and SEEDBench2+. On MMBench and MMStar, the Supervised and KD settings exhibit relatively large differences between random replacement and padding replacement, indicating that their outputs are noticeably affected by how corruption is implemented. In contrast, models trained with CCD generally show smaller Δ_{strategy} values, demonstrating more stable behavior across the evaluated corruption strategies. In particular, CCD trained with padding-based corruption exhibits near-invariant performance on MMStar. The results on SEEDBench2+ reveal a more nuanced pattern. While padding-trained CCD remains highly stable, CCD trained with random corruption is more sensitive to strategy differences. This benchmark appears to place stronger emphasis on token-level semantic precision, and the distinction between semantic distortion and semantic deletion appears to have a larger impact. When the corruption distribution used during training does not align with the evaluation strategy, the response behavior learned through CCD may not transfer uniformly across corruption types.

Overall, these findings indicate that CCD does not enforce unconditional invariance to all forms of corruption. Rather, it aligns the student model's response behavior with the corruption distribution observed during training. When training and evaluation corruption

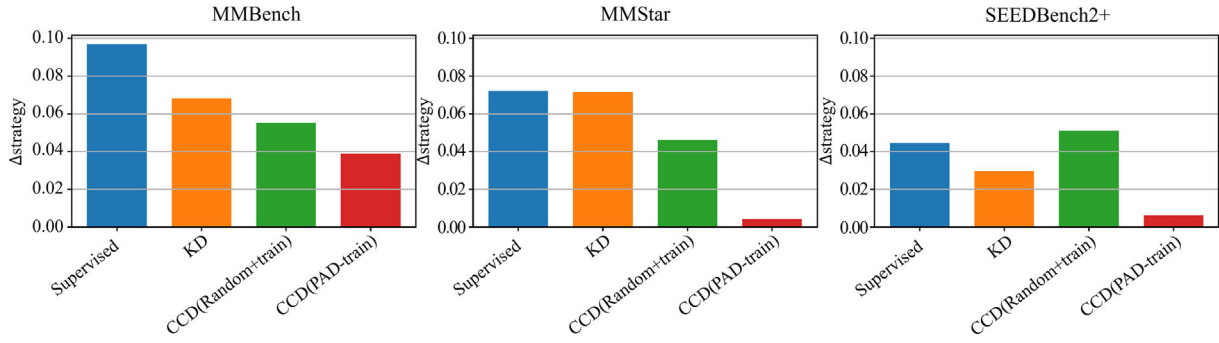


FIGURE 3 Strategy sensitivity under fixed corruption probability ($p = 0.1$).

regimes are consistent, CCD produces very small Δ_{strategy} values. This supports the interpretation that CCD transfers conditional response structures rather than merely regularizing output magnitudes. Accordingly, the robustness claims in this work should be interpreted within the evaluated token-level corruption settings (random replacement and PAD-based replacement) rather than as evidence of robustness to all possible real-world corruption types such as OCR noise or typographical errors.

5.4 | Sensitivity to training corruption probability

In the previous experiments (Sections 5.1–5.3), the corruption probability used during training was fixed at $p = 0.1$. This choice provides a practical balance between inducing observable output variations and avoiding excessive semantic degradation of the input. However, it also assumes a specific corruption strength during training. In this section, we examine how sensitive CCD is to the choice of training corruption probability. We compare two training schedules. In the first setting (fixed- p), the corruption probability is fixed at $p = 0.1$ throughout training. In the second setting (random- p), the corruption probability is sampled independently at each training step from a uniform distribution:

$$p_{\text{train}} \sim \text{Uniform}(0.05, 0.20). \quad (10)$$

Both settings use the student architecture, training data, optimizer configuration, number of epochs, and CCD loss weight. During evaluation, corruption probability p_{eval} is varied from 0.0 to 0.3.

Table 2 summarizes the results on MMBench, MMStar, and SEEDBench2+. For each benchmark, we report the average normalized performance drop within the training corruption range $0.05 \leq p_{\text{eval}} \leq 0.2$ (In-range Avg Drop), as well as the average normalized performance drop in the out-of-range region where

$p_{\text{eval}} > 0.2$ (Out-range Avg Drop). The normalized drop is computed using (8). Across all benchmarks, the differences between fixed- p and random- p training remain modest. In the in-range region, the two schedules exhibit nearly identical average degradation trends, suggesting that fixing $p = 0.1$ during training is generally sufficient for learning stable response behavior under moderate corruption levels. In the out-of-range region ($p_{\text{eval}} > 0.2$), the differences also remain limited, suggesting that randomizing corruption probabilities during training does not provide a consistent additional generalization benefit under stronger perturbations.

Overall, these results indicate that CCD is not highly sensitive to the specific choice of training corruption probability. Fixing corruption probability at $p = 0.1$ is a reasonable and stable training configuration, and sampling p from a moderate range does not substantially alter the induced robustness behavior, suggesting that CCD captures response structures with respect to input degradation rather than relying on a narrowly tuned corruption strength.

5.5 | Response behavior analysis

As representative robustness-oriented baselines, we additionally implemented augmentation CE and consistency regularization using the same corruption process employed in CCD training. Augmentation CE applies cross-entropy supervision to both clean and corrupted inputs, while Consistency Regularization encourages similar predictions for clean and corrupted inputs. Both methods aim to improve robustness under corruption through prediction stability.

While Sections 5.2 and 5.3 showed that corruption-aware training can improve robustness, improved robustness does not necessarily imply that a student reproduces the teacher's response behavior under corruption. To better understand what is transferred by different objectives, we analyze response behavior alignment

TABLE 2 Sensitivity to training corruption probability.

Benchmark	Training schedule	In-range avg↓	Out-range avg↓
MMBench	Fixed $p = 0.1$	0.6231	0.0919
	Random $p \sim U(0.05, 0.20)$	0.6367	0.0882
MMStar	Fixed $p = 0.1$	0.3149	0.1393
	Random $p \sim U(0.05, 0.20)$	0.3120	0.1291
SEEDBench2+	Fixed $p = 0.1$	0.3427	0.1730
	Random $p \sim U(0.05, 0.20)$	0.3355	0.1722

TABLE 3 Response behavior analysis across three multimodal benchmarks.

Bench- mark	Method	Clean accuracy	Corrupted accuracy ($p = 0.1$)	Accuracy drop	Teacher agreement (correct)	Teacher agreement (incorrect)	Flip alignment	Transition alignment
MMBench	CCD	0.7020	0.3668	0.3351	0.8420	0.5857	0.5925	0.2530
	KD+CCD	0.7167	0.3707	0.3459	0.8598	0.5818	0.5370	0.2160
	Augmentation CE	0.7174	0.5905	0.1269	0.8993	0.5562	0.3765	0.1296
	Consistency reg.	0.7051	0.4473	0.2577	0.8700	0.5779	0.4814	0.1975
MMStar	CCD	0.5260	0.4786	0.0473	0.6050	0.3675	0.4541	0.1256
	KD+CCD	0.5426	0.4780	0.0646	0.6207	0.3943	0.4589	0.1521
	Augmentation CE	0.5240	0.4886	0.0353	0.5942	0.3244	0.3188	0.0676
	Consistency reg.	0.5293	0.4866	0.0426	0.6050	0.3482	0.4202	0.1111
SEED Bench2+	CCD	0.5300	0.4356	0.0944	0.5736	0.3972	0.5652	0.1556
	KD+CCD	0.5371	0.4259	0.1111	0.5736	0.4322	0.5599	0.1896
	Augmentation CE	0.5309	0.4791	0.0518	0.6290	0.3724	0.4239	0.1144
	Consistency reg.	0.5305	0.4554	0.0750	0.5873	0.3950	0.5026	0.1413

Note: Corrupted accuracy and accuracy drop measure robustness, while teacher agreement, flip alignment, and transition alignment measure behavior transfer. Higher values indicate stronger transfer. Bold values indicate the best performance in each setting.

under corruption. For reliable transition analysis, we use constrained option-level decoding (A/B/C/D) to ensure consistent answer extraction from both clean and corrupted inputs. The accuracy values in this section are therefore intended for diagnostic analysis and should be interpreted primarily in terms of relative behavioral trends across methods.

Table 3 compares CCD-based methods with robustness-oriented baselines, revealing the distinction between corruption robustness and response behavior transfer. Augmentation CE and consistency regularization generally achieve higher corrupted accuracy and smaller performance drops than CCD-based methods, reflecting their focus on prediction stability under corrupted inputs.

To analyze response behavior transfer, we consider four complementary metrics. Teacher agreement (correct) and teacher agreement (incorrect) measure student–teacher agreement conditioned on whether the teacher prediction is correct or incorrect, respectively. Flip alignment evaluates whether the student changes its prediction when the teacher changes its prediction under corruption, while transition alignment further requires the student to follow the same prediction transition as the teacher.

An interesting trend emerges as the analysis moves from static teacher agreement toward corruption-induced response behavior. On MMBench, robustness-oriented baselines achieve competitive or even higher teacher agreement (correct) scores than CCD-based

methods, indicating that they can reproduce teacher predictions when the teacher is correct. However, the relative advantage of CCD becomes increasingly pronounced for teacher agreement (incorrect), flip alignment, and transition alignment. Similar trends are observed on MMStar and SEEDBench2+, where CCD-based methods consistently exhibit stronger alignment with teacher prediction transitions despite lower corruption robustness.

These results suggest that CCD does not primarily improve static teacher imitation. Instead, its advantage emerges when evaluating how the student responds to corruption relative to the teacher. The higher teacher agreement (incorrect) scores indicate that CCD transfers response behavior patterns regardless of whether the teacher prediction is correct. While this may also propagate teacher errors, it is consistent with the objective of CCD, which aims to transfer corruption-response patterns rather than selectively imitate only correct teacher predictions. Furthermore, the consistently stronger flip alignment and transition alignment scores demonstrate that CCD transfers not only the tendency to react to corruption but also the direction of the teacher's response change under perturbations.

Overall, these findings demonstrate that corruption robustness and response behavior transfer represent different model properties. Robustness-oriented methods are effective at preserving prediction accuracy under corruption, whereas CCD more effectively reproduces the teacher's corruption-induced response behavior. These results highlight that robustness and response behavior transfer represent distinct learning objectives. The results support the interpretation that CCD is not merely a robustness regularization technique but a behavior-level distillation method that transfers response structures under input perturbations.

6 | CONCLUSION

In this paper, we reconsidered KD beyond static output alignment for a single input and instead formulated it as the transfer of response behavior with respect to input variations. To this end, we proposed CCD, which directly aligns the output variations of teacher and student models under identical input corruptions.

Through extensive experiments on diverse multi-modal benchmarks, we showed that CCD maintains competitive clean performance and in some cases improves upon conventional logit-based KD while effectively mitigating the rate of performance degradation under moderate corruption levels. Furthermore, CCD

exhibits relatively stable output behavior not only under varying corruption strengths but also across the evaluated corruption strategies. It was also shown to be insensitive to the specific choice of corruption probability used during training.

These findings indicate that CCD is not merely a regularization or data augmentation technique but a behavior-level distillation method that explicitly transfers the functional response structure of a model under input degradation. By extending KD from static output alignment to response-aware learning, this work provides a response-aware distillation framework that explicitly transfers perturbation-induced response behavior and incorporates robustness and stability as learning objectives. Future work will include investigating the generalization of CCD to broader classes of input transformations, such as visual perturbations, structured linguistic modifications, and semantically preserved rephrasing.

CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest.

ORCID

Yongjin Bae  <https://orcid.org/0000-0002-0227-8933>

Kyoungman Bae  <https://orcid.org/0000-0001-9007-4027>

REFERENCES

1. G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, arXiv preprint (2015), DOI [10.48550/arXiv.1503.02531](https://doi.org/10.48550/arXiv.1503.02531).
2. J. H. Cho and B. Hariharan, *On the efficacy of knowledge distillation*, (Proc. International Conference Computer Vision, Seoul, Republic of Korea), 2019, pp. 4794–4802.
3. L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, *Revisiting knowledge distillation via label smoothing regularization*, (Proc. IEEE Conference Computer Vision Pattern Recognition), 2020, pp. 3903–3911.
4. A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, *FitNets: hints for thin deep nets*, arXiv preprint (2014), DOI [10.48550/arXiv.1412.6550](https://doi.org/10.48550/arXiv.1412.6550).
5. J. Yim, D. Joo, J. Bae, and J. Kim, *A gift from knowledge distillation: fast optimization, network minimization and transfer learning*, (Proc. IEEE Conference Computer Vision Pattern Recognition, Honolulu, HI, USA), 2017, pp. 4133–4141.
6. B. Heo, M., Lee, S. Yun, and J. Y. Choi, *Knowledge transfer via distillation of activation boundaries formed by hidden neurons*, (Proc. AAAI Conference Artificial Intelligence, Honolulu, HI, USA), 2019, pp. 3779–3787.
7. F. Tung and G. Mori, *Similarity-preserving knowledge distillation*, (Proc. International Conference Computer Vision, Seoul, Republic of Korea), 2019, pp. 1365–1374.
8. W. Park, D. Kim, Y. Lu, and M. Cho, *Relational knowledge distillation*, (Proc. Conference Computer Vision Pattern Recognition, Long Beach, CA, USA), 2019, pp. 3967–3976.

9. Y. Liu, C. Shun, J. Wang, and C. Shen, *Structured knowledge distillation for dense prediction*, IEEE Trans. Pattern Anal. Mach. Intell. **45** (2023), no. 6, 7035–7049.
10. B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, and Z. Zhang, *Correlation congruence for knowledge distillation*, (Proc. International Conference Computer Vision, Seoul, Republic of Korea), 2019, pp. 5007–5016.
11. V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*, arXiv preprint (2019), DOI [10.48550/arXiv.1910.01108](https://doi.org/10.48550/arXiv.1910.01108).
12. S. Sun, Y. Cheng, Z. Gan, and J. Liu, *Patient knowledge distillation for BERT model compression*, (Proc. Conference Empirical Methods Natural Language Processing, Hong Kong, China), 2019, pp. 4323–4332.
13. X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, *TinyBERT: distilling BERT for natural language understanding*, (Findings Association Computational Linguistics EMNLP), 2020, pp. 4323–4332.
14. W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, *MINILM: deep self-attention distillation for task-agnostic compression of pre-trained transformers*, (Proc. Conference Neural Information Processing Systems), 2020, pp. 5776–5788.
15. J. Hoffman, D. A. Roberts, and S. Yaida, *Robust learning with Jacobian regularization*, arXiv preprint (2019), DOI [10.48550/arXiv.1908.02729](https://doi.org/10.48550/arXiv.1908.02729).
16. T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, *Virtual adversarial training: a regularization method for supervised and semi-supervised learning*, IEEE Trans. Pattern Anal. Mach. Intell. **41** (2019), no. 8, 1979–1993.
17. M. He, Y. Liu, B. Wu, J. Yuan, Y. Wang, T. Huang, and B. Zhao, *Efficient multimodal learning from data-centric perspective*, arXiv preprint (2024), DOI [10.48550/arXiv.2402.11530](https://doi.org/10.48550/arXiv.2402.11530).
18. H. Liu, C. Li, Y. Li, and Y. J. Lee, *Improved baselines with visual instruction tuning*, (Proc. Conference Computer Vision and Pattern Recognition, Seattle, WA, USA), 2024, pp. 26296–26306.
19. A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, and C. Zheng, *Qwen3 technical report*, arXiv preprint (2025), DOI [10.48550/arXiv.2505.09388](https://doi.org/10.48550/arXiv.2505.09388).
20. S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, and W. Ge, *Qwen3-VL technical report*, arXiv preprint (2025), DOI [10.48550/arXiv.2511.21631](https://doi.org/10.48550/arXiv.2511.21631).

AUTHOR BIOGRAPHIES



Yongjin Bae received his BS degree in computer education from Mokwon University, Daejeon, Republic of Korea, in 2012 and his MS degree in computer software and engineering from the University of Science and Technology, Daejeon, Republic of Korea, in 2014. Since 2014, he has been with the Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea, where he is currently a Senior Researcher. His research interests include natural language processing, vision-language models, neural information retrieval, and retrieval-augmented generation.



Kyoungman Bae received the BS, MS, and PhD degrees in computer engineering from the Department of Computer Engineering, Dong-A University, Busan, Republic of Korea, in 2004, 2006, and 2016, respectively. Since 2016, he has been working for the Visual Intelligence Research Section at the Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea. His research interests include large language model, natural language processing, explainable artificial intelligence, and generative artificial intelligence.

How to cite this article: Y. Bae and K. Bae, *Cross-corruption distillation for robust instruction following under text corruptions*, ETRI Journal **48** (2026), 852–867, DOI [10.4218/etrij.2026-0159](https://doi.org/10.4218/etrij.2026-0159).

APPENDIX A: TRAINING HYPERPARAMETERS

To ensure reproducibility and fair comparison across different distillation strategies, all experiments were conducted under identical optimization settings unless otherwise specified.

TABLE A.1 Optimization settings.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	1e−5
Scheduler	Cosine
Finetuning type	Full
Frozen modules (vision tower /projector/ language model)	Frozen /frozen/ trainable
Context (cutoff length)	3584
Warmup ratio	0.1
Batch size (per device)	4
Gradient accumulation	9
#GPUs	8
Global batch size	288
Mixed precision	BF16
Training epochs (continual stage)	1.0
Training epochs (distillation stage)	1.0

TABLE A.2 Distillation-specific hyperparameters.

Parameter	Value
Temperature (τ)	2.0
KD loss weight (α)	0.5 (used in KD and KD+CCD setting)
CCD loss weight (γ)	0.05
Corruption probability (fixed- p)	[0.10]
Corruption probability (random- p)	[0.05–0.20]

APPENDIX B: RAW PERFORMANCE UNDER INPUT PERTURBATIONS

This section reports the absolute performance scores under varying perturbation probabilities. These results complement the normalized degradation curves presented in Section 5.2 and confirm that the observed trends are consistent in terms of raw performance. Consistent with the main analysis, the relative advantage of CCD is most evident under moderate perturbation levels, while the benefit diminishes under extreme corruption

rates where all methods experience substantial performance degradation.

TABLE B.1 MMBench.

Method	$p = 0.0$	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
Supervised	0.6881	0.4566	0.2066	0.0804	0.0402
KD	0.6749	0.3947	0.1478	0.0565	0.0239
CCD	0.7198	0.5085	0.2469	0.1160	0.0510
KD+CCD	0.6796	0.3885	0.1625	0.0843	0.0317

TABLE B.2 MMStar.

Method	$p = 0.0$	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
Supervised	0.5193	0.4500	0.3466	0.2840	0.2313
KD	0.5293	0.4606	0.3226	0.2440	0.1920
CCD	0.5613	0.4633	0.3653	0.3020	0.2280
KD+CCD	0.5400	0.4426	0.3273	0.2393	0.1760

TABLE B.3 SEEDBench2+.

Method	$p = 0.0$	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
Supervised	0.5195	0.4896	0.3434	0.2411	0.1752
KD	0.5283	0.4879	0.3263	0.2173	0.1466
CCD	0.5437	0.4861	0.3346	0.2564	0.1646
KD+CCD	0.5314	0.4756	0.3061	0.2143	0.1251

APPENDIX C: COMPUTATIONAL COST

CCD requires both clean and corrupted inputs to be processed by the teacher and student during training. Consequently, CCD incurs additional training overhead compared to supervised learning, conventional knowledge distillation, and robustness-oriented methods that rely solely on student-side forward passes. In particular, augmentation CE and consistency regularization require clean and corrupted student forward passes, whereas CCD additionally requires clean and corrupted teacher forward passes to transfer teacher-conditioned prediction transitions under corruption.

The additional computational cost is incurred only during training. During inference, all methods use the distilled 1.6B student model and therefore introduce neither additional model size nor inference overhead. Hence, CCD trades increased training-time computation for improved teacher-conditioned corruption-response behavior transfer while preserving deployment efficiency.

TABLE C1 Computational cost.

Method	Training objective	Student clean pass	Student corrupt pass	Teacher clean pass	Teacher corrupt pass	Relative training cost	Relative inference cost	Final model size
Supervised	Cross-entropy on clean input	✓	–	–	–	1.0×	1.0×	1.6B
KD	KD on clean input	✓	–	✓	–	$\sim 1.0 \times -1.5 \times$	1.0×	1.6B
Augmentation CE	Cross-Entropy on clean and corrupted inputs	✓	✓	–	–	$\sim 2.0 \times$	1.0×	1.6B
Consistency Regularization	Consistency loss between clean and corrupted student predictions	✓	✓	–	–	$\sim 2.0 \times$	1.0×	1.6B
CCD	Teacher-conditioned clean-to-corrupted response alignment	✓	✓	✓	✓	$\sim 2.5 \times -3.0 \times$	1.0×	1.6B
KD+CCD	KD+teacher-conditioned response alignment	✓	✓	✓	✓	$\sim 3.0 \times$	1.0×	1.6B

APPENDIX D: EVALUATION ON AN ADDITIONAL STUDENT ARCHITECTURE

To assess whether the effectiveness of CCD generalizes beyond the layer-pruned student model used in the main experiments, we additionally evaluate CCD on Qwen3-VL-2B-Instruct. The model was trained using the same dataset and optimization settings as the main experiments. Results are reported using the checkpoint obtained after 4000 training steps. This experiment serves as a supplementary validation of architectural generality.

Across all three benchmarks, CCD consistently outperforms both KD and KD+CCD, demonstrating that CCD remains competitive and achieves the strongest overall performance among the evaluated distillation methods, suggesting that its effectiveness is not specific to the layer-pruned student architecture.

TABLE D1 Evaluation on an additional student architecture.

Method	MMBench	MMStar	SEEDBench2+
Baseline	0.7686	0.5407	0.6755
KD	0.7964	0.5733	0.6768
CCD	0.8026	0.5860	0.6798
KD+CCD	0.7988	0.5793	0.6750

Note: Bold values indicate the best performance in each setting.

APPENDIX E: SEED SENSITIVITY ANALYSIS

To assess the sensitivity of the observed trends to random initialization, we trained KD, CCD, and KD+CCD using three random seeds (42, 123, and 3407). All models were trained using the revision experiments configuration and evaluated using checkpoints after 4000 training steps. This analysis examines stability across seeds rather than reproducing the results in Table 1.

Table E.1 shows that moderate performance variations are observed across random seeds. Nevertheless, the overall trends remain broadly consistent across seeds, suggesting the observed behavior is not attributable to a particular random initialization.

TABLE E.1 Mean and standard deviation across three random seeds.

Method	MMBench	MMStar	SEEDBench2+
KD	0.6860 ± 0.0060	0.5302 ± 0.0077	0.5390 ± 0.0030
CCD	0.6901 ± 0.0105	0.5360 ± 0.0072	0.5299 ± 0.0035
KD+CCD	0.7227 ± 0.0064	0.5584 ± 0.0148	0.5387 ± 0.0055

Table E.2 Raw benchmark scores for individual random seeds.

Method	Seed	MMBench	MMStar	SEEDBench2+
KD	42	0.6927	0.5340	0.5393
KD	123	0.6811	0.5353	0.5419
KD	3407	0.6842	0.5213	0.5358
CCD	42	0.7020	0.5280	0.5296
CCD	123	0.6865	0.5420	0.5336
CCD	3407	0.6819	0.5380	0.5266
KD+CCD	42	0.7175	0.5413	0.5367
KD+CCD	123	0.7299	0.5673	0.5450
KD+CCD	3407	0.7206	0.5667	0.5345