Distribution-Level Calibration Beats Sample-Level Reweighting: Confusion-Calibrated Supervision Distillation for Tool-Use Invocation

Anonymous ACL submission

Abstract

We introduce Confusion-Calibrated Supervision Distillation (CCSD), a weak-teacher distillation objective for the first decision a tool-using language model must make: whether to call a tool, ask for missing information, or abstain. In the function-calling SFT setups we study, valid-call-heavy training data degrades this invocation decision and produces over-callers. CCSD treats a moderately accurate teacher as a noisy annotator. Given the teacher's 3-class reliability matrix R, estimated by 5-fold out-of-fold on fine-tuning data, it uses the corrected target $q_{\text{cal}} = p_{\text{teacher}}^{\top} R$ as an auxiliary KL loss alongside full-sequence SFT. The main effect is both accuracy and stability. On a Qwen3-4B student with a weaker Qwen2.5-7B teacher (overall accuracy 0.56), CCSD-hard improves OOD-1 BFCL macro-F1 by 5.3 pp over a focal-loss auxiliary baseline and reduces the across-seed standard deviation from 0.070 to 0.006, about a $12\times$ drop. It also outperforms naïve logit distillation by 10.8 pp and our matched-budget DPO-style preference instantiation by 14.7 pp on the same split. On a held-out APIGen OOD split with independently constructed refusal cases, CCSD's macro-F1, tool-F1, and over-call point estimates favor CCSD across all four seeds (with one seed essentially tied at -0.005, within seed noise); the pooled bootstrap reports $p<0.05$, while a seed-level t-test at $n=4$ is inconclusive. A directional Llama cross-backbone check (2 seeds) preserves the mean ranking, with focal's high-variance pattern reappearing on OOD-2.

1 Introduction

Tool-using language models face a three-way invocation decision at every turn: call a tool, ask the user for missing information, or abstain because no available tool can answer. Modern function-calling training data, however, is almost entirely composed of valid tool calls (Liu et al., 2024; Patil et al., 2025). Supervised fine-tuning on such data consistently degrades the non-call axes, producing models that over-call tools on queries they should have questioned or refused. Lin et al. (2025) and Ross et al. (2025) both document this execution–calibration interference and treat it as a primary failure mode of current tool-use fine-tuning.

Existing fixes either add data, by mixing in counterexamples that demand abstention or clarification (Lin et al., 2025), or change the objective, by replacing SFT with preference optimisation over paired calibration examples (Ross et al., 2025). Neither asks a more basic question: can we distil calibration from a moderately competent off-the-shelf model into a student that is otherwise fine-tuned in the standard way? Our answer is yes, provided the teacher is not trusted literally. The empirical error pattern of the teacher is itself useful supervision. Once we correct for that pattern, a weaker teacher supplies a compact auxiliary signal that improves both OOD accuracy and seed-to-seed stability.

Direct logit distillation does not work here. The only teacher we have access to that is even partially calibrated on the three-way decision is itself quite imperfect. Our teacher (Qwen2.5-7B) has overall accuracy 0.56 on the 3-class task, and mislabels 82% of the request-for-information class as `tool_call`. In absolute terms it is weaker than the base Qwen3-4B student. A 4-seed naïve-KD control (NAÏVE-KD) confirms that distilling its soft predictions directly is not competitive with CCSD in our setting: NAÏVE-KD underperforms CCSD-hard by 10.8 pp macro-F1 on OOD-1 and 9.5 pp on OOD-2 (see Table 2). We also compare against a DPO-style objective on chosen/rejected mode pairs constructed from the When2Call MCQ training data (RPO; full configuration in Section 4.1), following the spirit of Ross et al. (2025)'s reward-weighted policy alternative to SFT. In our single matched-budget RPO instantiation (no preference-optimisation hyperparameter sweep), RPO underperforms CCSD-hard by 14.7 pp on OOD-1 and

8.6 pp on OOD-2; we do not claim this rules out preference optimisation in general.

Key insight. The teacher's errors are structured rather than random. Its confusion matrix $R_{ij} = \mathbb{P}(\text{gold} = j \mid \text{teacher-predicts} = i)$ is stable across batches and can be estimated cheaply from the training set itself. Patrini et al. (2017) show that, given a known class-conditional noise transition matrix, one can train on noisy labels by forward correction: multiply the label distribution by the transition matrix. We import this idea as a heuristic into distillation. The teacher's soft labels play the role of noisy annotations, and R plays the role of an empirical reliability matrix. The resulting target $q_{\text{cal}} = p_{\text{teacher}}^{\top} R$ is used as a fixed soft target in an auxiliary $\mathrm{KL}(q_{\text{cal}} \| p_{\text{student}})$ term on top of standard full-sequence SFT. This small change is enough to turn an otherwise weak teacher into a useful calibration source in our setting: on OOD-1, CCSD-hard gains 10.8 pp over direct logit distillation and reduces the focal baseline's seed sensitivity by an order of magnitude. We do not claim that our R satisfies the independence assumptions required for the formal unbiasedness guarantees of Patrini et al. (2017); we use the construction as a heuristic and evaluate it empirically.

Contributions. We introduce Confusion-Calibrated Supervision Distillation (CCSD), which treats a teacher as a structured noisy annotator and uses its empirical reliability matrix to derive an auxiliary distillation target. The construction is inspired by forward correction for noisy labels (Patrini et al., 2017), but we use it as a heuristic rather than a formal instantiation. We study a continuous soft variant and a bucket-indexed hard variant.

On OOD-1, CCSD-hard improves macro-F1 by 5.3 pp over a focal-loss auxiliary baseline, 10.8 pp over naïve logit distillation, and 14.7 pp over our DPO-style RPO instantiation. The stability gain is larger than the mean gain: FOCAL's 4-seed tool-F1 standard deviation is roughly $20\times$ CCSD-hard's, and one of the four focal seeds drops to macro 0.595 (tool-F1 0.295), while CCSD-hard stays in 0.747–0.759 macro-F1.

Further results offer supportive evidence. On a held-out APIGen OOD split with independently constructed refusals, CCSD's macro-F1, tool-F1, and over-call point estimates favor CCSD across all four seeds (the smallest delta, on seed 44, is -0.005, well within seed noise); the pooled bootstrap reports $p<0.05$, while the seed-level test is inconclusive at $n=4$. A 2-seed Llama replication preserves the method ranking and focal high-variance pattern. Per-step training trajectories further show that FOCAL's peak P(refuse) varies 2–3$\times$ more across seeds than CCSD's on a fixed probe set. This is descriptive evidence for early-training volatility, not a causal mechanism claim. Single-seed ablations show that estimating R from training data matches an oracle holdout estimate within 2 pp, and that CCSD is flat over $\lambda \in [0.5, 1.0]$.

Figure 1 summarises the technical route. The remainder of the paper formalises the setup (Section 3), benchmarks CCSD against the primary distillation, reweighting, preference-optimisation, and SFT baselines on one in-distribution and two out-of-distribution test sets, adds a cross-backbone replication (Section 4), and ablates R, λ, and the soft/hard variants (Section 5).

2 Related Work

Our work sits at the intersection of four lines of research. To our knowledge none has previously been combined with the others.

Tool-use invocation calibration. Recent function-calling benchmarks (Patil et al., 2025; Liu et al., 2024) treat tool selection and argument filling as the primary objective, and they do not ask whether a tool should be called at all. Lin et al. (2025) and Ross et al. (2025) are the first to document a systematic regression on the invocation decision after SFT on standard function-calling data, which they call execution–calibration interference. Hammer addresses it by masking and data mixing; When2Call addresses it by learning from paired preference data via a reward-weighted policy objective. Both remedies reshape the training distribution but keep the supervision token-level. CCSD, by contrast, keeps the SFT distribution fixed and adds a single auxiliary distillation term that targets the invocation distribution directly.

LLM calibration and selective prediction. A related body of work asks when an LLM should abstain rather than answer (Kadavath et al., 2022; Yoshikawa and Okazaki, 2023). Selective prediction and calibration methods typically act at inference time, through confidence thresholds, self-consistency, or verifier models, and do not change the fine-tuning objective. Our method changes training so that the student's first-token invocation

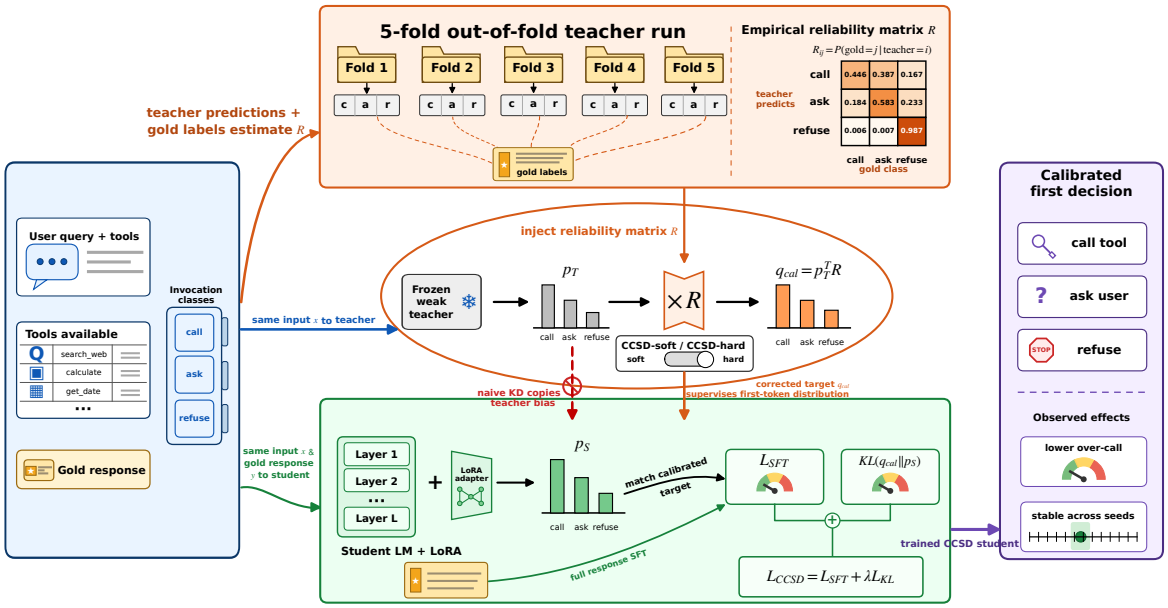

Figure 1: CCSD technical route. We estimate a teacher reliability matrix $R = P(\text{gold} \mid \text{teacher})$ by 5-fold out-of-fold on the fine-tuning data, then use R to convert the teacher's first-token distribution into a corrected target $q_{\text{cal}} = p_{\text{teacher}}^{\top} R$. The student is trained with the usual full-sequence SFT loss plus a first-token $\text{KL}(q_{\text{cal}} \| p_{\text{student}})$ auxiliary term. The central design choice is to transfer the teacher's confusion structure rather than copying its biased soft labels directly.

distribution is already calibrated, without needing an extra inference-time mechanism. The two directions can be combined.

Knowledge distillation with imperfect teachers. Starting from Hinton et al. (2015), knowledge distillation usually assumes that the teacher provides a more accurate signal than the student. Recent work studies sequence-matching objectives (Agarwal et al., 2024) and reasoning-chain distillation (Hsieh et al., 2023); in all of these the assumed hierarchy is teacher $\succ$ student. A smaller line of work has asked what happens when the teacher is itself unreliable. For example, teacher-annealing mixes teacher labels with gold (Clark et al., 2019), and label-smoothing variants (Yuan et al., 2020; Müller et al., 2019) show that coarser teacher signals can regularise. Our teacher (overall accuracy 0.56) is in fact weaker than the student base, and straight logit distillation from it is not competitive (Table 2). Our contribution is to show that a weaker teacher can still be useful when its errors are characterised: applying the empirical reliability matrix redistributes the teacher's biased marginal, and the auxiliary loss then pulls the student away from that bias.

Learning with noisy labels. Patrini et al. (2017) establish forward and backward loss correction when a class-conditional noise transition matrix is known. Hendrycks et al. (2018) extend this to the setting where a small trusted dataset is available for matrix estimation, and Han et al. (2018) use peer networks to filter mislabelled examples. This body of work was developed for image classification with human-provided labels and assumes label noise that is conditionally independent of the input. Our "noisy annotator" is a teacher network whose predictive distribution is itself the corrupted label, and the class-conditional independence assumption does not cleanly apply. We therefore use the construction as a heuristic inspired by forward correction rather than as a direct instantiation of the formal framework.

Focal loss and calibration. Lin et al. (2017) introduce focal loss as a sample-level reweighting that up-weights hard or rare examples. Mukhoti et al. (2020) show that focal loss tends to improve calibration as a byproduct. Focal is thus a natural baseline for our auxiliary loss and is our closest competitor in the empirical section. The distinction is mechanistic: focal operates at the sample level, by reweighting instances according to cur-

rent prediction confidence, while CCSD operates at the distribution level, by targeting a corrected label distribution. We test whether this distinction matters empirically in Section 4.

3 Method

3.1 Setup and notation

We consider a three-way invocation decision $c \in \mathcal{C}$, with $\mathcal{C} = \{\texttt{call}, \texttt{ask}, \texttt{refuse}\}$ denoting tool call, request-for-information, and cannot-answer. A student language model S and a frozen teacher T both receive the same user query together with the list of available tools, and each produces a first response token. We define a verbalizer $v(c)$ that maps each class to a short lexical item ("`call`", "`ask`", "`refuse`") and read off the three-class marginals by restricting and renormalising the next-token logits on this set. This gives two distributions $p_{\text{teacher}}, p_{\text{student}} \in \Delta^2$ on the three classes for each training example.

Our student is Qwen3-4B-Instruct (Yang et al., 2025a) with LoRA adapters of rank 16 (Hu et al., 2022); our teacher is Qwen2.5-7B-Instruct (Yang et al., 2025b), chosen for tokenizer compatibility. The student receives full-sequence supervised fine-tuning loss $\mathcal{L}_{\text{SFT}}$ on the gold response as the primary objective.

Teacher quality. On our 6000-example fine-tuning set (a 2:2:2 balanced mixture of APIGen tool calls, W2C requests-for-information, and W2C cannot-answer cases), the teacher's overall accuracy is 0.56. Its per-class accuracy is highly non-uniform: 0.94 on `tool_call`, 0.17 on `request_for_info`, and 0.57 on `cannot_answer`. In absolute terms the teacher is weaker than the Qwen3-4B base student. What it has is a structured bias: 82% of `request_for_info` gold examples are misclassified as `tool_call`. The key question is whether this structure can be used without the student also inheriting the bias.

3.2 The reliability matrix R

Inspired by loss-correction methods for noisy labels (Patrini et al., 2017), we describe the teacher as a structured noisy annotator and estimate a row-stochastic reliability matrix

$$R_{ij} \;=\; \mathbb{P}\big(\text{gold} = j \;\big|\; \arg\max_k p_{\text{teacher}}(k) = i\big), \tag{1}$$

i.e. the empirical conditional distribution of the gold class given the teacher's hard prediction. We estimate R on 4998 training examples (the "R-train" split) by 5-fold out-of-fold: for each fold we run the teacher on the held-out split using the other four for context, then aggregate and row-normalise. Table 1 shows the resulting matrix.

Table 1: Estimated reliability matrix R (5-fold OOF on R-train, n=4998). Row i: when the teacher predicts class i, these are the probabilities of each gold class. The teacher is reliable only when it predicts `cannot_answer`; its `tool_call` predictions are almost random among the three classes. Cells above 0.5 are bolded.

Teacher prediction	n	P(tool)	P(req)	P(cannot)
tool_call	3515	0.446	0.387	0.167
request	512	0.184	**0.583**	0.233
cannot_answer	971	0.006	0.007	**0.987**

Note that R here plays the role of an empirical reliability matrix, not the symmetric class-conditional noise transition matrix assumed by the formal analysis of Patrini et al. (2017). In particular we do not assume the teacher's errors are independent of the input, and we do not claim that the corrected target is unbiased in their sense. We use the construction as a heuristic inspired by forward correction and evaluate it empirically.

The structure of R tells us when the teacher is reliable: its `cannot_answer` prediction is almost always correct ($P(\text{gold} = \text{cannot} \mid \text{teacher-cannot}) = 0.987$), but its `tool_call` prediction is only weakly informative ($P(\text{gold} = \text{tool} \mid \text{teacher-tool}) = 0.446$, barely above the prior) because it absorbs many `request_for_info` examples. A student that naïvely distils the teacher's `tool_call` soft label will inherit this bias.

3.3 Forward-corrected auxiliary distillation

Given R, we define a corrected target by analogy with forward correction in noisy-label learning (Patrini et al., 2017):

$$q_{\text{cal}} \;=\; p_{\text{teacher}}^{\top} R \;\in\; \Delta^2. \tag{2}$$

Intuitively, q_{cal} redistributes each unit of the teacher's mass on class i across the gold classes according to the empirical conditional $R_{i,\cdot}$. We distil q_{cal} into the student's first-token distribution, but also keep a gold-grounded cross-entropy term so the gradient stays well-defined when q_{cal} becomes nearly uniform. The auxiliary loss is the

Algorithm 1 CCSD training step

Require: mini-batch (x, y), teacher T, student S, $R \in \mathbb{R}^{3\times 3}$, weight λ, variant $v \in \{\text{soft}, \text{hard}\}$

$p_{\text{teacher}} \leftarrow \text{softmax}(\text{verb-token logits}, T(x))$
$p_{\text{student}} \leftarrow \text{softmax}(\text{verb-token logits}, S(x))$
if $v = \text{soft}$ **then** $q_{\text{cal}} \leftarrow p_{\text{teacher}}^{\top} R$
else $q_{\text{cal}} \leftarrow R_{\arg\max p_{\text{teacher}},:}$
end if
$\mathcal{L}_{\text{SFT}} \leftarrow \text{cross-entropy}(S(x), y)$ ▷ full-sequence SFT
$\mathcal{L}_{\text{aux}} \leftarrow \gamma\, \text{KL}(q_{\text{cal}} \,\|\, p_{\text{student}}) + (1-\gamma)\, \text{CE}(p_{\text{student}}, y_{\text{1st}})$
$\mathcal{L} \leftarrow \mathcal{L}_{\text{SFT}} + \lambda\, \mathcal{L}_{\text{aux}}$; back-propagate into LoRA params

equal-weight blend

$$\mathcal{L}_{\text{aux}} = \gamma\, \text{KL}\big(q_{\text{cal}} \,\|\, p_{\text{student}}\big) + (1-\gamma)\, \text{CE}\big(p_{\text{student}},\, y_{\text{1st}}\big), \tag{3}$$

with γ=0.5 throughout. Here $\text{CE}(p_{\text{student}}, y_{\text{1st}})$ is the cross-entropy of p_{student} against the gold first-token class, which equals $-\log p_{\text{student}}(y_{\text{1st}})$. The full training loss combines standard SFT with this auxiliary blend:

$$\mathcal{L} = \mathcal{L}_{\text{SFT}} + \lambda\, \mathcal{L}_{\text{aux}}. \tag{4}$$

We study two variants that differ only in how the teacher output enters Equation (2). CCSD-soft uses the teacher's continuous first-token distribution, $q_{\text{cal}} = p_{\text{teacher}}^{\top} R$. CCSD-hard uses only the teacher's bucket identity, $q_{\text{cal}} = R_{\arg\max p_{\text{teacher}},:}$, the row of R corresponding to the teacher's hard prediction. CCSD-hard discards the teacher's confidence and treats every prediction within a bucket identically; empirically, this variant has lower seed variability on OOD-1 (Section 4).

3.4 Why a weaker teacher can still help

The signal available to the auxiliary loss depends on the off-diagonal structure of R, not on the teacher's overall accuracy. If R were the identity, Equation (2) would reduce to distilling p_{teacher} directly, and the student would inherit the teacher's class bias. When R has structured off-diagonals (as in Table 1), applying it redistributes a teacher "`tool_call`" prediction according to [0.446, 0.387, 0.167], which reflects empirically that a non-trivial fraction of such predictions correspond to `request_for_info` gold. The auxiliary loss therefore pulls the student away from the teacher's marginal bias. This is a heuristic argument, not a formal unbiasedness guarantee; we evaluate it empirically in Section 5.1, where replacing R with the identity costs 5.8 pp of OOD-1 macro-F1 and raises the over-call rate from 0.156 to 0.383.

4 Experiments

4.1 Setup

Training data. We fine-tune the student on a balanced 6000-example mixture: 2000 `tool_call` examples sampled from APIGen (Liu et al., 2024), and 2000 `request_for_info` plus 2000 `cannot_answer` examples sampled from the When2Call training set (Ross et al., 2025). We reserve 4998 of these examples as the R-train split used to estimate R and 1002 as an R-holdout split used only for the oracle ablation in Section 5.2.

Test sets. We evaluate on three test splits. The ID llm_judge split (n=300) is the When2Call evaluation split, balanced 100/100/100. OOD-1 BFCL (n=384) is constructed from the Berkeley Function Calling Leaderboard v3 `live_simple` and `live_irrelevance` subsets (Patil et al., 2025), then re-labelled into the three-way scheme and balanced 128/128/128. OOD-2 APIGen (n=384) is a held-out split built from APIGen queries that were fully excluded from training, augmented with 128 refusal cases constructed independently by pairing each query with a tool from a different functional family. OOD-2 shares the query source with the APIGen tool-call training data but has no query overlap, and its refusal construction does not reuse any refusal source from training.

Baselines. We compare against baselines that isolate the main design choices, named by their distinguishing characteristic rather than by code. W2C-ONLY is full-sequence SFT on the When2Call training set only. NAÏVE-KD uses the same SFT mixture plus a first-token KL distillation loss against the teacher's raw soft labels, with no R correction; this is our negative control for the direct-distillation failure claim in Section 1. FOCAL uses the same SFT mixture plus a first-token focal cross-entropy auxiliary on the gold verbalizer token (Lin et al., 2017; Mukhoti et al., 2020) (γ=2.0), in place of CCSD's KL target.

We also include RPO, a DPO-style preference-optimisation baseline inspired by Ross et al. (2025)'s reward-weighted policy objective. RPO

is trained on 2000 chosen/rejected preference pairs constructed from the When2Call MCQ training data, where the gold mode answer is chosen and one randomly sampled non-gold mode answer is rejected. We use β=0.1 for the DPO term and add an SFT anchor of weight 0.2 on the chosen response. Learning rate, optimizer (LoRA r=16), and total optimiser steps (560) are matched to CCSD. Reference logits are obtained by toggling the LoRA scaling to zero on the same parameters, with no separate reference checkpoint. RPO provides a preference-optimisation alternative that does not use teacher distillation.

Protocol. Every method is trained for a matched budget of 560 optimiser steps (8960 micro-batches, gradient accumulation 16). We evaluate with 4 seeds $\{42, 43, 44, 45\}$ and report mean $\pm$ standard deviation across seeds. For significance we report two complementary analyses: (i) a within-sample paired bootstrap (B=10,000) on per-sample $(y_{\text{true}}, y_{\text{pred}})$ pooled across the 4 seeds ($n_{\text{sample-seed pairs}}$=1536), which has high power but treats each sample–seed observation as independent; and (ii) a seed-level paired t-test on 4 per-seed deltas, which respects the dependency structure but has low power at n=4. We report macro-F1 as the primary metric, with per-class F1 (tool, request, cannot) and over-call rate $P(\hat{y} = 0 \mid y \neq 0)$.

4.2 Main results: ID and OOD-1

Table 2 reports 4-seed aggregates for CCSD and four primary baselines (FOCAL, NAÏVE-KD, RPO, W2C-ONLY); Figure 2 visualises the OOD macro-F1 comparison. The main result is that CCSD-hard improves the OOD-1 mean while sharply reducing seed sensitivity. On BFCL OOD-1, CCSD-hard reaches macro-F1 0.753 ± 0.006, 5.3 pp above FOCAL (0.700 ± 0.070), 10.8 pp above NAÏVE-KD (0.645 ± 0.040), and 14.7 pp above RPO (0.606 ± 0.075). The stability gain is even more pronounced than the mean gain: CCSD-hard's macro-F1 seed standard deviation is roughly $12\times$ smaller than FOCAL's (0.006 vs. 0.070), and on tool-F1 the ratio is roughly $20\times$ (CCSD-hard 0.010 vs. FOCAL 0.194). These results match the intended setting for CCSD: a teacher too noisy to distil directly, but structured enough that its reliability matrix can correct the auxiliary target.

The OOD-1 variance gap is driven by a concrete failure mode. Three of FOCAL's four seeds fall in the range 0.730–0.742; the fourth (seed 45) drops to macro 0.595 with tool-F1 0.295. CCSD's four seeds lie in 0.747–0.759, and we do not observe a drop of comparable magnitude in any CCSD run across our four seeds. A leave-one-seed-out check gives a more cautious reading. FOCAL's macro-F1 standard deviation of 0.070 is dominated by this single low seed: dropping seed-45 leaves FOCAL at 0.736 ± 0.007 over the remaining three seeds, compared to CCSD-hard's median of 0.754 and minimum of 0.747. The 4-seed mean and median therefore differ by 3.2 pp for FOCAL versus 0.05 pp for CCSD-hard. Thus, even under this median-based view, CCSD-hard remains above FOCAL; the variance gap is what shrinks under leave-one-out. The pattern suggests that FOCAL has a non-trivial probability of a low-performing seed in our setup, rather than being lower on every seed. RPO, despite using preference data the other methods do not, also exhibits high variance (macro std 0.075) and a substantially higher over-call rate (0.462 ± 0.121 versus CCSD-hard's 0.122 ± 0.018).

4.3 OOD-2: held-out APIGen with independent refusal construction

OOD-2 tests whether the method still helps when the queries come from the same source as training but the refusal cases are constructed independently. Every OOD-2 query is held out from APIGen, with no uuid or hash overlap with training. The 128 refusal cases pair a weather query with a stock-price tool, a booking query with a translation tool, and so on, using a generation procedure that shares no data pipeline with the refusal examples the student was trained on.

CCSD-soft achieves 0.821 ± 0.013 and CCSD-hard 0.820 ± 0.008 macro-F1 on OOD-2, against 0.806 ± 0.018 for FOCAL and 0.783 ± 0.025 for W2C-ONLY. We report two complementary significance analyses in Figure 3:

Within-sample pooled bootstrap. Pooling per-sample predictions across seeds and resampling examples (B=10,000) gives CCSD-soft vs. FOCAL Δmacro $= +0.014$, p=0.006; Δtool $= +0.011$, p=0.049; Δover-call $= -0.016$, p=0.010. This analysis has high power but does not fully respect the dependency between seeds; it should be read as within-sample rather than across-training-run evidence.

Seed-level paired t-test. Computing per-seed deltas and testing against zero with n=4 gives CCSD-soft vs. FOCAL Δmacro $= +0.015$,

Table 2: Main results on three test splits. All rows report mean ± std over 4 seeds (42/43/44/45). The best value in each column is bolded. Splits: ID llm_judge (When2Call, n=300); OOD-1 BFCL (constructed from BFCL v3 live, n=384); OOD-2 APIGen (held-out queries with independent refusal construction, n=384). Columns: ID macro, ID cannot-F1, ID tool-F1, OOD-1 macro, OOD-1 tool-F1, OOD-2 macro, OOD-2 tool-F1. CCSD-hard is best or tied on every out-of-distribution column except OOD-2 macro-F1, where CCSD-soft is ahead by 0.1 pp. The largest separation is on OOD-1, where FOCAL, NAÏVE-KD, and RPO trail CCSD-hard by 5.3, 10.8, and 14.7 pp respectively; on OOD-2 the CCSD–FOCAL gap is smaller but has the same sign.

Method	ID macro	ID cannot	ID tool	OOD-1 macro	OOD-1 tool	OOD-2 macro	OOD-2 tool
CCSD-hard (ours)	**0.669±0.029**	0.750±0.015	0.589±0.055	**0.753±0.006**	**0.721±0.010**	0.820±0.008	**0.787±0.009**
CCSD-soft (ours)	0.658±0.009	0.757±0.009	0.553±0.031	0.743±0.007	0.698±0.017	**0.821±0.013**	0.786±0.014
Focal	0.634±0.057	**0.801±0.026**	0.441±0.187	0.700±0.070	0.585±0.194	0.806±0.018	0.775±0.005
Naïve-KD	0.622±0.008	0.668±0.022	0.630±0.014	0.645±0.040	0.675±0.022	0.725±0.035	0.731±0.020
RPO	0.582±0.061	0.580±0.070	0.687±0.035	0.606±0.075	0.667±0.041	0.734±0.030	0.724±0.014
W2C-only	0.589±0.028	0.525±0.039	**0.650±0.038**	0.617±0.035	0.698±0.024	0.783±0.025	0.768±0.016

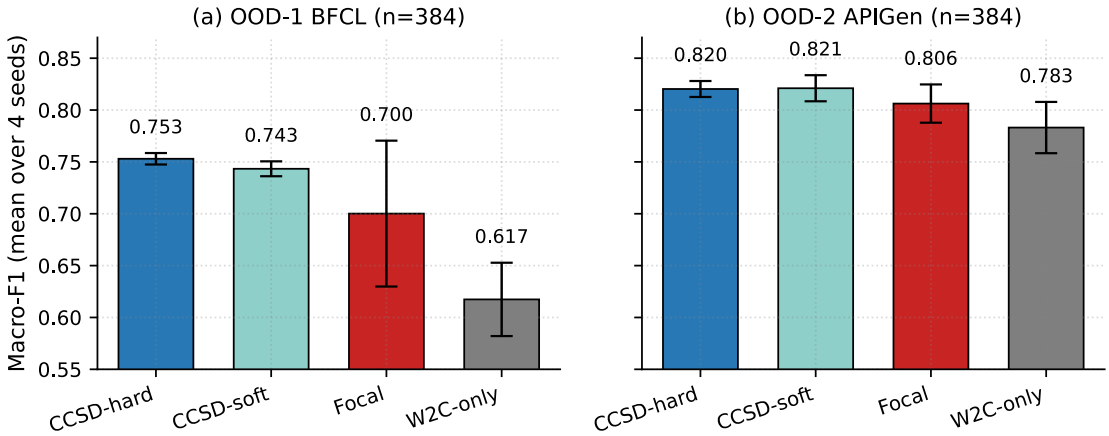

Figure 2: Macro-F1 on the two out-of-distribution splits, averaged over 4 seeds. Error bars are standard deviation. CCSD leads on both. OOD-2 reuses the APIGen query source but is fully held out from training with independently-constructed refusals; the within-sample pooled bootstrap favors CCSD on OOD-2, while the seed-level test is inconclusive at n=4 (Section 4.3). FOCAL's large error bar on OOD-1 is driven by a single outlier seed discussed in Section 4.2.

$t(3)$=1.88, p=0.157; Δtool = +0.011, p=0.170; Δover-call = −0.016, p=0.382. All three point estimates are in the same direction as the pooled analysis. Per-seed macro-F1 deltas favor CCSD on three of the four seeds (the fourth, seed 44, is −0.005, well within seed-level noise of typical magnitude ∼ 0.02), but with n=4 the seed-level test does not cross p=0.05.

Taken together, the two analyses point in the same direction. Because they respect different sources of randomness, we report both tests instead of collapsing them into a single significance statement.

4.4 Training-step trajectory: FOCAL has wider P(refuse) excursions than CCSD

To localise the variance gap, we log the student's first-token distribution every 20 training steps on a fixed 128-sample probe set for FOCAL and both CCSD variants × 4 seeds. FOCAL's training-time peak P(refuse) has across-seed standard deviation 0.10, versus 0.04 for CCSD-soft and 0.03 for CCSD-hard; the largest excursion is FOCAL seed-44 at P(refuse) = 0.604, while no CCSD seed exceeds 0.477. All methods converge near P(refuse) ≈ 0.35 by step ∼ 250; the differentiating signal is early-training volatility. This is descriptive evidence for early-training volatility; the full trajectory plot is in Appendix B.

4.5 Cross-backbone replication on Llama

To check whether the CCSD-vs-focal pattern is specific to the Qwen family, we also run a directional replication with a Llama-3.2-3B-Instruct student and a Llama-3.1-8B-Instruct teacher (2 seeds; full table in Appendix A). The teacher is weaker and has a different bias structure (refuse-heavy rather than tool-heavy), but the ranking is preserved

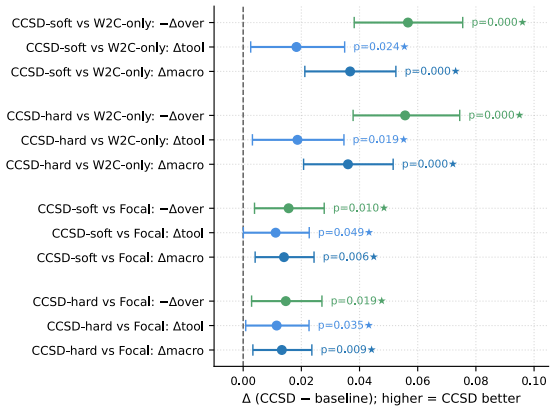

Figure 3: Within-sample paired bootstrap on OOD-2 (4 seeds pooled per sample, n=1536, B=10,000). Positive Δ means CCSD is better; for over-call we plot $-\Delta$ so that all three columns point in the same direction. The pooled bootstrap gives $p < 0.05$ for every CCSD-vs-FOCAL column shown next to each CI; the seed-level paired test (n=4) is inconclusive (Section 4.3).

across all three splits. On OOD-2, CCSD-hard improves +12.6 pp over FOCAL (0.744 vs. 0.618), and FOCAL again has the largest seed standard deviation (0.097 on macro-F1 vs. 0.011 for CCSD-hard).

5 Analysis and Ablations

This section isolates three design choices: whether the off-diagonal structure of R matters, whether R can be estimated reliably from training data alone, and how sensitive CCSD is to λ. The ablations in this section use a single seed (seed 42) unless noted; the 4-seed results reported in Section 4 use the same λ and estimated R.

5.1 Is R doing anything? An R-matrix ablation

We re-train CCSD-soft (single seed) with four different choices of R while holding every other hyperparameter fixed: the 5-fold OOF estimate used throughout, the oracle matrix fit on the held-out 1002-example R-holdout split, the uniform matrix $R = \mathbf{1}/3$, and the identity $R = I$.

Replacing R with the identity costs 5.8 pp of OOD-1 macro-F1 (from 0.749 to 0.691) and increases over-call rate from 0.156 to 0.383; the full table is in Appendix C. This rules out the hypothesis that the student learns from SFT alone while the auxiliary loss contributes nothing: the identity target is still a live gradient signal, but it lacks the off-diagonal correction.

5.2 How sensitive is CCSD to the choice of R?

The oracle matrix, fit on a labelled holdout of 1002 examples, achieves OOD-1 macro-F1 0.736 in the single-seed run. The 5-fold OOF matrix, which requires no extra labels, achieves 0.749 and stays within 2 pp across every metric we tested. This shows that R can be estimated directly on the SFT data itself, without a separate labelled resource.

5.3 Sensitivity to the auxiliary-loss weight λ

We sweep $\lambda \in \{0.5, 1.0, 2.0, 4.0\}$ at fixed R (single seed; plot in Appendix C). On OOD-1, macro-F1 is $\{0.741, 0.749, 0.720, 0.705\}$: within 1 pp over $[0.5, 1.0]$, declining monotonically for $\lambda > 1$. We use $\lambda = 1.0$ throughout.

5.4 Soft versus hard CCSD

The within-sample pooled bootstrap on OOD-2 finds no significant difference between CCSD-soft and CCSD-hard on macro-F1, tool-F1, or over-call rate (all $p > 0.5$, n=1536); the seed-level test likewise gives $p > 0.4$ at n=4. On OOD-1, CCSD-hard's mean macro-F1 is 1.0 pp above CCSD-soft's, within the 4-seed standard deviation. The two variants are therefore statistically indistinguishable under both analyses. We use CCSD-hard as the default because it is simpler to implement, requiring only $\arg\max p_{\text{teacher}}$ rather than the full soft distribution, and its observed tool-F1 standard deviation is 0.010 versus 0.017 for CCSD-soft, although this difference itself is not tested.

6 Conclusion

We introduced CCSD, a weak-teacher distillation objective that treats the teacher's invocation distribution as a noisy annotation and calibrates it with a 5-fold out-of-fold reliability matrix. Across Qwen experiments, CCSD-hard improves OOD-1 macro-F1 by 5.3 pp over focal reweighting, 10.8 pp over naïve KD, and 14.7 pp over RPO while sharply reducing seed variance. OOD-2, Llama, and ablation results provide supportive but more limited evidence: OOD-2 is significant under a within-sample pooled bootstrap but inconclusive under a seed-level test at n=4; the Llama replication is directional at n=2, with focal's high-variance pattern reappearing only on OOD-2; and the R/λ ablations are single-seed.

Limitations

Reliability matrix is class-distribution dependent. R is estimated on the same class distribution as the training data. Deployment scenarios with large prior shifts (e.g., a domain where refusals are far rarer than in training) would require R to be re-estimated from a small labelled probe set drawn from the deployment distribution; we have not tested this setting.

Cross-backbone evidence is limited by the number of seeds. Our cross-backbone replication on Llama-3.2-3B / Llama-3.1-8B uses 2 seeds rather than 4. The ranking of methods and the focal-vs-CCSD seed-variance pattern carry over, but the Llama numbers provide directional support rather than a statistically independent confirmation.

OOD-2 seed-level test is inconclusive at n=4. On OOD-2, the within-sample pooled bootstrap reports $p<0.05$ on macro-F1, tool-F1, and over-call, but a seed-level paired t-test at n=4 does not cross p=0.05 (e.g., macro p=0.157). We therefore treat OOD-2 as supportive evidence that the OOD-1 advantage is not specific to the BFCL labelling, rather than as an independently significant generalisation result.

The RPO comparison is a single instantiation. We compare against one specific DPO-style preference-optimisation configuration (β=0.1, SFT anchor λ=0.2, 2000 chosen/rejected pairs, matched 560 optimiser steps). We do not tune RPO carefully and we do not claim our results rule out preference-optimisation methods in general; they only show that this particular instantiation underperforms CCSD in our setup.

No causal mechanism for the focal-loss seed sensitivity. The focal-loss training dynamics reported in the main analysis are descriptive rather than causal. We do not run interventions such as a focal-γ sweep, auxiliary-weight ablation, or on-stream prediction-rate tracking that would let us attribute the observed seed sensitivity to a specific feedback path. The mechanism behind the observed volatility therefore remains unverified.

Three-way invocation labels are coarse. The invocation decision is reduced to three classes (call / ask / abstain). Finer-grained calibration, such as per-parameter argument confidence, multi-turn invocation chains, or structured tool selection, is outside the present scope. CCSD's mechanism is class-level, so extending to finer-grained settings would require redefining what counts as a noisy-annotator class.

No end-to-end agent metric. We evaluate at the invocation-decision level only. We do not measure downstream tool-use success (e.g., whether the called tool actually executes successfully and the user's intent is satisfied). A full agent benchmark would test whether reduced over-call rate translates into measurable user-task improvement.

Ethics Statement

This paper studies an invocation-calibration subtask of tool-using language models. Our changes only affect when a model decides to invoke a tool, not what the tool does once invoked. The intended effect of CCSD is to reduce false tool calls, specifically by teaching a student model to ask the user for missing information or to abstain instead of invoking a tool inappropriately. In deployment terms, this should reduce a class of agent failures (silent fabrication of tool arguments, executing tools on unsupported queries) rather than enable new capabilities.

Datasets. We use only public, English-language datasets: When2Call (NVIDIA, NAACL 2025), APIGen (NeurIPS 2024), and the Berkeley Function Calling Leaderboard v3 (ICML 2025). All are released for research use under permissive terms. We do not introduce new human-annotated data; the only human labels we rely on are those already present in the source benchmarks.

Models. The teacher (Qwen2.5-7B-Instruct) and students (Qwen3-4B-Instruct, Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct) are publicly released by their respective maintainers under the model licenses listed in their model cards. We use them as-is for fine-tuning experiments and do not redistribute weights.

Compute. Each training run uses one consumer-grade GPU and approximately 45 minutes of wall-clock time. Total compute for the experiments reported here is on the order of a few GPU-days. We do not believe this constitutes a frontier-scale energy footprint.

Potential dual-use. The CCSD framework is a generic distillation method; it could in principle be used to transfer any class-level decision pattern

from a teacher to a student, including patterns that are not socially desirable. We see no specific dual-use concern arising from the invocation-calibration task we study, but caution that any framework that adjusts a model's decision distribution requires the user to verify that the target distribution itself is well-aligned.

Reproducibility. Code and trained adapters will be released under a permissive license upon publication. Hyperparameters and the exact preprocessing pipeline are in Appendix D; per-sample evaluation outputs (predictions and labels for every test example) will be released upon publication alongside the code so that the reported bootstrap statistics can be independently verified.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems 31*, pages 8536–8546.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems 31*, pages 10477–10486.

Geoffrey E. Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Qiqiang Lin, Muning Wen, Qiuying Peng, Guanyu Nie, Junwei Liao, Jun Wang, Xiaoyun Mo, Jiamu Zhou, Cheng Cheng, Yin Zhao, Jun Wang, and Weinan Zhang. 2025. Robust function-calling for on-device language model via function masking. In *The Thirteenth International Conference on Learning Representations*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh R N, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. 2024. APIGen: Automated PIpeline for generating verifiable and diverse function-calling datasets. In *Advances in Neural Information Processing Systems 37*, pages 54463–54482.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet K. Dokania. 2020. Calibrating deep neural networks using focal loss. In *Advances in Neural Information Processing Systems 33*, pages 15288–15299.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems 32*, pages 4694–4703.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. The Berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 48371–48392.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2233–2241.

Hayley Ross, Ameya Sunil Mahabaleshwarkar, and Yoshi Suhara. 2025. When2Call: When (not) to call tools. In *Proceedings of the 2025 Conference of the*

Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 3391–3409.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 42 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 24 others. 2025b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Hiyori Yoshikawa and Naoaki Okazaki. 2023. Selective-LAMA: Selective prediction for confidence-aware evaluation of language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2017–2028.

Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3903–3911.

Table 3: Cross-backbone replication on Llama-3.2-3B-Instruct student with Llama-3.1-8B-Instruct teacher (overall teacher accuracy 0.50, lower than the Qwen teacher's 0.56). We report macro-F1 mean±std over 2 seeds (42, 43). Llama is a weaker base than Qwen3-4B in our setup, so all absolute numbers are lower; the relevant result is that the mean ranking is preserved and focal's high-variance pattern reappears on OOD-2 (std 0.097, $\sim 9\times$ CCSD-hard's 0.011); on ID and OOD-1 the focal std is comparable to or smaller than CCSD-hard's at n=2.

Method	ID macro	OOD-1 macro	OOD-2 macro
CCSD-hard (ours)	**0.486±0.022**	**0.491±0.014**	**0.744±0.011**
CCSD-soft (ours)	0.477±0.010	0.480±0.035	0.734±0.008
Focal	0.459±0.010	0.427±0.001	0.618±0.097
W2C-only	0.323±0.014	0.441±0.032	0.554±0.036

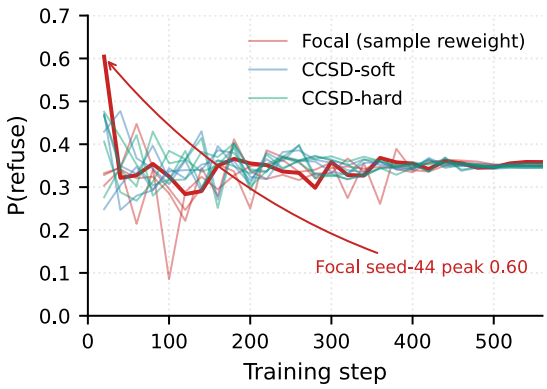

Figure 4: First-token P(refuse) trajectory over 560 training steps, evaluated every 20 steps on a fixed 128-sample probe set, for three methods $\times$ 4 seeds (12 runs total; the W2C-ONLY baseline does not use the auxiliary loss and is omitted). Each thin line is one seed; the thick line annotated "seed-44 excursion" (FOCAL) marks the largest peak among all 12 runs. FOCAL curves (red) exhibit the largest early-training swings; CCSD-soft (blue) and CCSD-hard (green) curves are tightly bundled. Across 4 seeds the peak P(refuse) is 0.46±0.10 for FOCAL, 0.45±0.04 for CCSD-soft, and 0.44±0.03 for CCSD-hard. The across-seed standard deviation of the training-time peak is roughly 2–3$\times$ larger for FOCAL than for either CCSD variant in our setup. By the end of training all three methods converge near $P(\text{refuse}) \approx 0.35$.

A Cross-backbone Llama details

Table 3 reports the 2-seed Llama replication. The Llama-3.1-8B-Instruct teacher has overall accuracy 0.50 on the 3-class probe (vs. 0.56 for Qwen2.5-7B) and shows a refuse-heavy rather than tool-heavy bias structure ($\sim 62\%$ of training samples are predicted as `cannot_answer` by the Llama teacher, vs. Qwen's bias of mislabeling 82% of `request_for_info` as `tool_call`). We trained 2 seeds per method (42, 43). The replication therefore covers a different bias structure from the Qwen result; the relevant signal is the carry-over of method ranking and the focal-vs-CCSD seed-variance pattern, not the absolute numbers.

B Training Dynamics

Figure 4 is descriptive evidence for early-training volatility rather than a causal mechanism test. The probe set is small (n=128 over 28 probes per seed), and we do not run interventions such as a focal-γ sweep, auxiliary-weight ablation, or prediction-rate tracking on the actual training stream.

C Additional Ablations

Table 4 and Figure 5 provide the tabular and plotted versions of the single-seed ablations summarized in Section 5.

D Reproducibility Details

D.1 Hyperparameters

The student uses LoRA rank 16, $\alpha = 32$, dropout 0.05, and target modules $\{q, k, v, o\}_{\text{proj}}$. We train with AdamW at learning rate 2×10^{-4}, weight decay 0, and a cosine schedule with 3% warmup. Each step uses batch size 1 with gradient accumulation 16 (effective batch 16), and all methods run for 560 optimiser steps, or 8960 micro-batches, to match training budget. Mixed precision is bfloat16 throughout, with gradient checkpointing enabled. The auxiliary KL uses weight λ=1.0 and γ=0.5 blending between KL and cross-entropy on gold by default. Verbalizer IDs for `call`, `ask`, and `refuse` are resolved per tokenizer: $\{1618, 2548, 25066\}$ for Qwen tokenizers and $\{1650, 2610, 26122\}$ for Llama-3 tokenizers.

D.2 Test sets

The ID llm_judge split is the NVIDIA When2Call MCQ test split, with 300 samples balanced 100/100/100 across the three modes. OOD-1 BFCL contains 384 samples (128/128/128) constructed from the BFCL v3 `live_simple` and `live_irrelevance` subsets, re-labelled into the three-way scheme by mapping "no relevant tool" to `cannot_answer` and incomplete-argument cases to `request_for_info`. OOD-2 APIGen contains 384 samples (128/128/128): the 128 `tool_call` examples are held out from APIGen with no uuid or hash overlap with training, the 128

Table 4: R matrix ablation (single seed). Identity (no correction) loses 5.8pp OOD-1 macro-F1 and raises the over-call rate from 0.156 to 0.383, showing that the off-diagonal entries of R matter. The 5-fold OOF estimate matches the oracle holdout fit within 2pp, so a separate labeled holdout is not required. The identity row is bolded.

R variant	ID (llm_judge)		OOD-1 (BFCL)		
	macro	cannot	macro	cannot	over ↓
Estimated (5-fold OOF)	0.665	0.759	0.749	0.826	0.156
Oracle (holdout fit)	0.676	0.774	0.736	0.819	0.133
Uniform $R{=}1/K$	0.645	0.766	0.732	0.842	0.125
Identity $R{=}I$	**0.631**	**0.688**	**0.691**	**0.840**	**0.383**

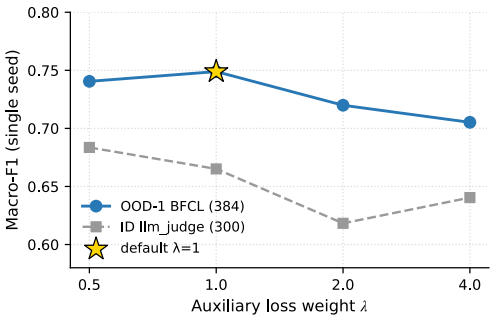

Figure 5: CCSD-soft sensitivity to the auxiliary-loss weight λ on OOD-1. Macro-F1 is flat over $\lambda \in [0.5, 1.0]$ and declines for larger weights.

`request_for_info` examples are generated by an LLM rewrite that drops one required argument, and the 128 `cannot_answer` examples pair an APIGen query with a tool from a different functional family, such as pairing a weather query with a stock-price tool.

D.3 RPO baseline

RPO is trained on 2000 chosen/rejected preference pairs constructed from the When2Call MCQ training data (gold mode answer = chosen; one randomly-sampled non-gold mode answer = rejected, sampled per pair). We use a DPO-style loss with $\beta{=}0.1$ for the preference term and add an SFT anchor of weight 0.2 on the chosen response. Reference logits are obtained by toggling the LoRA scaling to zero on the same parameters (no separate reference checkpoint), so the policy and reference share all base-model weights and differ only by the LoRA delta.

D.4 Statistical analyses

For OOD-2 we report two complementary tests on the same underlying paired data. The within-sample pooled bootstrap uses $B{=}10{,}000$ resamples on per-sample $(y_{\text{true}}, y_{\text{pred}})$ pairs pooled across the 4 seeds, for an effective $n_{\text{sample-seed pairs}}{=}1536$. The two-sided p-value is computed as $2 \cdot \min\{\Pr[\hat{\Delta}_b \leq 0], \Pr[\hat{\Delta}_b \geq 0]\}$ for the observed point-delta direction. The seed-level paired t-test instead uses 4 per-seed deltas and tests against zero with $n{=}4$ (degrees of freedom 3).

D.5 Per-seed result release plan

We attach an anonymous supplementary data archive to this submission containing the per-sample y_{true}, y_{pred}, and first-token probability vectors for every method-seed-split combination reported in Tables 2 and 3 (120 `.npz` files, ∼1.5 MB), the 12 per-step training-dynamics traces underlying Figure 4, the aggregate JSONs underlying all reported means and standard deviations, and a single reproduction script (`verify_stats.py`) that regenerates Tables 2, 3, and the OOD-2 statistical tests from the raw per-sample arrays. Trained LoRA adapters and full training scripts will be released upon publication.