Diagnosing and Calibrating Tool-Call Boundary Drift in Multi-Teacher On-Policy Distillation

Jiabin Shen[1], Guang Chen[2], and Chengjun Mao[3]
[1]tjusjb@gmail.com
[2]cg234573@antgroup.com
[3]chengjun.mcj@antgroup.com

Abstract

Agentic language models must learn when to call tools and when to answer directly. Multi-teacher on-policy distillation is attractive because different teachers can specialize in tool calls and direct responses. We show that this strategy can shift the student's tool-call boundary in a way that aggregate losses miss. In a two-teacher tool-use setting, vanilla generalized knowledge distillation improves tool-call recall but also increases tool-call entry on examples that should be answered directly. Training logs do not support the simplest explanation: tool-call samples receive neither more token exposure nor larger full-sequence per-token divergence. We therefore study the problem as behavior-boundary calibration and compare hard clipping, global reweighting, localized soft compression, and validation-tuned inference-time entry bias. We introduce Soft Clamp as the localized training-time option, which compresses extreme token-level Jensen-Shannon divergence while preserving nonzero gradients. On an APIGen-MT-derived decision set, Soft Clamp reduces over-calling from 14.2±2.1% to 9.0±0.2% while maintaining comparable decision accuracy and retaining much of the call-recall gain over SFT. Global Reweight is a strong comparator, and no strategy dominates every metric and scale. A validation-tuned entry bias reproduces most strict-decoder call-frequency and loop reduction, but only part of the joint call-recall, non-tool-final, and invalid-call profile. These results frame multi-teacher OPD as a behavior-boundary calibration problem: practitioners should monitor where teacher signals act and choose calibration strategies according to locality, observed variability, recall, and deployment trade-offs.

1 Introduction

Agentic language models often act through interleaved dialogue and tool-use trajectories, such as

$$\text{System} \to \text{User} \to \text{ToolCall}_1 \to \text{ToolResponse}_1 \to \text{ToolCall}_2 \to \text{ToolResponse}_2 \to \text{Response}_1 \to \text{User} \to \cdots .$$

Along such trajectories, the model must decide not only *how* to use a tool, but also *whether* to use one and when to stop using tools. This decision is brittle in multi-turn systems. A model that overuses tools increases latency and cost, and it can enter repeated tool-call loops after receiving observations. A model that underuses tools fails tasks that require external information or actions.

Multi-teacher on-policy distillation (MOPD) appears well matched to this problem [11]. A tool-call teacher can specialize in structured function calls; a response teacher can specialize in direct natural-language answers. The student then learns from teacher distributions on its own rollouts, as in generalized knowledge distillation (GKD) [1]. This setup offers specialization without manually merging heterogeneous behaviors.

This paper makes a simple point: multi-teacher OPD can bend a behavior boundary while aggregate loss looks benign. In our APIGen-MT tool-use setting, vanilla GKD improves decision accuracy and tool-call recall, but it also raises over-calling. The model becomes better at calling tools when tools are needed, yet it becomes more likely to call tools when a direct answer is appropriate.

Several aggregate explanations are not supported by the training logs. Tool-call samples do not dominate token exposure. Full-sequence per-token Jensen-Shannon divergence (JSD) and a squared-divergence proxy are not larger for the tool-call teacher. In several runs, response tokens receive more exposure and larger aggregate divergence. The final behavior therefore cannot be explained by sample count, token count, or total divergence alone.

We argue that one missing variable is *behavior leverage imbalance*. A token-level learning signal matters not only by its magnitude, but also by where it lands in the generation process. In a tool-use model, mode-entry tokens such as <tool_call>, function names, and structural markers expose the boundary between direct answers and tool-call trajectories. Natural-language response signals are often spread across many content tokens. This makes multi-teacher OPD vulnerable to boundary bending: a small amount of local signal can redirect the future trajectory. Targeted-clamp diagnostics refine this view. The benefit of Soft Clamp is not reproduced by clamping only the first few entry tokens, so high-leverage signals can also appear at later structural or teacher-dispreferred trajectory positions. We treat this as a diagnostic interpretation, not as a complete causal decomposition.

This observation leads to a calibration comparison. We study hard clipping, Global Reweight, Soft Clamp, and inference-time entry bias as different ways to counter a tool-call boundary shift. Soft Clamp is our localized training-time option. For each token divergence d_i, it uses a detached batch threshold and leaves tokens below it unchanged. Tokens above the threshold are scaled by a detached factor, which caps their forward contribution and scales their gradient without dropping it. Unlike hard clipping, Soft Clamp retains a learning signal for difficult tokens. Unlike global reweighting, it acts only on extremes. Unlike inference-time entry bias, it does not require per-checkpoint validation tuning at deployment.

Our contributions are:

- **Failure discovery.** We identify a reproducible tool-call boundary drift in multi-teacher OPD: vanilla GKD improves should-call recall but also increases tool-call entry on should-respond examples.
- **Diagnostic framework.** We connect aggregate training statistics, sparse token-level divergence concentration, response-side entry pressure, full-generation decisions, and multi-turn loop behavior. These diagnostics support a behavior-leverage interpretation without claiming a complete causal decomposition.
- **Calibration comparison.** We compare hard clipping, global reweighting, localized soft compression, and validation-tuned inference-time entry bias, exposing trade-offs in gradient preservation, locality, observed seed variability, call recall, and deployment requirements.
- **Localized training-time option.** We introduce Soft Clamp, a lightweight per-token divergence calibration method that preserves gradients while compressing extreme token-level signals. It provides strong 9B calibration with lower observed standard deviation across the three evaluated seeds, remains competitive at 4B, and requires no per-checkpoint inference-time bias selection.

The broader message is that multi-teacher OPD should be audited at behavior boundaries, not only at aggregate losses. When different teachers supervise different generation modes, local signals can bend global behavior. Several calibration strategies can mitigate the drift, but they trade off locality, recall, observed variability, and deployment cost.

2 Problem Setup

2.1 Multi-teacher on-policy distillation

Let x be a dialogue context and $y = (y_1, \ldots, y_T)$ a student-generated continuation. In on-policy distillation, the student samples from its current policy and then learns from teacher distributions evaluated on the sampled

trajectory. In our setting, each training example has a teacher tag $m \in \{\text{tool}, \text{response}\}$ that routes the example to a tool-call teacher or a response teacher.

We use GKD-style token losses based on the divergence between teacher and student next-token distributions [1]. Let $p_t(\cdot \mid x, y_{<i})$ be the teacher distribution and $p_s(\cdot \mid x, y_{<i})$ be the student distribution. In the teacher-API runs, p_t and p_s are renormalized on the teacher's top-32 returned tokens, so the reported token divergence is a top-k approximation of full-vocabulary JSD. The token loss is

$$d_i = \text{JSD}(p_t(\cdot \mid x, y_{<i}) \,\|\, p_s(\cdot \mid x, y_{<i})) \,. \tag{1}$$

The sequence loss averages token losses over supervised target positions. The main runs use a mixed rollout source: with probability $\lambda = 0.8$, the continuation comes from the current student; otherwise, the dataset trajectory is used. We add a supervised format anchor only on dataset-source trajectories:

$$\mathcal{L} = \mathcal{L}_{\text{distill}} + \alpha \, \mathbf{1}[z = \text{dataset}] \, \mathcal{L}_{\text{SFT}}, \tag{2}$$

where z is the trajectory source and $\alpha = 0.3$ in the main experiments. This anchor keeps structured tool-call targets in the expected textual protocol; it is held fixed across GKD variants and is not part of the proposed loss modifier.

2.2 Tool-call versus response supervision

The tool-call teacher and response teacher start from the same base model. The tool-call teacher is supervised on examples whose final target is a structured tool call. The response teacher is supervised on examples whose final target is a direct natural-language answer. Training uses the final assistant turn as the supervised target; previous turns serve as context.

For tool-call examples, our training implementation consumes the final assistant target as text. We therefore pre-render structured tool calls into the model chat template's textual tool-call form; Appendix A.2 gives an example. This is an implementation-specific serialization detail, not a requirement of multi-teacher OPD itself: other training frameworks may represent tool calls with different structured fields or templates. The detail matters for our experiments because treating the final target as a generic assistant response without the expected rendering can make the model learn an invalid schema even when decision metrics appear plausible.

2.3 Behavior leverage

We define *behavior leverage* as the degree to which a token position controls or constrains the future generation mode. Mode-entry tokens have high behavior leverage because changing them redirects the continuation. In tool-use generation, `<tool_call>` and function names expose this boundary. Later structural positions can also have leverage because they constrain whether the model follows a valid tool-call schema, continues with natural language, or enters further tool interactions.

This differs from ordinary token difficulty. A content token in a direct response can have high loss, but correcting it may only change a local phrase. A structural token with comparable loss can change the trajectory type or constrain the rest of the output. Multi-teacher OPD becomes vulnerable when one teacher's signals concentrate on high-leverage trajectory positions and another teacher's signals spread across lower-leverage content positions.

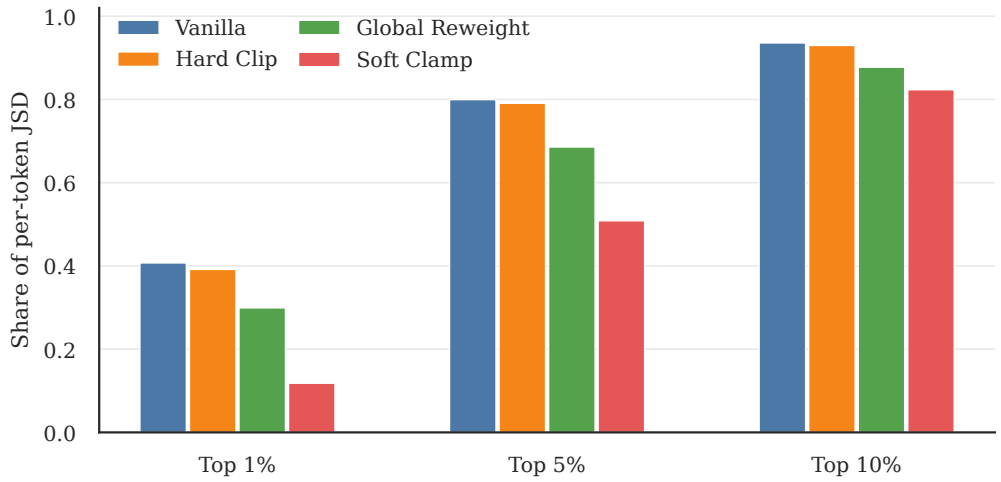

Figure 1: Token-level JSD concentration. In vanilla GKD, a small fraction of supervised tokens accounts for most of the token-level divergence. Soft Clamp substantially reduces this concentration, which is consistent with its goal of limiting extreme local signals rather than changing every token uniformly.

3 Diagnosing Tool-Call Boundary Drift

3.1 Aggregate statistics do not explain over-calling

Aggregate statistics rule out the simplest explanation for over-calling. In the representative diagnostic runs, response examples receive more token exposure, larger full-sequence per-token JSD, and comparable or larger squared-divergence proxy than tool-call examples. Appendix B.4 gives the full table and plot. The tool-call shift therefore cannot be explained by sample count, token count, or total divergence alone. This points to local signals whose behavioral effect is not proportional to their share of total loss.

Entropy and overlap do not fully explain the shift. The monitoring logs also rule out a simple account in which the tool-call teacher dominates only because it is more certain or better aligned with the student. The tool-call teacher is indeed lower-entropy in the late training window: its entropy is 0.079–0.084 across GKD variants, compared with 0.102–0.106 for the response teacher. However, this lower entropy does not translate into larger aggregate response-normalized divergence or squared-divergence proxy, as shown above. Response examples also have higher top-k overlap with the student (0.812–0.829 versus 0.793–0.808). Thus entropy and overlap statistics alone do not explain the observed tool-call shift. These monitored-run diagnostics point to signal location as a candidate explanation.

3.2 Extreme local signals are concentrated

We next ask whether a small number of tokens carry a disproportionate amount of the token-level divergence. Figure 1 visualizes a representative diagnostic run. For each diagnostic sample, we sort supervised tokens by per-token JSD and compute how much total JSD is contributed by the largest 1%, 5%, and 10% of tokens. Appendix B.6 reports the same diagnostic over all three seeds.

The concentration pattern is consistent across seeds. In vanilla GKD, the top 1% of tokens account for 41.2±0.4% of diagnostic-sample JSD on average, and the top 5% account for 76.8±2.9%. Hard Clip barely changes this concentration. Global Reweight reduces it to 32.7±2.4% and 67.2±1.5%, respectively, but does so through broad batch-relative scaling. Soft Clamp reduces the top-1% share to 15.4±3.2% and the top-5%

Method	APIGen over-call	$P(\text{tool})$ on response	Top-1 tool on response
Vanilla GKD	14.2±2.1	0.177±0.018	16.4±1.8
Hard Clip	11.9±0.9	0.169±0.020	15.8±2.0
Global Reweight	9.7±0.8	0.158±0.026	13.9±3.2
Soft Clamp	**9.0±0.2**	**0.147±0.014**	**12.7±1.4**

Table 1: Multi-seed decision pressure and final behavior. Values are mean±std over seeds 42, 44, and 60; values except probabilities are percentages.

share to 50.7±2.1%. This is consistent with the intended intervention of Soft Clamp: the failure mode is not only that the total signal is large, but that a small set of local positions can dominate the divergence mass. Because Soft Clamp directly compresses large token-level divergences, this concentration analysis should be read as a diagnostic of where the method acts, not as independent causal proof.

3.3 Decision pressure aligns with final behavior

We next inspect decision-level metrics on response examples. These metrics ask whether the student assigns high probability to <tool_call> even when the training example belongs to the response teacher. Table 1 reports the same process metric over three seeds and shows that response-side tool-call probability and response-side tool-call top-1 rate move with final APIGen over-calling. We use this alignment as diagnostic evidence: it identifies where the shift appears in the model distribution, but it does not by itself prove that a single token position caused the entire downstream behavior.

Table 1 averages the final logged training-step metrics over the three seeds. This evidence suggests that the model is not only improving on should-call examples. It is also moving the should-respond boundary toward the tool-call mode. Across the three paired seeds, Soft Clamp lowers both APIGen over-calling and response-side top-1 tool rate relative to vanilla GKD in every run. We use this as the operational signature behind the behavior-leverage interpretation: the learned distribution changes around a boundary decision that strongly constrains the subsequent generation mode. Appendix B.5 provides representative scatter and step-level process plots. Appendix C.2 further tests a simple inference-time alternative: applying a validation-tuned bias to the <tool_call> entry decision reproduces most of the mean first-token response-entry reduction and much of the strict-decoder multi-turn reduction in call frequency, but not the same joint call-recall, non-tool-final, and invalid-call profile across seeds.

4 Calibration Strategies

4.1 Training-time interventions

We compare three training-time interventions under the same teacher pair, data, OPD settings, and format anchor. Hard Clip is a fixed-threshold contrast: it truncates extreme token losses and therefore removes marginal gradient beyond the threshold. Global Reweight is a broad batch-relative contrast: it changes token weights according to their relative divergence. Soft Clamp is the localized alternative: it leaves ordinary tokens unchanged and compresses only extreme token-level divergences.

4.2 Soft Clamp: token-level dynamic compression

Soft Clamp targets extreme token-level divergence. For a batch of supervised tokens with divergences $\{d_i\}$, we define

$$C = \text{stopgrad}(k \cdot \text{mean}_i(d_i)) \,, \tag{3}$$

Strategy	Stage	Scope	Role in this study
Vanilla GKD	Training	All supervised tokens	Reveals the boundary-drift failure.
Hard Clip	Training	Fixed extreme tokens	Tests whether direct truncation is enough.
Global Reweight	Training	Broad batch-relative weights	Strong global calibration comparator.
Soft Clamp	Training	Batch-adaptive extreme tokens	Local training-time calibration with nonzero gradients.
Entry bias	Inference	First tool-entry logit	Validation-tuned counterfactual for boundary suppression.

Table 2: Calibration strategies compared in this paper. The goal is not to rank universal tool-use methods, but to expose how different interventions trade off locality, gradient preservation, validation requirements, and behavior calibration.

where k is a clamp multiplier. Detaching C matches the implementation: the threshold adapts to the current batch scale but does not backpropagate through the batch mean. The calibrated divergence is

$$d'_i = \begin{cases} d_i, & d_i \le C, \\ d_i \cdot \frac{C}{\mathrm{stopgrad}(d_i)}, & d_i > C. \end{cases} \tag{4}$$

The detached denominator makes the forward value equal to C for extreme tokens while scaling their gradients by C/d_i. Thus Soft Clamp compresses extremes but keeps a nonzero gradient.

Algorithm 1: Soft Clamp token calibration
Input: supervised token divergences $\{d_i\}$, clamp factor k
Compute: $C \leftarrow \mathrm{stopgrad}(k\,\mathrm{mean}_i(d_i))$
For each token i:
if $d_i \le C$, set $d'_i \leftarrow d_i$
else set $d'_i \leftarrow d_i\, C/\mathrm{stopgrad}(d_i)$
Return: calibrated distillation loss $\mathrm{mean}_i(d'_i)$, combined with the shared SFT format anchor in the main runs.

The forward cap and the nonzero scaled gradient distinguish Soft Clamp from hard clipping, while the batch-adaptive detached threshold avoids choosing one fixed threshold for all training stages.

4.3 Comparison with hard clipping and global reweighting

Hard clipping tests whether directly truncating extreme token losses is enough. It is simple, but it can remove much of the learning signal from difficult high-leverage tokens. Global reweighting changes token weights according to their batch-relative divergence. It is useful as a contrast, but it scales broad regions of the loss rather than isolating local extremes.

Soft Clamp is deliberately narrower. It leaves ordinary tokens unchanged, preserves gradients for extreme tokens, and adapts the threshold to the current batch scale. It does not explicitly identify mode-entry tokens; instead, it limits the effect of extreme local divergences that concentrate token-level signal. It does not require a new teacher routing policy, a learned reward model, or a held-out calibration set.

Global Reweight is the strongest competing loss modifier in our experiments, so the distinction matters. It adapts the global-normalization idea behind GNDPO [4] to our JSD loss setting: tokens receive detached batch-relative weights rather than a fixed clamp. It can also reduce over-calling, but it changes all tokens through a normalized batch-relative weight. Soft Clamp instead leaves most tokens exactly unchanged and

only compresses tokens whose divergence exceeds a batch-adaptive extreme threshold. This local intervention matches the sparse high-divergence trajectory diagnostics in Figure 1 and gives lower observed APIGen-MT over-calling standard deviation across the three evaluated seeds. In the BFCL multi-turn diagnostic, Global Reweight is close in mean performance, and it slightly leads the 4B multi-turn means. Soft Clamp instead exhibits lower variance on the 9B loop-related metrics in our three-seed evaluation while using a more localized intervention. Appendix B.5 reports the representative intervention-strength diagnostic.

4.4 Format anchoring

Structured tool calls introduce a second risk: schema drift. A student can learn to enter the tool-call mode while drifting away from the required textual protocol for executable calls. Our main runs therefore include the dataset-source supervised anchor defined in Section 2. This anchor is not the proposed method; it is a controlled training component that substantially improves format stability while we compare GKD variants. Appendix K stress-tests a pure teacher-distillation OPD configuration without this anchor and with a fully student-generated rollout mixture.

5 Experiments

5.1 Models and training

The main experiments use the Qwen3.5-9B checkpoint (Qwen/Qwen3.5-9B) from the public Hugging Face model card [15]. The tool-call teacher and response teacher are initialized from the same checkpoint and trained on their respective target types. They serve as frozen online supervisors, not as deployable balanced baselines: "online" means that GKD queries them on student rollouts, not that teacher parameters are updated. The main comparison therefore reports the base model, a base SFT model trained on the mixed tool-use data, and four GKD students: vanilla GKD, Hard Clip, Global Reweight, and Soft Clamp. The GKD students share the same data, teachers, training length, and core OPD settings. The main GKD runs use $sft\_alpha = 0.3$ as a format anchor. Section 5.7 reports a smaller-student replication that replaces the Qwen3.5-9B student with a Qwen3.5-4B student while keeping the 9B teacher pair, data, and training recipe fixed.

For the four GKD variants, the main single-turn results are averaged over three random seeds. We report mean and standard deviation. The base and base SFT rows are single reference runs. Unless otherwise stated, the main benchmark tables report mean$\pm$std over three seeds for GKD variants. Process diagnostics, targeted-clamp controls, factor sensitivity, and pure-OPD format-stability checks use representative diagnostic runs or single-seed ablations.

The format anchor is held fixed across the GKD variants, so it cannot explain the differences among vanilla GKD, Hard Clip, Global Reweight, and Soft Clamp. Appendix K separately reports a pure teacher-distillation OPD stress test that removes this anchor and uses fully student-generated rollouts; that tested configuration is unstable for the tool-call schema.

We also run targeted-clamp controls to test whether the first few supervised tokens alone are sufficient to explain the benefit of Soft Clamp. Appendix E shows that clamping only the first four supervised tokens does not reproduce the full Soft Clamp result. This argues against a first-token-only account and supports a broader diagnostic interpretation: the compressed mismatches are local and sparse, but can occur beyond the initial target positions. Appendix F.3 further checks the clamp factor used by Soft Clamp and shows that factors 2, 3, and 4 all reduce over-calling relative to vanilla GKD with similar decision accuracy.

Method	Decision Acc	Over-calling	Call Recall	Respond Recall
Base	80.7	7.2	68.5	92.8
Base SFT	85.3	4.9	75.5	95.1
Vanilla GKD	88.6±0.3	14.2±2.1	**91.5±1.7**	85.8±2.1
Hard Clip	**88.7±1.0**	11.9±0.9	89.2±3.0	88.2±0.9
Global Reweight	88.6±1.3	9.7±0.8	86.9±3.3	90.3±0.8
Soft Clamp	**88.7±0.7**	**9.0±0.2**	86.5±1.4	**91.0±0.2**

Table 3: APIGen-MT-derived decision results. All values are percentages. GKD rows report mean±std over three seeds; Base and Base SFT are single reference runs. Bold marks the best displayed GKD value in each column, including displayed ties. Vanilla GKD improves call recall but raises over-calling. Soft Clamp gives the lowest mean over-calling and lowest observed over-calling standard deviation among GKD variants, with a call-recall trade-off relative to vanilla GKD.

5.2 Datasets and metrics

APIGen-MT-derived decision set. APIGen-MT [13] is the source of our main in-domain tool-use decision set. We construct a conversation-disjoint marker-level decision evaluation from held-out APIGen-MT conversations and report decision accuracy, over-calling rate, should-call recall, and should-respond recall. After removing thinking blocks, a generation is counted as a tool-call decision if it contains the `<tool_call>` marker. This metric does not require parser-valid XML, executable arguments, or an AST-correct function call; schema quality is analyzed separately. The decision test used here is balanced between should-call and should-respond examples, so decision accuracy is the average of the two recalls and over-calling is one minus should-respond recall. We report all four quantities because they expose the trade-off between gaining call recall and losing response recall.

BFCL. BFCL [12] evaluates function-calling quality and irrelevance refusal. We use it to test whether tool-call calibration transfers beyond APIGen-style decision points.

When2Call. When2Call [16] evaluates whether a model should call a tool, request more information, or refuse. We use the multiple-choice formulation as an out-of-domain decision diagnostic, not as the main success metric.

BFCL multi-turn loop diagnostic. We also run a multi-turn diagnostic on 800 BFCL tasks and 3136 user turns. It records tool calls per turn, loop rates, max-step hits, repeated same-call rates, and non-tool-final rates. This diagnostic is not an official BFCL leaderboard protocol. It measures whether the single-turn over-calling tendency becomes a practical multi-turn usability problem under a fixed simulated tool environment.

5.3 APIGen-MT main results

The main result should be read as a behavior-calibration comparison within the GKD family. We do not use Soft Clamp to replace supervised tool-use learning or to claim a new overall tool-use state of the art. Instead, we ask whether a GKD student can retain much of the call-recall gain over SFT while reducing the over-calling side effect introduced by vanilla GKD.

Table 3 shows the main operating-point trade-off. Base SFT favors response behavior: it has low over-calling and high respond recall, but lower call recall. Vanilla GKD moves the model in the opposite direction: call recall reaches 91.5% on average, but over-calling rises to 14.2%. Soft Clamp reduces over-calling to 9.0% with lower observed three-seed standard deviation and comparable decision accuracy, but its call recall

drops to 86.5%. This trade-off is desirable when unnecessary calls and multi-turn loops are costly; it is not a claim that lower tool-call tendency is universally better.

The paired-seed comparison gives the same picture. Across seeds 42, 44, and 60, Soft Clamp lowers over-calling relative to vanilla GKD in all three runs, by 4.7, 7.3, and 3.6 percentage points. Global Reweight is the closest competitor: Soft Clamp has lower mean over-calling and lower variance in our three-seed evaluation, and it is lower than Global Reweight in two of the three paired seeds. In the remaining seed, Global Reweight is lower by only 0.15 percentage points. We therefore treat Soft Clamp as a more localized intervention that exhibits lower variance in this three-seed evaluation, not as a method with a large absolute margin over Global Reweight on APIGen-MT. Appendix C.1 includes the APIGen-MT trade-off plot and qualitative should-respond examples.

5.4 Checkpoint-level behavior trajectory

Final checkpoints alone do not show whether the tool-call bias appears only at the end of training or persists through the training trajectory. We therefore evaluate intermediate checkpoints for three seeds of vanilla GKD and Soft Clamp. Figure 2 focuses on the post-transient region from checkpoint 150 onward. In this region, vanilla GKD has higher mean over-calling than Soft Clamp at every evaluated checkpoint; from checkpoint 200 onward, the ordering also holds for all three paired seeds. Its final over-calling is 14.2±2.1%, whereas Soft Clamp ends at 9.0±0.2%. The recall gap shows the same pattern. This is consistent with the narrower claim that behavior leverage imbalance is associated with a persistent observed tool-call bias during OPD, and that Soft Clamp reduces that bias without collapsing the model's ability to call tools.

We also observe seed-dependent early transients. Appendix J reports the full-range trajectory. These spikes are real generation behavior rather than evaluation parser artifacts, but their timing and magnitude vary by seed. We therefore use the post-transient trajectory as the main training-process evidence.

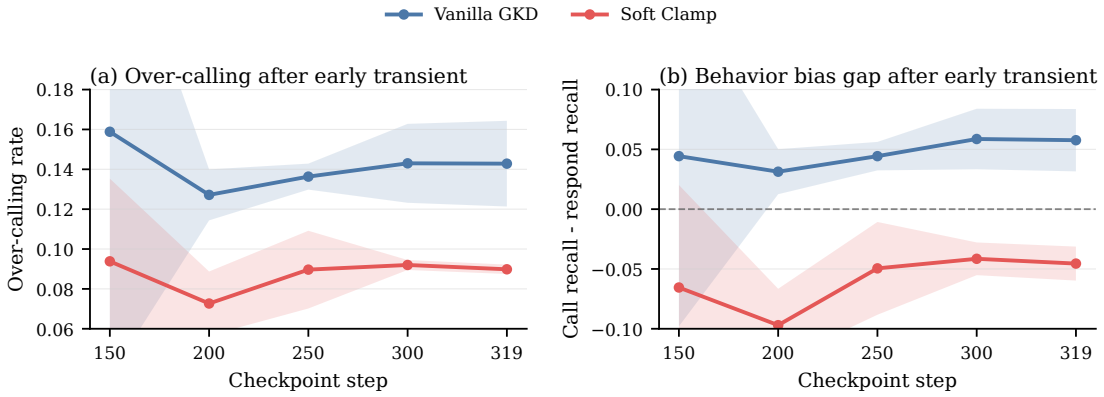

Figure 2: APIGen-MT checkpoint-level behavior trajectory after the early transient phase. Lines show means over three seeds and bands show standard deviations. We evaluate checkpoints 150, 200, 250, 300, and 319 on the same APIGen-MT test set. The upper panel reports over-calling, and the lower panel reports the recall gap between should-call and should-respond examples.

5.5 BFCL and When2Call

BFCL is consistent with the calibration interpretation but also limits the claim. Soft Clamp gives the best mean overall BFCL score among GKD variants and remains competitive on irrelevance refusal, but Global Reweight and Hard Clip have higher irrelevance-refusal means. Thus BFCL does not isolate Soft Clamp

Method	Calls/turn	Loop@3	Loop@5	Max-step	Repeat call	Non-tool final
Base SFT	0.974	5.1	0.7	0.7	2.5	96.5
Vanilla GKD	1.515±0.122	15.1±3.0	8.8±3.5	8.8±3.5	17.7±6.4	87.3±7.2
Hard Clip	1.348	11.5	6.3	6.3	14.4	91.5
Global Reweight	1.297±0.088	10.8±1.4	5.5±1.3	5.5±1.3	13.1±2.7	93.6±1.6
Soft Clamp	**1.289±0.023**	**10.3±0.4**	**5.0±0.3**	**5.0±0.3**	**11.3±0.6**	**93.9±0.2**

Table 4: BFCL multi-turn loop diagnostic. All values except calls/turn are percentages. "Non-tool final" means that the model eventually emits a non-tool response; it does not measure answer correctness. Vanilla GKD, Global Reweight, and Soft Clamp report mean±std over three seeds; Hard Clip and Base SFT are single diagnostic runs. Bold marks the best GKD value in each column. In this harness the maximum interaction depth is five tool-call steps, so Loop@5 and Max-step coincide. Soft Clamp has the lowest mean and lower observed standard deviation on the main loop/call metrics among the three-seed GKD variants, but Global Reweight is close.

as uniquely best across all submetrics, and none of the GKD variants improves BFCL overall beyond the base SFT. When2Call is more restrictive: GKD variants retain high tool-call accuracy but perform worse on request-for-information and cannot-answer cases. These results frame Soft Clamp as a GKD-internal behavior calibration method, not a universal tool-use decision solution.

This framing is important for interpreting negative transfer. A lower over-calling rate is valuable when a GKD run has already moved the model toward excessive tool use, but it does not imply that the resulting model should beat a well-anchored SFT baseline on every out-of-domain decision benchmark. The method addresses a specific training-induced failure mode. Appendix C.3 gives the full BFCL and When2Call tables.

5.6 Multi-turn tool-call loops

The multi-turn diagnostic connects the decision metric to deployment behavior. Across three seeds, vanilla GKD makes 1.515 tool calls per turn on average, triggers Loop@3 on 15.1% of turns, and repeats the same call on 17.7% of turns. Soft Clamp lowers these values to 1.289, 10.3%, and 11.3%, respectively. The non-tool final rate rises from 87.3% to 93.9%. This result does not replace official BFCL scoring, but it captures the practical risk of an excessive tool-call prior under a controlled interaction harness. Global Reweight is a strong comparator in this diagnostic: its mean calls/turn, Loop@3, and non-tool final rate are close to Soft Clamp. The difference is that Soft Clamp keeps slightly lower means and smaller observed standard deviations on the loop and repeated-call metrics, while intervening only on extreme local divergences rather than reweighting all tokens. Appendix H provides the corresponding diagnostic plot.

5.7 Smaller-student replication

To test whether the behavior-calibration pattern is specific to the 9B student, we repeat vanilla GKD, Global Reweight, and Soft Clamp with a Qwen3.5-4B student over the same three seeds, while keeping the 9B teacher pair, data, OPD settings, and evaluation harness fixed. Table 5 reports the key APIGen-MT and BFCL multi-turn metrics, with Base 4B and Base SFT 4B included as reference points. The 4B run shows the same qualitative failure mode as the main 9B experiments: vanilla GKD raises call recall but also has higher over-calling and multi-turn loop risk than the calibrated GKD variants. Soft Clamp lowers APIGen-MT over-calling from 12.8±0.7% to 9.2±1.2%, while Global Reweight obtains 9.5±0.8%. In multi-turn evaluation, both calibrated variants reduce calls per turn, Loop@3, and repeat calls relative to vanilla; Global Reweight is slightly stronger on the 4B multi-turn means, while Soft Clamp remains competitive and applies the more localized intervention.

Method	Dec. Acc	Over-call	Call Rec.	Resp. Rec.	Calls/turn	Loop@3	Non-tool final
Base 4B	79.6	16.9	76.0	83.2	1.317	12.3	95.2
Base SFT 4B	88.5	7.5	84.5	92.5	0.999	5.7	97.4
Vanilla GKD 4B	**88.4±0.8**	12.8±0.7	**89.6±2.1**	87.2±0.7	1.493±0.122	15.7±3.2	88.2±3.9
Global Reweight 4B	87.8±0.8	9.5±0.8	85.1±1.9	90.5±0.8	**1.200±0.034**	**8.9±1.4**	**95.4±1.7**
Soft Clamp 4B	87.6±0.6	**9.2±1.2**	84.4±1.8	**90.8±1.2**	1.235±0.097	9.2±2.6	94.8±2.1

Table 5: Qwen3.5-4B smaller-student replication. APIGen-MT values and multi-turn values except calls/turn are percentages. GKD rows report mean±std over three seeds; Base 4B and Base SFT 4B are single reference runs. Bold marks the best displayed GKD value in each column.

This replication does not prove broad cross-scale generality, but it removes a basic single-model concern. It also sharpens the trade-off: calibrated GKD reduces excessive tool use and loop risk, but vanilla keeps the highest call recall, and Base SFT remains a strong reference for direct decision calibration. As in the 9B setting, the improvement is concentrated on over-calling calibration and multi-turn loop behavior; broader BFCL and When2Call results remain mixed and do not indicate a uniform capability gain. Appendix I reports the full 4B BFCL, When2Call, multi-turn, and response-side decision-pressure diagnostics.

6 Related Work and Limitations

On-policy distillation and stabilization. On-policy distillation trains the student on its own generated distribution, and GKD instantiates this idea with divergence-based teacher supervision [1]. MiniLLM and related work study reverse-KL-style distillation and student-generated samples [10, 21]. Other OPD work studies self-distillation, entropy-aware stabilization, and global normalization [4, 7, 24]. These methods improve or stabilize OPD signals. We study a different failure mode: how fixed routed teacher signals move a tool-call versus response boundary.

Multi-teacher capability integration. Multi-teacher OPD introduces teacher routing, teacher specialization, and capability integration questions [11, 17, 18]. These works mainly ask how to combine or select teachers. We ask what happens after routing is fixed: where do teacher signals bend the student's behavior boundary?

Balancing losses, gradients, and tokens. Multi-objective training often uses loss weighting or gradient balancing to manage competing objectives [2, 9]. Distillation methods also reweight tokens or teacher contributions [8, 19, 20]. GNDPO uses global normalization to stabilize on-policy distillation signals [4]; our Global Reweight baseline adapts this idea to token-level JSD. Soft Clamp is not a global balancing rule. It leaves ordinary tokens unchanged and compresses only extreme token-level divergences, targeting sparse local domination rather than average teacher weight.

Tool-use behavior drift. Recent work reports tool overuse, tool-use collapse, and repeated calls in RL or agentic search settings [3, 5, 6, 14, 22, 23]. Our result shows that a related drift can arise earlier, inside supervised multi-teacher OPD. The failure is not only that a model calls too many tools; it is that distillation can move the response/tool-call boundary even when aggregate losses look balanced.

Limitations. Our study has five main limits. First, the main study uses Qwen3.5-9B and one primary two-teacher tool-call versus response setting. The Qwen3.5-4B replication in Appendix I supports the same

qualitative Soft-Clamp-versus-vanilla pattern across two student sizes within one model family under fixed 9B teachers, but broader validation across scales, teacher pairs, and heterogeneous skills remains future work. Second, our mechanism evidence is diagnostic, not a full causal proof: multi-seed decision pressure and token-concentration metrics identify where the shift appears, but they do not isolate every token-level cause. The E1 counterfactual in Appendix C.2 further shows that a validation-tuned inference-time entry bias nearly reproduces the loop-control effect under a strict decoder, while leaving residual gaps in non-tool-final and invalid-call behavior; this supports the boundary-calibration account without proving a separate trajectory-level mechanism. Third, the BFCL multi-turn loop diagnostic measures usability risk under a fixed simulated environment and should complement, not replace, official benchmark scores; its non-tool-final metric is a termination metric, not task-success accuracy. Fourth, the main GKD variants share a supervised format anchor, while the pure OPD comparison removes that anchor and changes the rollout mixture; we therefore treat pure OPD as a schema-stability stress test rather than an anchor-only causal ablation. Fifth, our teacher-API divergence uses the teacher's top-32 returned tokens and renormalizes teacher and student probabilities on that truncated support. This makes the logged JSD a support-truncated approximation rather than a full-vocabulary divergence.

Artifact availability. We plan to release the training code, analysis scripts, and evaluation harness after internal review. The released artifacts will include the logging metrics and plotting scripts needed to reproduce the main diagnostic figures.

7 Conclusion

Multi-teacher OPD can shift model behavior even when aggregate teacher contributions look balanced. In tool-use distillation, vanilla GKD can improve tool-call recall while also increasing over-calling. Our diagnostics show that this shift is better reflected by response-side decision pressure than by aggregate token exposure or full-sequence divergence.

The right response is not a single universal fix. Hard clipping, Global Reweight, Soft Clamp, and inference-time entry bias all mitigate parts of the drift, but they trade off gradient preservation, locality, observed seed variability, call recall, and deployment requirements. Soft Clamp offers a simple localized training-time option. By dynamically compressing extreme per-token divergence and preserving gradients, it reduces over-calling and multi-turn loop risk among GKD variants while maintaining APIGen decision accuracy, at the cost of lower call recall than vanilla GKD. Targeted-clamp and token-position diagnostics argue against an entry-token-only account; much compression occurs where student trajectories diverge from teacher expectations. The broader lesson is diagnostic: multi-teacher OPD should track not only how much signal each teacher supplies, but also where that signal acts and whether it pushes behavior boundaries.

AI assistance. The authors used AI-assisted tools for language polishing, draft organization, LaTeX editing, and code/documentation assistance. All scientific claims, equations, experimental results, citations, and final text were reviewed and verified by the authors, who take full responsibility for the manuscript.

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations (ICLR)*, 2024. URL `https://arxiv.org/abs/2306.13649`.

[2] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning (ICML)*, 2018.

[3] Wenlong Deng, Yushu Li, Boying Gong, Yi Ren, Christos Thrampoulidis, and Xiaoxiao Li. On grpo collapse in search-r1: The lazy likelihood-displacement death spiral. *arXiv preprint arXiv:2512.04220*, 2025. URL https://arxiv.org/abs/2512.04220.

[4] D. Hao, Zhiwei Jin, Chen Chen, and H. Lu. Stabilizing on-policy distillation for mllm reasoning with global normalization. *arXiv preprint arXiv:2606.09091*, 2026. URL https://arxiv.org/abs/2606.09091.

[5] Yupu Hao, Zhuoran Jin, Huanxuan Liao, Kang Liu, and Jun Zhao. Why multi-step tool-use reinforcement learning collapses and how supervisory signals fix it. *arXiv preprint arXiv:2606.26027*, 2026. URL https://arxiv.org/abs/2606.26027.

[6] Bowen Jin, Hansi Zeng, Zhenrui Yue, Wang Dong, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025. URL https://arxiv.org/abs/2503.09516.

[7] Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2603.07079.

[8] Seongryong Jung et al. Todi: Token-wise distillation via fine-grained divergence control. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. URL https://arxiv.org/abs/2505.16297.

[9] Alex Kendall, Roberto Cipolla, and Yarin Gal. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] Jongwoo Ko et al. Distillm-2: A contrastive approach boosts the distillation of llms. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2503.07067.

[11] Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026. URL https://arxiv.org/abs/2606.30406.

[12] Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *International Conference on Machine Learning (ICML)*, 2025. URL https://gorilla.cs.berkeley.edu/leaderboard.html.

[13] Akshara Prabhakar, Zuxin Liu, Weiran Yao, Jianguo Zhang, Ming Zhu, Shiyu Wang, Zhiwei Liu, Tulika Awalgaonkar, Haolin Chen, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, Silvio Savarese, and Caiming Xiong. APIGen-MT: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. *arXiv preprint arXiv:2504.03601*, 2025. URL https://arxiv.org/abs/2504.03601.

[14] Cheng Qian, Emre Can Acikgoz, Hongru Wang, Xiusi Chen, Avirup Sil, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. Smart: Self-aware agent for tool overuse mitigation. In *Findings of the Association for Computational Linguistics: ACL*, 2025. URL https://aclanthology.org/2025.findings-acl.239/.

[15] Qwen Team. Qwen3.5-9B. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-9B.

[16] Hayley Ross, Ameya Sunil Mahabaleshwarkar, and Yoshi Suhara. 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*, 2025. URL https://github.com/NVIDIA/When2Call.

[17] Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026. URL https://arxiv.org/abs/2604.00626.

[18] Jianze Wang, Ying Liu, Jinlong Chen, Xuchun Hu, Qilong Zhang, Yu Cao, Jun Wang, Hua Yang, Yong Xie, and Qianglong Chen. Mad-opd: Breaking the ceiling in on-policy distillation via multi-agent debate. *arXiv preprint arXiv:2605.01347*, 2026. URL https://arxiv.org/abs/2605.01347.

[19] Xurong Xie, Zhucun Xue, Jiafu Wu, Jian Li, Yabiao Wang, Xiaobin Hu, Yong Liu, and Jiangning Zhang. Llm-oriented token-adaptive knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/40701.

[20] Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Tip: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026. URL https://arxiv.org/abs/2604.14084.

[21] Yuxian Xu et al. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2306.08543.

[22] Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *arXiv preprint arXiv:2509.02479*, 2025. URL https://arxiv.org/abs/2509.02479.

[23] Yirong Zeng, Shen You, Yufei Liu, Qunyao Du, Xiao Ding, Yutai Hou, Yuxian Wang, Wu Ning, Haonan Song, Dandan Tu, Bibo Cai, and Ting Liu. The tool-overuse illusion: Why does llm prefer external tools over internal knowledge? *arXiv preprint arXiv:2604.19749*, 2026. URL https://arxiv.org/abs/2604.19749.

[24] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026. URL https://arxiv.org/abs/2601.18734.

A Additional Experimental Details

A.1 Training data and supervision

The main GKD experiments use two training splits derived from APIGen-MT: 15,419 tool-call examples and 15,420 response examples. We first split 5,000 APIGen-MT conversations at the conversation level with split seed 42: 3,500 conversations for training, 500 for validation, and 1,000 for test. The conversation IDs are disjoint across splits. Training examples are balanced by target type; the validation split is approximately balanced, with 2,188 tool-call examples and 2,172 response examples. The APIGen-MT-derived decision test is extracted only from the held-out test conversations, with four balanced decision buckets of 1,000 examples each, for 4,000 total test decision points. Each GKD training example is routed to one teacher by its teacher tag. The tool-call teacher supervises examples whose final target is a structured tool call, and the response teacher supervises examples whose final target is a natural-language answer.

Only the final assistant turn is used as the supervised target. Earlier turns provide dialogue context. In our ms-swift/Megatron implementation, the training backend treats this final assistant target as text. For tool-call examples, we therefore pre-render the final target into the model chat template's XML-style textual form, with tags such as `<tool_call>`, `<function=...>`, and `<parameter=...>`. This preprocessing is backend-specific rather than an algorithmic requirement. It is important in our setup because leaving a tool call as an unrendered object can make the model learn an invalid tool-call schema.

A.2 Tool-call target rendering

For structured tool-call examples, the final supervised target is rendered as text before training in our backend. A representative target is:

```
<tool_call>
<function=get_flight_cost>
<parameter=travel_to>
LAX
</parameter>
</function>
</tool_call>
```

This appendix example is illustrative; the experiments use the rendered targets from the APIGen-MT-derived training split. Other frameworks may serialize the same structured call differently.

A.3 Main training configuration

Table 6 lists the shared training configuration for the four GKD variants. All variants use the same data, teachers, model initialization, training length, and batch settings. They differ only in the loss modifier: none for vanilla GKD, hard clipping for Hard Clip, batch-relative reweighting for Global Reweight, and dynamic token compression for Soft Clamp.

The supervised anchor is not part of the proposed method. It is held fixed across all GKD variants to improve structured-output stability while comparing the distillation losses. In the main runs, student rollouts are sampled with probability $\lambda = 0.8$ and dataset trajectories are used otherwise. The SFT anchor is applied only on dataset-source trajectories, as defined in Section 2.

Teacher and SFT reference training. The 9B tool-call teacher, response teacher, and Base SFT reference are full fine-tunes from the Qwen3.5-9B checkpoint. They use the same ms-swift SFT backend, bfloat16 precision, DeepSpeed ZeRO-3, Qwen3.5 template, maximum length 10,000, learning rate 5×10^{-6}, cosine schedule with 0.1 warmup fraction, micro batch size 4, gradient accumulation 2, and two epochs. The

Item	Value
Initialization checkpoint	Qwen/Qwen3.5-9B
Training backend	ms-swift Megatron GKD backend
Precision	bfloat16
Epochs	1
Micro batch size	4
Global batch size	96
Maximum input length	14,000 tokens
Maximum completion length	512 tokens
Learning rate	5×10^{-6}
Warmup fraction	0.1
Tensor parallel size	8
GKD rollout mixture λ	0.8
GKD β	0.5
Sequence KD	false
Supervised format anchor	`sft_alpha`=0.3
Teacher logit top-k	32
Loss scale	`last_response+ignore_empty_think`
Checkpoint interval	50 steps

Table 6: Shared GKD training configuration.

tool-call teacher is trained only on the tool-call SFT split, the response teacher only on the response SFT split, and Base SFT on the mixed SFT split. The 4B Base SFT reference uses the same recipe with Qwen3.5-4B. We select the epoch-2 checkpoints used throughout the evaluation.

Loss-modifier hyperparameters. The main loss-modifier settings are fixed before the reported multi-seed test evaluation: Soft Clamp uses clamp factor $k = 3$, Hard Clip uses threshold $c = 0.5$, and Global Reweight uses $\alpha = 0.3$, $z_{\max} = 3.0$, $w_{\min} = 0.25$, and $w_{\max} = 2.0$. These values were chosen as simple engineering defaults after inspecting early diagnostic runs and training-log scale, not by optimizing the final multi-seed test metrics. The small single-seed Soft Clamp factor sweep in Appendix F.3 is therefore an exploratory robustness check, not a test-independent model-selection protocol. We qualify method comparisons as fixed-setting comparisons rather than matched-strength Pareto comparisons.

B Metric Definitions

B.1 Evaluation metrics

For the APIGen-MT-derived decision set, *decision accuracy* is the fraction of examples for which the model makes the correct tool-use mode-entry decision. *Over-calling* is the fraction of should-respond examples whose generated response contains the `<tool_call>` marker after removing thinking blocks. *Call recall* is marker-level recall on should-call examples, and *respond recall* is recall on should-respond examples. These metrics do not require parser-valid XML or executable arguments; parser-valid schema behavior is analyzed separately. The APIGen-MT-derived decision test used in this paper is balanced, so decision accuracy is the average of call recall and respond recall, while over-calling equals one minus respond recall. We retain all columns to make the call/response operating-point trade-off explicit. In this paper, boundary calibration means operating-point calibration between tool-call and direct-response decisions; it is not probability calibration measured by ECE, Brier score, or reliability diagrams.

Decoding settings. Main single-turn APIGen-MT, BFCL, and When2Call inference uses a vLLM-backed OpenAI-compatible server with thinking disabled, greedy decoding (`temperature`=0, `top_p`=1), and dataset-specific generation lengths: 512 new tokens for APIGen-MT, 256 for BFCL, 16 for When2Call multiple choice, and 512 for When2Call judge-style outputs. The BFCL multi-turn diagnostic also uses greedy decoding, a maximum of five tool-call steps per user turn, and a fixed simulated tool-observation harness. The strict E1 counterfactual in Appendix C.2 instead uses a local Transformers decoder so that a first-token-only bias can be applied; that bias is applied at the first generated assistant token of every assistant generation step.

For BFCL, we use local BFCL v4-style processed data from seven single-turn subsets: simple-python, multiple, parallel, live-simple, live-multiple, irrelevance, and live-irrelevance. The processed files contain 400, 200, 200, 258, 1053, 240, and 884 examples, respectively. Our project-local scorer checks exact AST-style matches for tool-call subsets, allowing any acceptable BFCL ground-truth value, and counts irrelevance examples as correct when no tool call is parsed. These are local BFCL-v4-style scores rather than official leaderboard submissions.

For When2Call, we use the public processed MCQ test snapshot with 3,652 examples. The four choices map A to direct response, B to tool call, C to request for information, and D to cannot answer; the scorer extracts the final standalone A–D answer letter. The raw 300-example judge-style file is preprocessed with a 512-token generation length for completeness, but the reported tables use only the MCQ protocol.

B.2 Training process metrics

The training logs contain both raw and adjusted statistics. Raw statistics are computed before any loss modifier. Adjusted statistics are computed after Soft Clamp, Hard Clip, or Global Reweighting. In vanilla GKD, raw and adjusted values are identical.

The main aggregate diagnostic is the raw per-token top-k JSD ratio:

$$R_{\mathrm{JSD}} = \frac{\sum_{i \in \mathrm{tool}} d_i / N_{\mathrm{tool}}}{\sum_{i \in \mathrm{resp}} d_i / N_{\mathrm{resp}}}, \tag{5}$$

where d_i is the token-level top-k divergence and N is the number of supervised loss tokens. In teacher-API runs, teacher log probabilities are returned for the top 32 teacher tokens; student logits are gathered on those token IDs, and both distributions are renormalized on that truncated support before computing JSD. We also use a squared-divergence proxy ratio:

$$R_{\mathrm{grad}} = \frac{\sum_{i \in \mathrm{tool}} d_i^2 / N_{\mathrm{tool}}}{\sum_{i \in \mathrm{resp}} d_i^2 / N_{\mathrm{resp}}}. \tag{6}$$

This proxy is not a parameter-gradient norm; it is a scale-sensitive diagnostic for whether a small number of large token divergences dominate the divergence mass. These ratios remove the first-order effect of response length and batch composition. Values below one mean that response examples have larger average full-sequence divergence or squared-divergence proxy.

Decision-boundary metrics inspect the first supervised token. We log the student's probability of `<tool_call>`, the log-odds margin between `<tool_call>` and all non-tool-call tokens, the margin between `<tool_call>` and common natural-language response starters, and whether `<tool_call>` is the student's top-1 token. These metrics connect training dynamics to final over-calling because the first supervised token determines whether the model enters the tool-call mode. They should be interpreted as boundary diagnostics, not as a complete causal decomposition of all later generation behavior.

Signed pressure measures whether the teacher pushes the student toward `<tool_call>` at the decision boundary:

$$P_{\mathrm{signed}} = p_t(\texttt{<tool\_call>} \mid x, y_{<1}) - p_s(\texttt{<tool\_call>} \mid x, y_{<1}). \tag{7}$$

Positive values mean that the teacher assigns more probability to <tool_call> than the student does. Negative values mean that the teacher pushes against tool-call entry. Cumulative exposure metrics record the number of tool-call and response samples and loss tokens consumed up to each step, which helps separate training-order effects from teacher-signal effects.

B.3 Failure-mode taxonomy

Table 7 summarizes the failure modes separated by our diagnostics. This taxonomy is useful because a single aggregate loss curve can hide several distinct problems: the model can choose the wrong behavior mode, loop after a tool observation, or emit a malformed structured call.

Failure mode	Observable symptom	Main diagnostic signal	Mitigation
Over-calling	Tool calls on should-respond examples	Response-side $P(\texttt{<tool\_call>})$, response-side top-1 <tool_call> rate, APIGen over-calling	Boundary calibration: Hard Clip, Global Reweight, Soft Clamp, entry bias
Tool-call loop	Repeated calls or max-step hits in multi-turn interaction	Calls/turn, Loop@3, Loop@5/Max-step, repeat-same-call rate	Boundary calibration; Global Reweight, Soft Clamp, and entry bias are strongest in our tested settings
Schema drift	Invalid or unparseable XML-style tool-call strings	Parser validity, AST match, closed function-call blocks	SFT anchor
Aggregate-metric mismatch	Total exposure or full-sequence JSD conflicts with final behavior	Cumulative token ratios, raw per-token JSD ratios, decision-boundary metrics	Decision-level monitoring

Table 7: Failure modes and diagnostics. The rows separate behavior selection, multi-turn usability, structured-output validity, and misleading aggregate metrics.

B.4 Aggregate sanity diagnostics

Table 8 and Figure 3 give the representative aggregate exposure and divergence diagnostics summarized in the main text. T/R denotes tool-call divided by response. Values below one mean that tool-call examples do not dominate that statistic in the monitored run. These diagnostics are used as sanity checks rather than as a full multi-seed causal decomposition.

Method	Cum token T/R	Raw per-token JSD T/R	Raw squared-div. T/R
Vanilla GKD	0.867	0.881	0.957
Hard Clip	0.870	0.893	0.980
Global Reweight	0.874	0.902	0.985
Soft Clamp	0.881	0.895	0.976

Table 8: Aggregate sanity checks from representative diagnostic runs. The over-calling shift is not explained in these monitored runs by more tool-call token exposure, larger full-sequence per-token divergence, or a larger squared-divergence proxy.

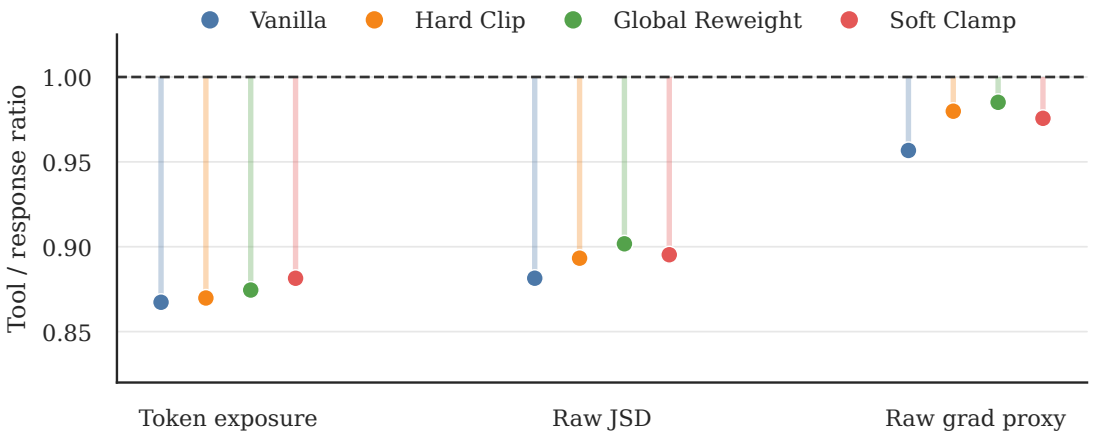

Figure 3: Token exposure and aggregate JSD sanity checks. Tool-call signals do not dominate in aggregate in the monitored run, so the evidence points to signal location as a candidate explanation rather than only how much signal each teacher contributes.

B.5 Additional process diagnostics

Figure 4 and Figure 5 show representative process diagnostics behind Table 1. Figure 6 reports a representative intervention-strength diagnostic. These plots are useful for auditing training dynamics, but the main paper relies on multi-seed summary tables for its central claims.

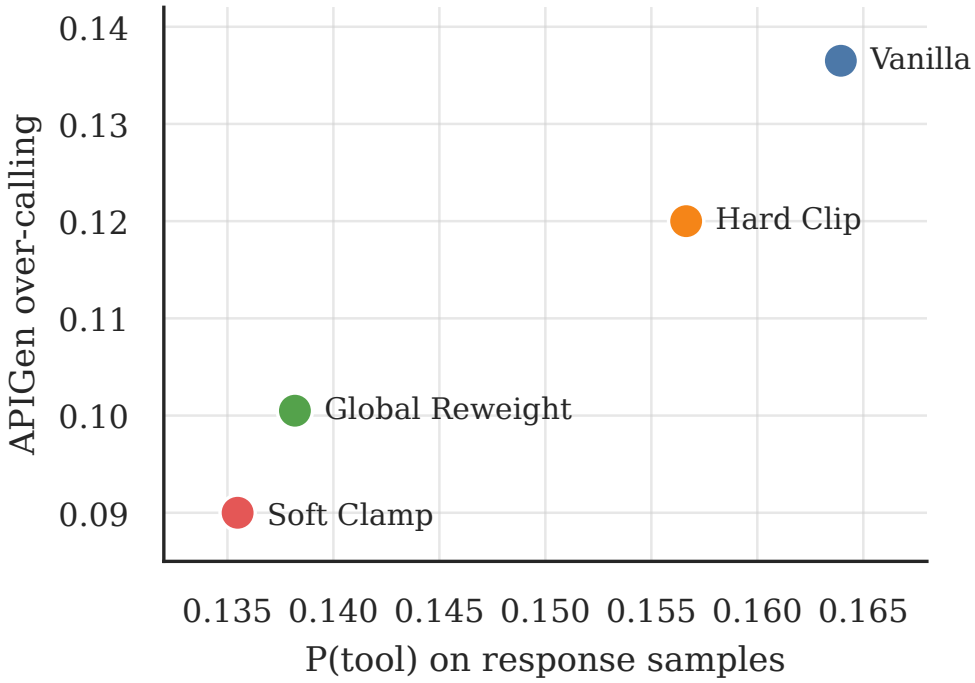

Figure 4: Response-side decision pressure and final over-calling in a representative diagnostic seed. Each point is one method from the same diagnostic run; the plot is descriptive and does not fit a regression trend.

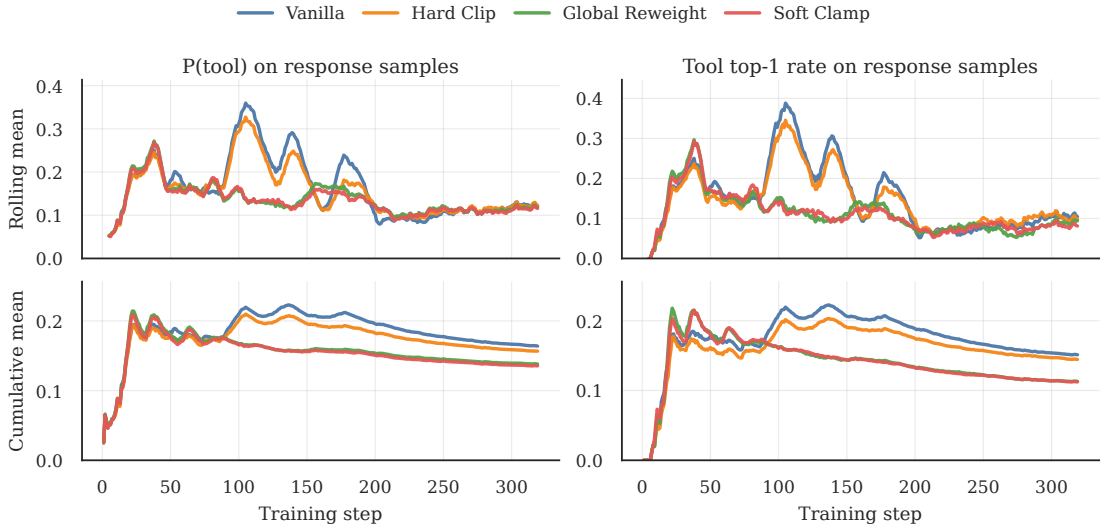

Figure 5: Step-level response-side decision pressure. Teacher-forced process metrics show that the pressure on response examples emerges during training.

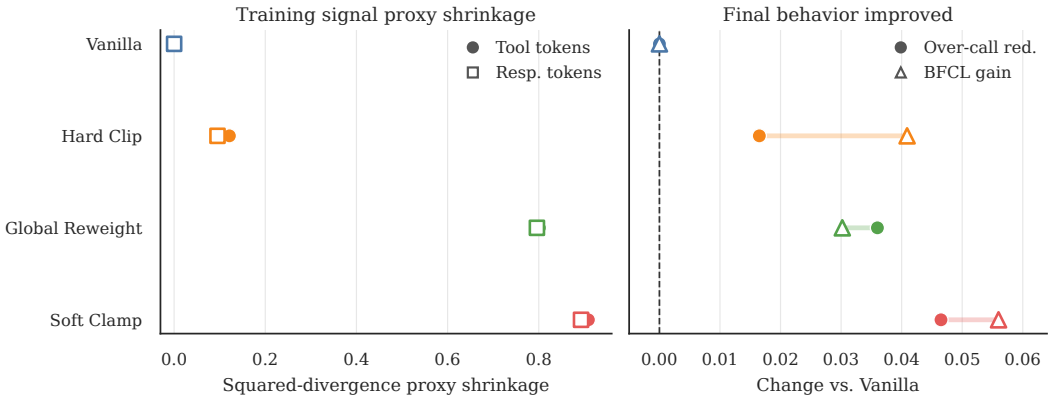

Figure 6: Intervention strength and behavior in the representative diagnostic run used for process analysis. Stronger compression of extreme token-level signals aligns with lower APIGen over-calling among GKD variants in this run, but this should not be read as "more compression is always better." Main benchmark rankings are reported separately with three-seed means in Section 5.

B.6 Token-level concentration diagnostics

Table 9 gives the multi-seed numerical values behind Figure 1. The concentration columns report the average share of per-sample JSD carried by the largest 1%, 5%, and 10% of supervised tokens. The shrinkage column reports the mean adjusted squared-divergence proxy divided by the raw squared-divergence proxy on diagnostic samples. The threshold-event column reports the fraction of diagnostic tokens that explicitly exceed a hard or soft clamp threshold; it is not defined for vanilla GKD or Global Reweight, because Global Reweight modifies token weights broadly rather than producing explicit clamp events.

Method	Top 1% JSD	Top 5% JSD	Top 10% JSD	Proxy shrinkage	Threshold events
Vanilla GKD	0.412±0.004	0.768±0.029	0.910±0.023	1.000±0.000	N/A
Hard Clip	0.389±0.003	0.763±0.026	0.907±0.022	0.972±0.002	0.003±0.001
Global Reweight	0.327±0.024	0.672±0.015	0.859±0.018	0.462±0.040	N/A
Soft Clamp	**0.154±0.032**	**0.507±0.021**	**0.794±0.031**	0.283±0.047	0.074±0.005

Table 9: Multi-seed diagnostic token-level signal concentration and compression. Values are mean±std over seeds 42, 44, and 60. Lower concentration indicates that fewer extreme tokens dominate the token-level JSD mass.

C Additional Benchmark Results

C.1 Additional APIGen-MT results

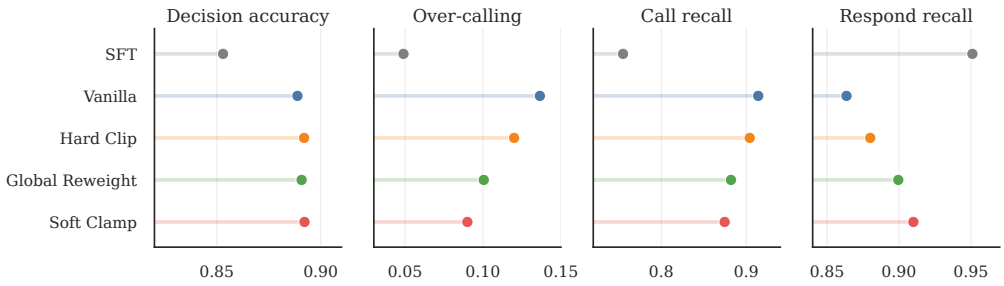

Figure 7: APIGen-MT-derived decision-set behavior trade-off. Error bars mark standard deviation over three seeds for GKD variants. Soft Clamp reduces over-calling relative to vanilla GKD while preserving decision accuracy, at the cost of lower call recall than vanilla GKD.

Case	User state	Vanilla GKD	Soft Clamp
Airline baggage	User asks to add one checked bag and believes it is covered by free allowance.	Calls a baggage-update tool directly.	Explains the free-baggage allowance and asks for confirmation before updating.
Retail address	User asks to change Suite 716 to Suite 1000 at the same street address.	Calls an address-modification tool directly.	Summarizes the address change and asks the user to confirm.
Flight refund	User asks whether a completed trip can receive refund or credit due to membership status.	Starts another tool-call trajectory after stating the policy.	Directly explains that completed flights cannot be refunded or cancelled.

Table 10: Qualitative APIGen-MT should-respond examples. Vanilla GKD enters a tool-call trajectory when the correct behavior is to answer, explain policy, or request confirmation. Soft Clamp stays on the response side of the behavior boundary.

C.2 Inference-time tool-entry bias counterfactual

We test whether the first-token boundary effect of Soft Clamp can be reproduced by a simple inference-time intervention: adding a non-positive scalar bias $b \in [-5, 0]$ to the <tool_call> entry logit of vanilla GKD. This diagnostic is computed at the first assistant token with thinking disabled, matching the inference template. It is not the same metric as full-generation APIGen over-calling. We therefore refer to the x-axis quantity as *response tool-entry error*: the fraction of should-respond examples for which <tool_call> is the top-1 first token.

For each vanilla seed, we sweep biases from -5 to 0 with step size 0.1 on the validation split. We select two operating points on validation only: one that matches Soft Clamp's validation response tool-entry error and one that matches Soft Clamp's validation tool-entry call recall. When multiple biases tie on the discrete top-1 metric, we choose the least invasive value, i.e., the one with the smallest $|b|$. The selected bias is then frozen and evaluated on the APIGen-MT test split. Because the first-token scores use a local Transformers forward pass while the main full-generation APIGen results use a vLLM-backed server, first-token and full-generation numbers are not directly backend-controlled comparisons. E1 should be read as a boundary counterfactual under a common local scoring protocol.

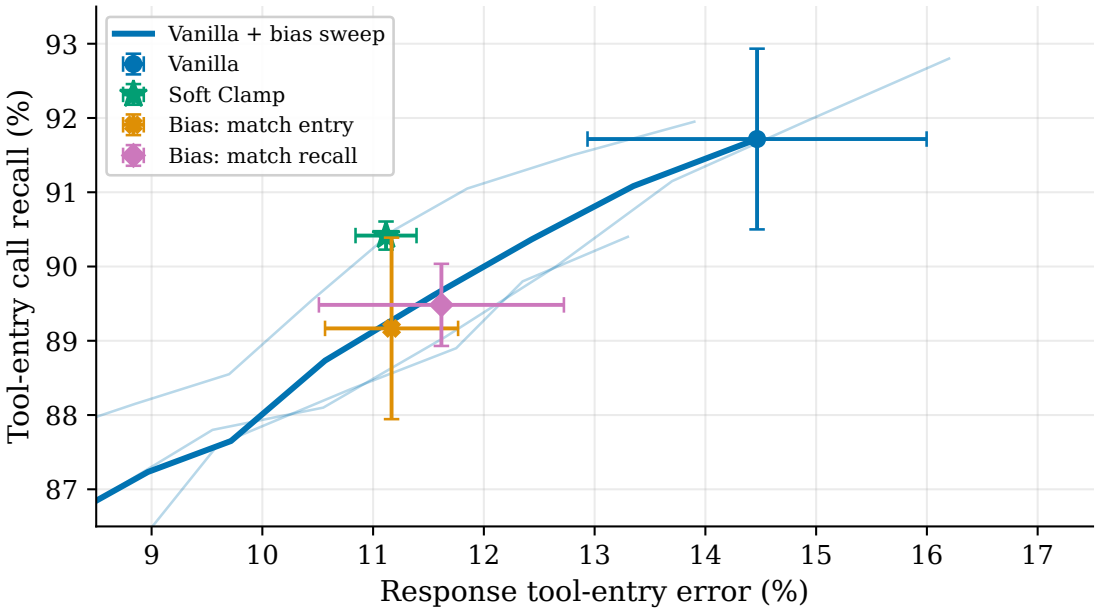

Figure 8: Inference-time tool-entry bias Pareto diagnostic on APIGen-MT test. Thin blue lines are per-seed vanilla GKD bias sweeps; the thick blue line is the three-seed mean. The green star is Soft Clamp without inference-time bias. Orange and purple markers show vanilla GKD with validation-selected biases, evaluated on test. A scalar `<tool_call>` bias reproduces most of the mean response-side entry reduction, but not the same joint call-recall and accuracy trade-off. Error bars are sample standard deviations over three seeds.

Setting	Response entry error	Tool-entry call recall	First-token acc.
Vanilla GKD, no bias	14.5±1.5	91.7±1.2	88.6±0.4
Soft Clamp, no inference bias	**11.1±0.3**	**90.4±0.2**	**89.7±0.1**
Biased vanilla, val-matched response entry error	11.2±0.6	89.2±1.2	89.0±0.6
Biased vanilla, val-matched call recall	11.6±1.1	89.5±0.6	88.9±0.5

Table 11: First-token boundary counterfactual on APIGen-MT test. Values are percentages over three paired seeds, with sample standard deviations. Biases are tuned on validation and frozen on test. The table should not be read as full-generation APIGen over-calling.

Seed	Val-matched response-entry bias	Val-matched call-recall bias
42	-0.40	-0.50
44	-0.65	-0.40
60	-0.25	-0.25

Table 12: Validation-selected non-positive `<tool_call>` entry biases used for the E1 counterfactual. Biases are selected on APIGen-MT validation and frozen for APIGen-MT test and strict BFCL multi-turn evaluation.

Setting	Boundary AUC
Vanilla GKD, no bias	0.9692±0.0023
Soft Clamp, no inference bias	0.9710±0.0011

Table 13: Threshold-free first-token boundary discrimination on APIGen-MT test under the local E1 scoring protocol. The small AUC difference indicates that much of the observed APIGen behavior change is an operating-point shift rather than a large separation improvement.

The result is nuanced. A per-seed validation-tuned scalar bias reproduces the mean reduction in response-side first-token entry error (11.2% versus 11.1%). However, at that operating point it yields lower tool-entry call recall (89.2% versus 90.4%), lower first-token decision accuracy (89.0% versus 89.7%), and larger seed variability than Soft Clamp. Thus scalar entry suppression explains most of the mean response-side entry reduction, but only part of the joint boundary-calibration benefit. Table 13 also shows that the threshold-free boundary AUC changes only slightly, which is consistent with an operating-point interpretation.

The first-token diagnostic is close to, but not identical with, full-generation APIGen over-calling. In the paired E1 diagnostic artifacts, vanilla GKD has 14.5±1.5% first-token response entry error and 14.3±2.2% full-generation over-calling, close to the canonical main-table value of 14.2±2.1%. Soft Clamp has 11.1±0.3% first-token response entry error and 9.0±0.2% full-generation over-calling. The larger gap for Soft Clamp is associated with examples whose top-1 first token is `<tool_call>` but whose generated output does not contain the `<tool_call>` marker after removing thinking blocks. Because the first-token and full-generation artifacts use different backends, this alignment check should not be used to isolate downstream generation behavior by itself.

We also run a strict multi-turn counterfactual on the BFCL multi-turn diagnostic. Because the original BFCL multi-turn runs use a vLLM server that does not expose a first-token-only logit hook, this counterfactual uses a local Transformers decoder for all compared runs. These numbers should therefore be compared only within Table 14, not mixed with the vLLM-decoder multi-turn results in Table 4. The purpose of this table is counterfactual isolation rather than benchmark replacement: by using the same decoder, prompt rendering, parser, and first-token bias hook for all rows, it tests whether inference-time entry suppression can match Soft Clamp under a controlled protocol. Its absolute values can differ from the vLLM-based multi-turn table for the same checkpoint because the local strict decoder and the main vLLM protocol differ in generation backend, request batching, and chat-template rendering details, although both use thinking-disabled greedy decoding. We therefore interpret only within-protocol differences in this counterfactual.

Setting	Calls/turn	Loop@3	Loop@5	Repeat	Non-tool final	Invalid call
Vanilla GKD, no bias	1.734±0.057	20.5±1.5	15.1±1.6	23.7±4.3	78.8±7.0	2.3±2.2
Soft Clamp, no inference bias	1.494±0.053	15.6±0.8	10.1±1.1	**16.6±1.5**	**88.2±1.3**	**0.7±0.1**
Biased vanilla, val-matched response entry error	**1.492±0.089**	**15.7±1.0**	**10.3±1.2**	17.6±0.6	85.8±1.5	1.6±1.2
Biased vanilla, val-matched call recall	1.525±0.062	16.3±1.3	11.0±1.3	18.8±4.0	84.7±4.8	1.8±1.5

Table 14: Strict BFCL multi-turn counterfactual with the same local first-token-bias decoder for all rows. Values are three-seed mean±sample std. The per-seed biases selected on APIGen-MT validation are reused here without BFCL tuning. Inference-time scalar bias reproduces nearly all of the mean reduction in tool-call frequency and loop rate under this strict decoder, but Soft Clamp retains higher non-tool-final rate and lower invalid-call rate. Non-tool final is a termination metric, not answer correctness.

This multi-turn diagnostic strengthens the same nuanced interpretation. Under these validation-selected operating points, scalar entry suppression nearly matches Soft Clamp on call frequency and loop rate, but it does not fully match the mean non-tool-final and invalid-call profile. We therefore treat scalar bias as a

strong counterfactual baseline, not as a complete replacement for training-time calibration. This first-token and strict-decoder diagnostic does not determine whether the remaining gap would persist under other full-generation or multi-turn inference-time calibration protocols.

C.3 BFCL and When2Call tables

Method	Overall	Tool Call Quality	Irrelevance Refusal
Base	82.2	83.5	79.8
Base SFT	82.9	82.0	84.4
Vanilla GKD	79.0±1.7	79.4±1.5	78.2±2.9
Hard Clip	79.9±0.6	78.1±1.2	83.2±1.2
Global Reweight	79.8±2.7	77.8±5.7	**83.5±3.4**
Soft Clamp	**80.8±0.1**	**79.7±1.3**	82.7±2.4

Table 15: BFCL results. All values are percentages. GKD rows report mean±std over three seeds. Bold marks the best mean among GKD variants in each column. Soft Clamp has the strongest GKD mean overall score, but it does not dominate the base or base SFT.

Method	MCQ Acc	tool_call	request_for_info	cannot_answer
Base	72.8	88.1	61.6	68.1
Base SFT	71.6	88.8	63.3	63.1
Vanilla GKD	65.6±1.1	**88.0±1.4**	58.9±5.8	51.2±4.1
Hard Clip	64.3±1.3	87.9±1.8	57.3±4.7	49.2±4.5
Global Reweight	**67.1±2.2**	87.3±2.5	**61.5±6.9**	**53.8±3.4**
Soft Clamp	65.9±2.5	87.7±1.2	60.3±2.2	51.3±6.7

Table 16: When2Call results. All values are percentages. GKD rows report mean±std over three seeds. This benchmark limits the scope of the claim: the current GKD variants do not improve all out-of-domain tool-use decisions.

D Soft Clamp Token-position Diagnostics

Figure 9 complements the targeted-clamp ablation. Only 2.1% of sampled clamped events occur in the first four target positions, and XML markers account for 1.7% of clamped events. The strongest localized pattern appears in student-generated tool-call trajectories after a closed tool-call block, where the tool-call teacher assigns low probability to natural-language continuation. This supports the sparse trajectory-level mismatch interpretation and argues against the sufficiency of a simple XML-entry-token-only mechanism.

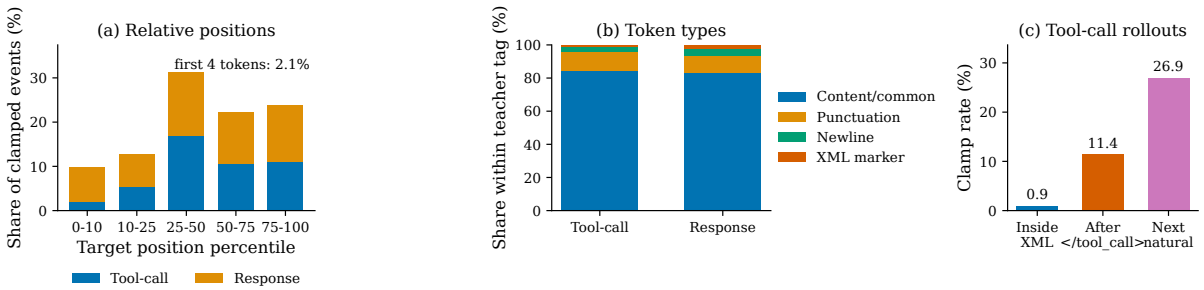

Figure 9: Sampled Soft Clamp token-position diagnostics from 62 diagnostic rows, 23,923 supervised loss tokens, and 1,804 clamped events. Panel (a) shows that clamped events are distributed across the target rather than concentrated in the first few supervised tokens. Panel (b) shows that most clamped events are ordinary content or common tokens, not XML markers; this is event composition, not a category-normalized clamp rate. Panel (c) focuses on student-generated tool-call trajectories under the tool-call teacher: clamp rates are much higher after a closed `</tool_call>` block, especially when the student continues with natural language, than inside the XML block. These sampled diagnostics show where Soft Clamp compresses mismatch; they do not by themselves prove how the tool-call entry boundary changes.

E Targeted Clamp Ablation

To test whether the benefit of Soft Clamp comes only from suppressing the first few supervised target tokens, we run two eligible-budget-matched variants. Entry-k Soft Clamp applies the same dynamic threshold as Soft Clamp but only allows the first four supervised loss tokens in each sample to be eligible for clamping. Random-k Soft Clamp allows four pseudo-random supervised positions per sample to be eligible for clamping. Both variants use the same training data, teachers, OPD settings, SFT anchor, and clamp factor as the main Soft Clamp run.

Method	Decision Acc	Over-calling	Call Recall	Respond Recall
Vanilla GKD	88.88	13.65	91.40	86.35
Soft Clamp	**89.22**	**9.00**	87.45	**91.00**
Entry-k Soft Clamp	88.52	13.75	90.80	86.25
Random-k Soft Clamp	88.60	13.15	90.35	86.85

Table 17: Single-seed targeted-control ablation on APIGen-MT. All values are percentages. Entry-k makes only the first four supervised loss tokens eligible for clamping, while Random-k uses an eligible-budget-matched pseudo-random set of supervised tokens.

Table 17 shows that Entry-k does not reproduce the full Soft Clamp result. Its over-calling rate is 13.75%, close to vanilla GKD's 13.65%, while full Soft Clamp lowers over-calling to 9.00%. Random-k is slightly below vanilla GKD at 13.15%, but remains much closer to vanilla GKD than to full Soft Clamp. Thus, this single-seed control does not support the sufficiency of first-four-token-only clamping.

The diagnostic records in Table 18 confirm that the controls behaved as intended: Entry-k only triggers clamping at indices 0–3, whereas Random-k triggers clamping at distributed later positions. The ablation therefore refines the diagnostic picture rather than establishing a complete causal mechanism. Compressed mismatches appear local and sparse, but in the current XML-style rendering they are distributed beyond the first few supervised target tokens. Full Soft Clamp lets the data determine which extreme token-level divergences are compressed, instead of hand-coding a narrow entry-token mask.

Variant	Eligible rule	Actual clamped events	Min index	Median index	Max index	Events with index < 4
Entry-k	0–3 / sample	36	0	2	3	36
Random-k	4 random / sample	20	16	147	596	0

Table 18: Scope validation for the targeted-clamp diagnostic samples. The two variants use the same eligible-token budget per sample. The number of actual clamped events can differ because a token is clamped only when its divergence exceeds the dynamic threshold.

F Loss Modifier Details

F.1 Vanilla GKD

Vanilla GKD uses the original token divergence d_i without additional calibration. It is the reference point for measuring whether a loss modifier reduces behavior imbalance.

F.2 Soft Clamp

Soft Clamp sets a detached dynamic threshold $C = \mathrm{stopgrad}(k \,\mathrm{mean}_i(d_i))$ for each batch. We use $k = 3.0$ in the main experiment. Tokens with $d_i \leq C$ are unchanged. Tokens with $d_i > C$ use

$$d_i' = d_i \frac{C}{\mathrm{stopgrad}(d_i)}. \tag{8}$$

The forward value is capped at C, but the token still receives a nonzero gradient scaled by C/d_i. Detaching the threshold prevents gradients from flowing through the batch mean and matches the implementation.

F.3 Soft Clamp factor sensitivity

In a single-seed factor sweep using the representative training seed, we compare clamp factors 2.0, 3.0, and 4.0 under the same data, teachers, OPD settings, and supervised format anchor. The factor changes only the dynamic threshold $C = \mathrm{stopgrad}(k \,\mathrm{mean}_i(d_i))$.

Method	Decision Acc	Over-calling	Call Recall	Respond Recall
Vanilla GKD	88.88	13.65	91.40	86.35
Soft Clamp factor 2	88.85	11.50	89.20	88.50
Soft Clamp factor 3	**89.22**	**9.00**	87.45	**91.00**
Soft Clamp factor 4	88.92	9.40	87.25	90.60

Table 19: Single-seed APIGen-MT clamp-factor sensitivity. All values are percentages. Factor 3 is the default used in the main experiments.

Table 19 shows that the effect is not tied to a single narrow threshold. Factors 2, 3, and 4 all reduce over-calling relative to vanilla GKD while preserving decision accuracy. Factor 3 gives the strongest result in this small sweep, but the main conclusion is robustness over a reasonable range, not universal optimality of this value.

F.4 Hard Clip

Hard Clip applies

$$d'_i = \min(d_i, c), \tag{9}$$

with $c = 0.5$ in the main experiment. This baseline tests whether simply truncating extreme token losses can reduce over-calling. It is a practical contrast rather than a mechanism-matched control: it uses a fixed threshold, removes marginal gradients above the threshold, and affects a different number of tokens from Soft Clamp.

F.5 Global Reweight

Global Reweight is a batch-relative baseline. Let

$$z_i = \mathrm{clip}\left(\frac{d_i - \mathrm{mean}(d)}{\mathrm{std}(d)}, -z_{\max}, z_{\max}\right). \tag{10}$$

The token weight is

$$w_i = \frac{\exp(-\alpha z_i)}{\mathrm{mean}_j \exp(-\alpha z_j)}, \tag{11}$$

followed by clipping to $[w_{\min}, w_{\max}]$. The adjusted divergence is $d'_i = d_i \,\mathrm{stopgrad}(w_i)$. We use $\alpha = 0.3$, $z_{\max} = 3.0$, $w_{\min} = 0.25$, and $w_{\max} = 2.0$. This method is useful as a contrast because it reweights all tokens by batch-relative divergence, while Soft Clamp leaves non-extreme tokens unchanged.

G Reproducibility Checklist

The main experiments use one model initialization, one pair of teachers, one training split, and one anchored GKD recipe. Student GKD training is repeated over three random seeds, and final single-turn evaluation is then run on the resulting checkpoints. The following items are fixed across all GKD variants unless a loss modifier explicitly changes them: initialization checkpoint, train/validation data, teacher routing, teacher API type, maximum sequence lengths, batch size, learning rate schedule, rollout mixture, SFT-anchor weight, teacher top-k logit truncation, checkpoint interval, and evaluation datasets. The loss modifier is the only intended difference among vanilla GKD, Hard Clip, Global Reweight, and Soft Clamp.

For each run, we store final outputs, benchmark scores, training logs, diagnostic samples, and the plotting inputs used to generate the paper figures. The planned public artifact will include the modified GKD backend, anonymized training-script templates, evaluation scripts, metric definitions, and plotting scripts. It will not include private service URLs or internal filesystem paths.

H BFCL Multi-turn Loop Diagnostic

The BFCL multi-turn diagnostic evaluates whether a model with a stronger tool-call prior becomes less usable in an interactive tool environment. It uses 800 BFCL v4 multi-turn tasks and 3,136 user turns sampled from the local processed BFCL multi-turn data. When the model emits a tool call, the harness returns a fixed simulated tool observation generated by the local BFCL-style harness and continues the dialogue until the model gives a non-tool final answer or reaches the maximum step limit. Invalid calls are counted when the model emits a tool-call marker that the harness cannot parse into a supported function call.

We report six diagnostic metrics. *Tool calls per turn* counts the average number of tool calls emitted per user turn. *Loop@3* and *Loop@5* measure whether the model keeps calling tools for at least three or five consecutive steps. *Max-step hit* records turns that end by reaching the harness limit. Because this harness

stops after five tool-call steps, Loop@5 and Max-step are numerically identical; we report both names to make the stopping condition explicit. *Repeat same call* records repeated calls to the same function with the same arguments. *Non-tool final answer* records whether the model eventually returns a non-tool response; it does not measure whether that response is task-correct.

This diagnostic is not a replacement for official BFCL scoring. It is a usability stress test for the specific failure mode studied in this paper: a model can improve should-call recall while becoming too willing to enter tool-call trajectories during multi-turn interaction.

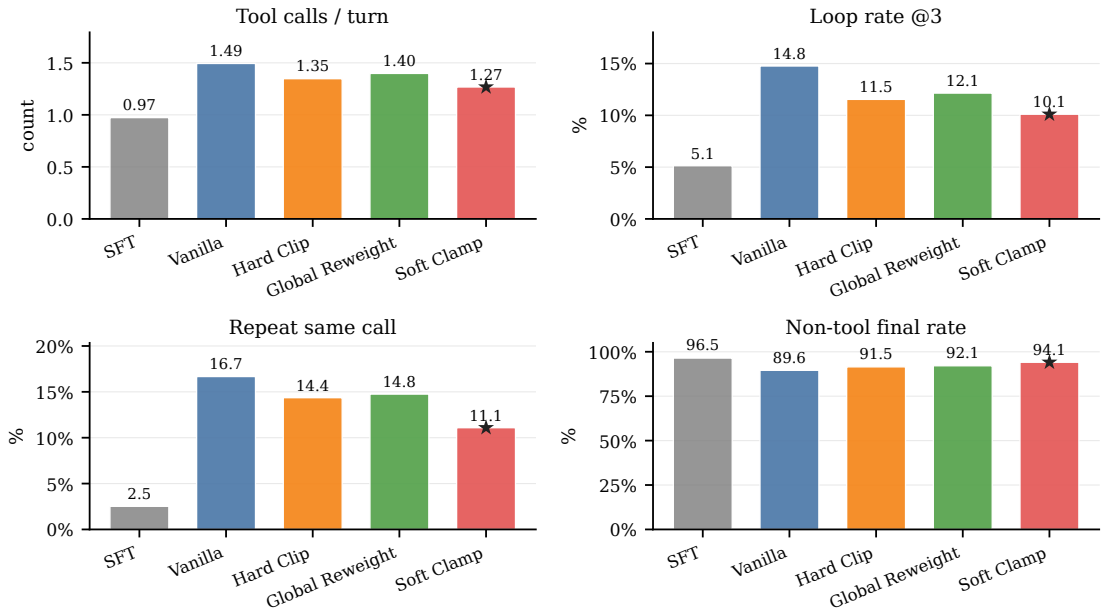

Figure 10: BFCL multi-turn loop diagnostic plot. Error bars mark standard deviation for Vanilla GKD, Global Reweight, and Soft Clamp; Base SFT and Hard Clip are single diagnostic runs. Over-calling in single-turn evaluation corresponds to more tool calls, higher loop rates, and fewer non-tool final answers in a simulated multi-turn tool environment.

I Qwen3.5-4B Smaller-Student Replication

Section 5.7 reports the main 4B replication summary. This appendix gives the full single-turn and multi-turn diagnostics. The 4B experiments use the same data, teachers, OPD settings, and evaluation harness as the main 9B experiments, but replace the Qwen3.5-9B student with a Qwen/Qwen3.5-4B student under the fixed 9B teacher pair. GKD rows report mean$\pm$std over seeds 42, 44, and 60; the Base 4B row is a single reference run. Base SFT 4B is also included as a single reference run.

Method	Decision Acc	Over-calling	Call Recall	Respond Recall
Base 4B	79.6	16.9	76.0	83.2
Base SFT 4B	88.5	7.5	84.5	92.5
Vanilla GKD 4B	**88.4±0.8**	12.8±0.7	**89.6±2.1**	87.2±0.7
Global Reweight 4B	87.8±0.8	9.5±0.8	85.1±1.9	90.5±0.8
Soft Clamp 4B	87.6±0.6	**9.2±1.2**	84.4±1.8	**90.8±1.2**

Table 20: Qwen3.5-4B APIGen-MT decision results. All values are percentages. Bold marks the best displayed GKD value in each column.

Method	BFCL	BFCL TC	BFCL IR	W2C	request	cannot	tool
Base 4B	79.8	80.6	78.2	70.4	63.0	67.5	80.2
Base SFT 4B	78.2	73.1	87.9	62.0	64.7	45.8	78.4
Vanilla GKD 4B	75.6±0.1	69.8±0.4	86.3±0.4	**59.9±3.0**	**68.2±7.3**	**38.6±5.4**	77.5±3.4
Global Reweight 4B	75.6±0.6	68.9±2.1	**88.1±2.8**	58.1±1.6	66.8±3.0	34.5±4.9	77.8±1.9
Soft Clamp 4B	**76.0±0.4**	**70.7±1.3**	85.9±1.8	58.1±1.0	67.2±2.9	33.9±3.4	**78.1±2.1**

Table 21: Qwen3.5-4B BFCL and When2Call results. All values are percentages. BFCL TC denotes BFCL tool-call quality, BFCL IR denotes irrelevance refusal, and W2C denotes When2Call MCQ accuracy. Bold marks the best displayed GKD mean in each column.

Method	Calls/turn	Loop@3	Loop@5	Repeat call	Non-tool final
Base 4B	1.317	12.3	4.8	10.0	95.2
Base SFT 4B	0.999	5.7	0.8	2.5	97.4
Vanilla GKD 4B	1.493±0.122	15.7±3.2	9.9±3.4	17.6±4.2	88.2±3.9
Global Reweight 4B	**1.200±0.034**	**8.9±1.4**	**3.8±1.7**	**9.7±3.8**	**95.4±1.7**
Soft Clamp 4B	1.235±0.097	9.2±2.6	4.2±2.1	10.1±4.1	94.8±2.1

Table 22: Qwen3.5-4B BFCL multi-turn loop diagnostic. All values except calls/turn are percentages. Non-tool final is a termination metric, not answer correctness. Bold marks the best displayed GKD value in each column.

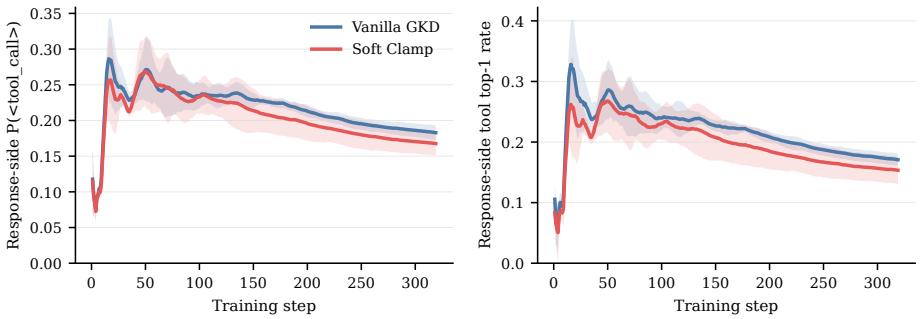

Figure 11: Cumulative Qwen3.5-4B response-side decision pressure through student training. Lower response-side $P(\texttt{<tool\_call>})$ and lower response-side tool top-1 rate indicate weaker pressure toward entering tool-call mode on response-labeled targets.

The 4B diagnostics match the main 9B story but should be interpreted as a targeted smaller-student replication, not as a broad scaling law. Soft Clamp and Global Reweight both reduce over-calling and

multi-turn loop risk relative to vanilla GKD. In this 4B setting, Global Reweight is slightly stronger on the multi-turn means, while Soft Clamp has the lowest APIGen-MT over-calling mean among GKD variants and remains competitive on loop metrics. The response-side diagnostic also moves in the same direction for Soft Clamp: at step 319, response-side $P(\texttt{<tool\_call>})$ drops from 0.1828±0.0103 to 0.1676±0.0169, and response-side tool top-1 rate drops from 17.1±1.0% to 15.3±2.2%. At the same time, the 4B results retain the same call-recall trade-off seen in the 9B experiments, and the broader BFCL and When2Call metrics remain mixed rather than showing a uniform capability gain.

J Checkpoint-level Transient Analysis

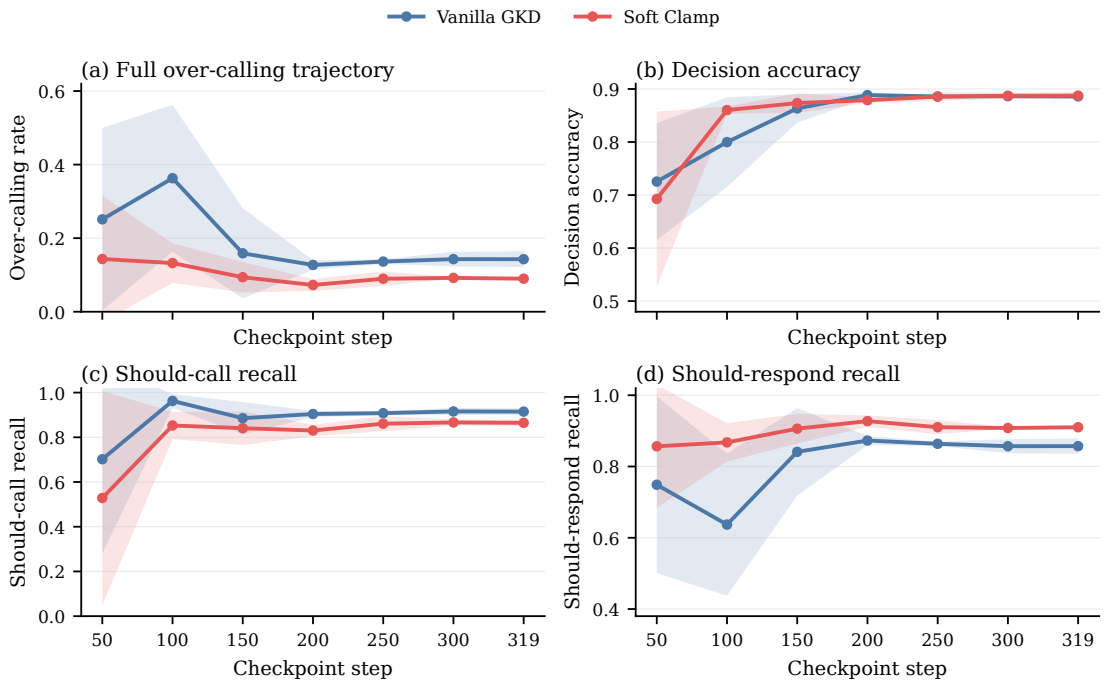

Figure 12: Full APIGen-MT checkpoint-level trajectory for vanilla GKD and Soft Clamp. Lines show means over three seeds and bands show standard deviations. Early over-calling transients are seed-dependent, but the post-transient region shows a consistent gap between vanilla GKD and Soft Clamp.

Figure 12 shows the full checkpoint range. Vanilla GKD has a larger early over-calling transient than Soft Clamp on average, but the exact spike location is not fixed across seeds. One vanilla seed spikes at checkpoint 100, another spikes at checkpoint 50, and another peaks later at checkpoint 150. We therefore do not interpret checkpoint 100 as a universal training phase. Instead, the early region indicates that OPD can temporarily destabilize the tool-call decision boundary.

The post-transient region is more consistent. Vanilla GKD has higher mean over-calling than Soft Clamp at every evaluated checkpoint from 150 onward; from checkpoint 200 onward, the paired-seed ordering also holds in all three seeds. The final checkpoint remains separated: 14.2±2.1% for vanilla GKD versus 9.0±0.2% for Soft Clamp. This is why the main trajectory analysis in Figure 2 focuses on the post-transient region from checkpoint 150 onward.

K Pure OPD Format Stability

Structured tool-call training has two separable failure modes. The model can choose the wrong behavior mode, such as calling tools on should-respond examples, and it can also emit a malformed structured call after choosing the tool-call mode. The main experiments use a small supervised loss with `sft_alpha`=0.3 as a format anchor. Because this anchor is shared by all GKD variants, it does not explain the differences between vanilla GKD, Hard Clip, Global Reweight, and Soft Clamp.

We separately evaluate a pure teacher-distillation OPD setting without this anchor. This run uses `lmbda`=1.0, `sft_alpha`=0.0, and `beta`=0.5. It therefore should be interpreted as a pure OPD format-stability comparison, not as a single-parameter ablation of the main vanilla GKD run, whose main settings use `lmbda`=0.8 and `sft_alpha`=0.3.

Method	Decision Acc	Over-calling	Call Recall	Respond Recall
Pure OPD	58.35	83.20	99.90	16.80
Vanilla GKD + anchor	88.88	13.65	91.40	86.35
Soft Clamp + anchor	89.22	9.00	87.45	91.00

Table 23: APIGen-MT pure-OPD format-stability comparison. All values are percentages. Anchored GKD rows are representative single-run values included for format-stability comparison; main APIGen-MT multi-seed benchmark results are reported in Table 3. Pure OPD nearly always enters the tool-call mode, which maximizes call recall but collapses should-respond behavior.

Table 23 shows the decision-level effect on APIGen-MT. Pure OPD reaches 99.9% marker-level should-call recall, but over-calling rises to 83.2%. The model therefore does not merely improve should-call recall; it loses the decision boundary between calls and direct responses.

Method	Tool-call rate	Valid XML	Malformed	Valid among calls
Pure OPD	91.55	14.75	76.80	16.11
Vanilla GKD + anchor	52.52	43.10	9.43	82.06
Soft Clamp + anchor	48.23	38.77	9.45	80.40

Table 24: APIGen-MT strict tool-call format statistics. Anchored GKD rows are representative single-run values included for format-stability comparison. A valid XML call must contain a closed `<tool_call>` block with a nested `<function=...>` block and parameter tags.

Table 24 shows the schema effect. Pure OPD emits `<tool_call>` markers on most examples, but only 16.1% of emitted calls are valid under the strict XML parser. A common failure is JSON-like content inside the XML marker, for example `<tool_call>{...}`, instead of the expected XML-style `<function=...>` and `<parameter=...>` fields.

On BFCL single-turn evaluation, pure OPD obtains only 5.16% overall accuracy and 0.05% tool-call AST accuracy (Table 25). A strict format scan gives the same conclusion: 94.87% of BFCL outputs contain a `<tool_call>` marker, but 0.00% are valid XML-style calls under our parser. Thus the failure is not simply a lower benchmark score; the structured protocol itself is unstable.

Table 26 reports the multi-turn diagnostic. The zero parsed calls for pure OPD should not be read as conservative behavior. The raw rollouts contain malformed `<tool_call>` strings, so the harness cannot parse them into executable calls. Consequently, every model step is counted as invalid and no final answers are produced.

The training logs also show the same direction before evaluation. On response-labeled examples, pure

Method	BFCL overall	Tool-call AST	Irrelevance refusal
Pure OPD	5.16	0.05	14.77
Vanilla GKD + anchor	79.85	80.91	77.85
Soft Clamp + anchor	80.62	79.11	83.45
Base SFT	82.87	82.05	84.43

Table 25: BFCL single-turn pure-OPD format-stability comparison. All values are percentages. Anchored GKD rows are representative single-run values; main BFCL multi-seed benchmark results are reported in Table 15. Pure OPD almost completely fails executable function-call evaluation.

Method	Calls/turn	Loop@3	Invalid call	Non-tool final
Pure OPD	0.000	0.0	100.0	0.0
Vanilla GKD + anchor	1.494	14.8	0.8	89.6
Soft Clamp + anchor	1.268	10.1	0.6	94.1
Base SFT	0.974	5.1	1.4	96.5

Table 26: BFCL multi-turn pure-OPD format-stability comparison. Values except calls/turn are percentages. Anchored GKD rows are representative single-run values; main BFCL multi-turn benchmark results are reported in Table 4. Pure OPD has zero parsed calls because every tool-call attempt is malformed under the harness parser. Non-tool final is a termination metric, not answer correctness.

OPD has a mean `decision_toolcall_prob_response_mean` of 0.5444 and a final-step value of 0.6729. The corresponding values are 0.1634 and 0.0731 for anchored vanilla GKD, and 0.1349 and 0.1049 for anchored Soft Clamp. This supports a limited conclusion: the tested pure teacher-distillation OPD configuration is unstable in this structured tool-call setting. Because this stress test changes both the rollout mixture and the supervised-anchor weight, it should not be read as an anchor-only causal ablation. The main GKD comparisons therefore keep the anchored recipe fixed and study behavior calibration within that recipe.