# General-Purpose ICL Simulator Roadmap (Expanded for Alias-Swap Transfer Testing)

## Alias-Swap Transfer Testing Experiment

**Overview:** We introduce an explicit *alias-swap* experiment in the in-context learning simulator to test the model’s ability to bind a concept to one alias and transfer that knowledge when a new alias is introduced. This leverages the platform’s anonymization system (per-user codebooks with symbol remapping) and evidence-gated scoring to ensure the model truly learns from context rather than from prior knowledge. In practice, the model is first trained in-context using **alias set A** (e.g. using a placeholder term like “Zim-7” in place of the real concept *classical conditioning*), then later in the same session tested on **alias set B** (e.g. “Concept-92” for the *same* concept). The model must infer that B refers to the previously learned concept by using contextual clues, accumulated notes, or prior examples – without any direct hint – since both aliases are gibberish tokens that never appeared in pretraining[[1]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU). A *witness-gated* scoring mechanism will validate that when the model answers correctly under alias B, it does so by citing or referencing the earlier alias-A context as evidence, proving the transfer is grounded in the session content[[2]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU).

### Structuring Alias-Swap Item Families

To implement this, we structure **alias-swap item families** in the content pool. Each family centers on one underlying concept/skill but has at least two versions of question content: - **Alias A variant (training instance):** An MCQ or SAQ that introduces and uses alias A in place of the target concept. For example, an MCQ could ask: *“Which scenario is an example of* *Zim-7?”* with options all phrased using "Zim-7" instead of the real term. The correct answer and rationale will likewise use alias A. This exposes the model to the concept under the alias A terminology. - **Alias B variant (transfer test):** A later question (MCQ or SAQ) covering the *same* concept but using alias B. For example: *“Concept-92* *was first demonstrated by Pavlov in his experiments with dogs. What principle does this refer to?”* Here **Concept-92** is a new nonsense label for the concept that was referred to as Zim-7 earlier. The content of the question provides contextual clues (e.g. mentioning Pavlov, dogs) that tie back to the notes from the alias A question. The model must recognize from context that “Concept-92” likely maps to the same idea as “Zim-7” and answer accordingly.

**Design considerations:** - Both alias terms A and B should be **equally novel** and meaningless to the model’s parametric memory. We ensure this by using the platform’s anonymization feature to generate unique codewords (e.g. random strings like “Zim-7”, “Concept-92”) that have not appeared in pretraining[[3]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,answer%3B%20no%20raw%20domain%20terms). Each alias-swap family will have its own pair of gibberish tokens for the concept. - *Content cues:* The A and B variants should share some identifiable context so the model can infer the mapping. For instance, if alias A’s explanation note says “**Zim-7** is a learning process first studied by Ivan Pavlov,” then alias B’s question can include the clue “first demonstrated by Pavlov” – allowing the model to connect **Concept-92** to the same concept via the Pavlov reference. We do **not** explicitly tell the model that A = B; it must deduce this from such contextual overlaps in the content, or by cross-referencing the **NOTES** it accumulated from the alias A question. - *Question format:* We can use MCQ for both A and B for consistency, or introduce variety (e.g. MCQ for A to teach with immediate feedback, then a short-answer question for B to test open-ended recall under the new alias). The key is that the *underlying concept and difficulty* remain the same between the variants, isolating the effect of alias change. Each family can thus be considered a pair (or small set) of items that differ only in the alias labels used.

### Experiment Protocol and Workflow

**Session flow integration:** The alias-swap test fits into the simulator’s run loop by scheduling the B variant after the model has seen the A variant. A typical workflow within a synthetic student session might be: 1. **Introduce alias A:** Present the alias-A version question for a concept. The model attempts it. If it answers incorrectly, standard remediation applies (feedback and possibly a retry or later re-test of A until the concept is at least partially learned or the notes are recorded). Once the model answers A correctly (or at least has seen the explanation), the system records a **NOTE** consisting of the correct answer and rationale for alias A[[4]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU). 2. **Intervening content:** Optionally, insert some spacing (unrelated questions or a time gap) before alias B to avoid an immediate trivial pattern match. This mimics a realistic scenario and allows testing spaced transfer (alias B could even appear in a later session to test long-term retention of the mapping). 3. **Present alias B:** Later in the session, the scheduler injects the alias-B variant question for the same concept. The simulator’s selection policy should recognize when to trigger this – for example, once the concept behind alias A is marked “mastered” or at least answered correctly, schedule the B version as a transfer test. The reason field for item selection can note it as an “alias-transfer test” in the API (similar to how we denote review or new skill) for clarity. 4. **Model infers and answers:** The model must rely on the prior alias A exposure to answer the alias B question. Operating in closed-book provable-novice mode, the synthetic student can only use the accumulated notes (and any clues in the question) – it cannot fall back on world knowledge of the actual concept since all domain terms are scrambled[[1]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU). Ideally, if the model truly learned, it will retrieve the note about **Zim-7** from earlier, realize Concept-92 appears to refer to the same concept described in that note, and produce the correct answer (with justification). 5. **Witness-gated evaluation:** When recording the model’s answer for alias B, the platform should apply **evidence-gated scoring**[[2]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU). In practice, this means the answer is only credited if two conditions hold: (a) it is factually correct (the chosen answer or response indeed corresponds to the concept), *and* (b) the model’s answer is well-supported by prior context that used alias A. The support is verified via **coverage and witness check** – i.e., the student model must cite the relevant earlier note or rationale, and the server performs a witness re-pick using those citations to see if the same answer is reached[[2]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU). This ensures that a correct answer under alias B isn’t a lucky guess or parametric recall, but is grounded in the alias A evidence. In other words, the model gets full credit only if it demonstrably leveraged the alias A context to answer alias B (the “witness” being the alias A note). 6. **Remediation if needed:** If the model fails on alias B, the system can provide an explanation (which now might explicitly link alias B to the concept for learning purposes) and log that the transfer attempt was unsuccessful. The family could be retried later (possibly introducing yet another alias or reusing B after more study) to see if additional exposures help, contributing data to the *binding slope* metric (described below).

This protocol should be integrated such that alias-swap tests occur organically in the session, much like other diagnostics (spacing, reflection prompts, etc.). The run loop (e.g., the logic in student\_bot\_deepinfra.py or the adaptive next API) can be extended to recognize alias-swap opportunities. For instance, when TUTOR\_ANONYMIZE=1 is enabled and a skill has an alias-swap family defined, the system inserts the B variant at the appropriate time with a special marker in the reason (like "reason": "alias\_transfer\_test"). This ensures the synthetic student and the server handle it with the needed evidence requirements.

### Data Logging for Alias-Swap Trials

To analyze alias-swap performance, the simulator should log detailed data for each alias family. Important data to capture includes: - **Alias Set Identifier:** Mark each question attempt with whether it was using alias set A or alias set B (or potentially C if more than two aliases in a chain). This could be a simple tag in the logs (e.g., alias\_set: "A" vs "B") or an extension of the item metadata. By tagging each item, we can segregate performance by alias set easily. - **Family/Concept ID:** An identifier for the underlying concept being tested. This ties the A and B items together. (In practice, if both alias A and B map to the same skill\_id or concept id in the content, that could serve as this link. If the platform treats them as separate skills for anonymization purposes, we may need a mapping table to indicate which skill aliases correspond to the same real concept.) - **Outcome and Accuracy:** Record whether the model’s answer was correct for each attempt. We specifically want **per-alias-set accuracy** – e.g., how often did the model get questions right when using alias A vs when using alias B. In the logs, we can maintain counters for correct/total on A items and on B items for each family (and aggregated across families as well). - **Coverage Score:** If using evidence-gated scoring, log the coverage value for the alias B answers. *Coverage* here refers to the fraction of the gold explanation content that was covered by the student’s cited notes[[2]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU). This indicates how much of the relevant alias A information the model brought to bear when answering alias B. A high coverage means the model cited most of the important points from the alias A note in its alias B answer, suggesting a strong transfer of knowledge. The server already computes and returns coverage for answers when citations are required[[5]](file://file-3LWaoPLYdoW8YgsE374zZh#:~:text=,witness_pass), so this can be recorded for analysis. - **Witness Check Result:** Log whether the **witness re-pick** succeeded for the alias B answer. The platform’s scoring returns a boolean witness\_pass (or equivalent) when citation gating is on[[5]](file://file-3LWaoPLYdoW8YgsE374zZh#:~:text=,witness_pass). We should capture this, as it reflects consistency: a witness\_pass=true on an alias B question means the evidence the model cited from alias A was indeed sufficient to determine the correct answer, reinforcing that the model’s answer was based on that evidence. - **Consistency Signals:** We may also log qualitative consistency markers. For example, did the model explicitly mention alias A or the concept’s definition from earlier while answering B? Did it confuse the alias with another concept? One way to quantify this is to check if the notes or rationale used for alias B contain the alias A term or other identifiable overlap. Another signal: whether the model’s answer under alias B is the same (in meaning) as its answer under alias A (for instance, if both answers ultimately point to “classical conditioning” concept). Consistency can be measured by overlap in cited note IDs or content between A and B responses. Logging which note lines were cited by the model for B is helpful – we can then see if those lines originated from the alias A question’s rationale. - **Timing or Attempts:** Optionally, log how many attempts or how much time between alias A and alias B exposure. This can help correlate with spacing effects (e.g., did a longer gap reduce transfer success?). Also record if the model saw additional examples or hints in between.

All this data should be stored per user (or per simulation run) in the stats or a separate log specific to research trials. We might extend the user\_stats.json or maintain a parallel log file for detailed trial-by-trial data. The goal is to enable analysis of how the alias mapping was learned and used.

### Metrics for Evaluation

Using the logged data, the following key metrics will be computed to quantify alias-transfer performance:

* **Alias-Transfer Delta:** This metric captures the drop (or change) in performance when switching from alias A to alias B. Concretely, we can define *alias-transfer delta* as the difference between the model’s accuracy on alias A questions vs alias B questions for the same concept. For example, if a model answers 100% of Zim-7 (alias A) questions correctly but only 60% of Concept-92 (alias B) questions correctly for a given concept, the transfer delta is -40 percentage points. We can aggregate this across all tested concepts to see the average delta. A smaller delta (close to 0) indicates robust transfer – the model performs equally well under the new alias – whereas a large negative delta indicates that the alias swap confused the model and performance dropped. In some cases, delta could be positive (if B happened to be easier or the model eventually did better), but generally we expect a drop initially. This metric directly measures the **cost of alias substitution** on performance.
* **Binding Slope:** The *binding slope* measures how quickly the model “binds” the new alias to the known concept with additional exposure or context. There are a couple of ways to quantify this:
* One interpretation is to plot the model’s success probability on alias B as a function of how many alias A examples it saw beforehand, and then measure the slope of that curve. For instance, if with only 1 exposure to alias A, the first alias B question has a 20% success rate, but with 3 exposures to alias A (e.g., repeated practice or notes review) the first alias B success rate rises to 80%, that indicates a steep positive binding slope. We could fit a simple linear model or calculate the difference per additional A example. A steeper slope means the model rapidly improves its alias-binding with more training on A.
* Another way is within the alias B attempts themselves: if we allow multiple alias B questions (or retries) for the same concept, does performance improve from the first B encounter to the second? For example, the model might fail the very first time it sees alias B, but after seeing the explanation for that failure (which effectively reveals the mapping in retrospect), it gets it right the next time alias B appears. The improvement from 0% to 100% over one trial could be considered an indicator of binding speed as well. We might measure the number of trials needed for the model to consistently answer alias B correctly after the initial introduction – fewer trials = stronger immediate binding.
* However we define it, the binding slope is about **learning rate of the alias mapping**. It complements the delta: delta measures initial drop, slope measures recovery or adaptation rate.
* **Coverage and Support Metrics:** We can also compute average *coverage* on alias B answers and the rate of *witness\_pass* on those answers. For example, an alias-transfer **support score** could be the percentage of alias B answers that passed the witness check (i.e., were supported by correct evidence from alias A notes). A high support rate means when the model does get B right, it usually has the evidence to back it up – implying true comprehension. A low support or coverage could flag that the model might be guessing or not clearly linking to prior context.
* **Consistency Index:** Based on the logged consistency signals, we can create a metric for consistency. For instance, the proportion of alias B responses where the model’s explanation or citations explicitly reference the alias A context might serve as a *consistency index*. Alternatively, consistency can be defined as the agreement between what the model does under alias A and alias B – if a model answered alias A correctly but alias B incorrectly, that’s an inconsistency in applied knowledge. We might track the fraction of concepts for which the model answered A correctly *and* B correctly on first try (successful transfer), versus those where A was correct but B was wrong (transfer failure due to inconsistency). This essentially is another view of the transfer success rate, but framed as consistency of knowledge across symbol change.
* **Per-Alias-Set Stats:** In addition to differences, we will report raw performance on alias A and alias B sets. For example, **Alias A accuracy** might often be high (since the model eventually learns with direct exposure), whereas **Alias B first-attempt accuracy** might be lower. These can be tracked over time or across variants to see if improvements to training (or model updates) yield better transfer.

All metrics should be computed at the level of each alias family (per concept) as well as aggregated to identify overall trends. For instance, we might find some concepts transfer better than others, or that on average alias B performance is X% lower than alias A. Monitoring the *alias-transfer delta* across different conditions (with vs without additional cues, or varying time delays) can provide insight into the model’s in-context learning robustness.

### Integration into Simulator Loop

We will incorporate the alias-swap experiment seamlessly into the existing ICL simulator run loop. Key integration points include:

* **Content Generation/Selection:** Ensure the content library or item generator is aware of alias-swap families. If questions are generated dynamically, we might add a template or prompt that produces alias variants given a real concept. If content is pre-authored, we label those pairs accordingly. The selection algorithm (/api/next or the student harness) should be able to pick the alias B counterpart when appropriate. One approach is to add a check after an item is answered: if that item was an alias A introduction and was answered correctly (or the notes from it are now available), schedule the alias B item after a certain number of other questions. This could be implemented by queuing the B item in the adaptive policy (for example, treating it similar to a “review” that is due soon, since it’s a test of applying learned knowledge in a new way).
* **Anonymization Controls:** Normally, anonymization is static per user[[6]](file://file-3LWaoPLYdoW8YgsE374zZh#:~:text=,%28default%200.4). To implement a true alias swap within one session, we have a few options:
* **Dynamic Codebook Update:** Programmatically update the anonymization codebook for that concept when it’s time to introduce alias B. For instance, initially map *classical conditioning* → "Zim-7". Later, alter the mapping to *classical conditioning* → "Concept-92". This would cause any new question generated on that concept to use the new alias. We must ensure the old alias doesn’t appear in subsequent content to avoid confusion (unless intentionally presenting both to see if model distinguishes them).
* **Alternate Representation:** Alternatively, treat the alias B question as if it were a different concept internally. For example, have a dummy skill id that is essentially the same content but triggers a different alias. The simulation logic knows that skill\_X\_aliasB actually refers to the same underlying skill\_X\_concept. It can then aggregate their stats for analysis. This way, the anonymization can remain per “skill,” and since we use a different skill entry for alias B, it naturally gets a different alias string from the codebook. After the run, we map those together for evaluation.
* The implementation choice may depend on ease: updating the codebook on the fly might be complex, so using dual skill identifiers (one for A, one for B) could be a simpler path. In either case, the run loop or content picker needs to manage that when alias B skill is served, it references the notes from alias A’s content. This could be handled by merging the note pools or by embedding a hint in the B question that links back to A’s notes as discussed.
* **Session Notes Handling:** Because the synthetic student relies on the notes\_file (accumulated evidence) in closed-book mode, we must ensure that the notes from the alias A question are retained and available when alias B question arrives[[4]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU). If using separate skill IDs for A and B, the notes system might treat them separately by default. We should thus design the notes such that the explanatory text for alias A contains the meaningful content (which it naturally will – e.g., the definition or key facts of the concept). The student bot should append those notes as usual. When alias B question comes, the student will search its notes; since the term "Concept-92" itself might not appear in notes, the model will look for related cues (like Pavlov, or the general description) that match the question. This should suffice for it to find the alias A notes relevant. We may not need any special handling beyond ensuring the clues line up. However, it’s important the student *not* be reset or restarted between A and B – it should be one continuous session so that the notes context carries over.
* **Recording Results:** Extend the /api/record or the student harness logging to include the alias-set tag and to store coverage/witness results. As noted, when TUTOR\_REQUIRE\_CITATIONS=1 is on, the server already returns fields like credited (score), coverage, and witness\_pass in the record response[[5]](file://file-3LWaoPLYdoW8YgsE374zZh#:~:text=,witness_pass). The run loop (student bot) can capture these and log them per question. We might add custom logging specifically when reason == alias\_transfer\_test to mark that this was an alias-swap trial.
* **Edge Cases:** If the model fails alias A repeatedly, alias B might be meaningless to test (since the concept wasn’t learned at all). The policy could either skip alias B or present it as an interesting case of “zero-shot transfer” (likely failure). Likewise, if the model answers alias B correctly without having answered A correctly (perhaps by chance using elimination or partial knowledge), the experiment still yields data (and witness gating would likely fail due to lack of cited evidence). We should handle these cases in analysis (e.g., filter out families where A itself wasn’t learned when interpreting transfer metrics).

By integrating at these points, the alias-swap experiment becomes part of the normal training/testing cycle of the simulator. For example, a session for a synthetic student might look like: new skill question → remediation → alias A question on key concept → correct → [notes recorded] → a few more questions → alias B question (transfer test) → record result → continue with next skill, etc. This ensures the model’s learning is probed from multiple angles in one run.

### Dashboard and Analysis Integration

To treat alias-transfer testing as a **first-class diagnostic**, we will extend the research dashboard and analysis tools to include the results of these alias-swap experiments, alongside existing metrics like learning curves, spacing effects, reflection outcomes, and position effects. Concretely:

* **Run Dashboard Updates:** If we have an admin or researcher dashboard (e.g., via /api/dashboard or an offline analytics notebook), include summary statistics for alias-swap tests. For instance, the dashboard can display a panel with:
* **Alias-Transfer Success Rate:** the percentage of alias B questions answered correctly (perhaps with witness criteria) given that alias A was learned. This could be shown overall and broken down by concept.
* **Avg. Alias-Transfer Delta:** a single number or small table showing the average accuracy on A vs B and the difference. If we want to be visual, a bar chart could show side-by-side performance for A and B, or a line chart could show A->B drop for each concept tested.
* **Binding Slope Visualization:** The dashboard could plot how performance on alias B improves with more alias A exposures. For example, one chart might have X-axis = number of times concept seen with alias A (in prior questions or notes) before alias B, and Y-axis = probability of correct on first alias B attempt. This would produce a curve where we can compute the slope. A steeper curve indicates faster binding. We might overlay curves for different model versions or different conditions (to see if an intervention improves symbol binding).
* **Coverage/Consistency Info:** We can display the average coverage of notes in alias B answers or the frequency of witness\_pass. A high-level indicator could be “Evidence-backed Transfers: X%” meaning X% of the time the model not only got alias B right but did so with proper evidence (witness pass). This helps distinguish lucky guesses from true in-context learning.
* **Examples and Flags:** It may be useful to list a few example alias-swap cases, especially failures, for qualitative insight. The dashboard might allow drilling down into a particular alias family to see what alias A was, what alias B was, and how the model responded. For instance, it could show: *Concept: classical conditioning; Alias A: Zim-7 (answered correctly, note saved); Alias B: Concept-92 (model answered incorrectly, witness\_pass=false)*. Such details help in debugging and refining the alias testing approach.
* **Integration with Learning Curves:** We should incorporate alias-swap data into existing learning curve plots. For example, if a learning curve normally shows the model’s accuracy improving over repeated practice on a concept, we can mark the point where the alias was swapped and see if there’s a dip. This can be visualized as a discontinuity or a separate curve. If the dashboard tracks performance over time, a special marker can indicate “alias B introduced here” and show the immediate effect on performance.
* **Spacing & Reflection Synergy:** Because alias-swap might interact with spacing (e.g., longer gaps could make it harder to connect aliases) and reflection (perhaps a reflective note by the model could solidify the alias mapping), the dashboard can cross-reference these. For instance, we might have filters or breakdowns: *alias-transfer delta for items with short gap vs long gap*, or *alias-transfer success with vs without a self-reflection prompt in between*. This helps integrate alias-swap testing with the other diagnostics rather than treating it in isolation.
* **Automated Reporting:** If the pipeline produces regular reports or if the SRS/STS requires tracking certain NFRs, alias-transfer metrics should be included. For example, if there’s a requirement that the system support “provable learning,” we can report alias-transfer as evidence of the model’s ability to learn new terms. This could be documented in the SRS as a test: *“Given an anonymized concept alias that the model has seen in context, when that concept is referenced with a new alias later, the model’s performance drop (alias-transfer delta) should be below some threshold and the model should cite prior context (witness\_pass=true) in at least Y% of cases.”* Making it first-class means it’s part of our success criteria.
* **User Interface (if needed):** Although primarily a research tool, if the tutor UI has a progress or diagnostics section (like the Progress tab showing mastery bars), we might not surface alias-transfer to end users. However, for internal use, we could have a hidden or admin view. The /api/dashboard could be extended to return alias-swap stats (e.g., number of alias tests delivered, % correct, etc.), which a developer can view. This was not in the original UI, so it might remain a back-end analytics feature.

In summary, by logging the right data and extending the analysis tools, alias-swap transfer testing becomes an integral part of the ICL simulator’s evaluation suite. We will be able to continuously monitor how well the model copes with arbitrary symbol remapping – a critical aspect of in-context learning. This integration means that every simulation run can produce not just learning curves and memory retention stats, but also a clear measure of **symbol binding and transfer**. Over time, this will allow us to answer questions like: *Is the model improving at learning new aliases? Does a certain training method (or model architecture) yield smaller alias-transfer deltas?*, making alias-swap tests as routine and informative as spacing or position effect diagnostics in our research pipeline.

**Sources:**

* Platform anonymization and evidence gating features[[1]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU)[[5]](file://file-3LWaoPLYdoW8YgsE374zZh#:~:text=,witness_pass), which enable alias-based testing and witness-verified scoring.
* Simulation mode enforcing learning from NOTES (no pre-trained knowledge)[[4]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU), ensuring alias terms like "Zim-7" and "Concept-92" are truly novel and must be learned in context.

[[1]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU) [[2]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU) [[3]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,answer%3B%20no%20raw%20domain%20terms) [[4]](file://file-CKrkCcfVVJgj6HEVRpw3KQ#:~:text=,TUTOR_COVERAGE_TAU) README.md

<file://file-CKrkCcfVVJgj6HEVRpw3KQ>

[[5]](file://file-3LWaoPLYdoW8YgsE374zZh#:~:text=,witness_pass) [[6]](file://file-3LWaoPLYdoW8YgsE374zZh#:~:text=,%28default%200.4) AGENTS.md

<file://file-3LWaoPLYdoW8YgsE374zZh>